Network Virtualization: Distributed Computing and a more secure Internet

Jason Lin
Enterprise Network Design
University of Southern California
jasonjli@usc.edu

Abstract

The Internet is arguably one of our civilization's greatest inventions. From its roots as DARPA's military funded research project ARPANET, the modern day Internet is built upon sophisticated and robust protocols that support billions of users at the same time. Scalability is enhanced by its rigid but complex infrastructure, however it was not originally built with security in mind.

Running in its backbone is the TCP/IP communications protocol, devised to ensure reliability when transferring duplex data. Through decades of popularization, the Internet has evolved into a form bounded by a complex set of diverse stakeholders. Due to its wide-use and multi-provider nature, adopting a new architecture or groundbreaking modification requires a near impossible consensus from conflicting stakeholders. Yet modern Internet was developed at a time that preceded the inventions of mobile computing and smart connected devices, which together have contributed immensely to the population of connected devices we have today – so much that the depletion of IPv4 addresses resulted in the introduction of the new standard IPv6.

For most enterprises, the very complexity of the network makes it hard to introduce changes from the perspectives of management, configuration, and control. In this paper, we explore network virtualization's existing applications and its potential for piloting a new network architecture rested upon proven concepts of distributed computing and virtualization technologies. In doing so, we justify an outlook on the future of network – a portable, dynamic and customizable heterogeneous Internet that is responsive to unpredicted contingencies.

1. INTRODUCTION

Rising to its popularity in both academia and industry in recent years, network virtualization explores the frontier of network architecture research. As a means of abstraction of resources, its original role served a similar purpose as virtual machines are to operating systems. A formal definition for the term virtualization is generally stated as the transparent abstraction of physical computing resources that exposes a platform supporting multiple logical views of their properties [1]. Among its many benefits, network virtualization emerges as a foundation for next generation networks, providing improved manageability and mobility for network administrators and users. Built upon a plethora of existing virtualization and cloud technologies, network virtualization serves as the completing piece that will fully interconnect all other virtualized appliances to create a complete ecosystem of a virtualized computing environment [2]. Existing concepts of virtualization applied to operating systems, storage systems, servers and data centers are thus highly transferrable to the development of virtual networks.

From a historical standpoint, the Internet was developed at a time when operating systems depended on a centralized publisher-subscriber model by IBM, dubbed the "Systems Network Architecture (SNA) model [1]. The concept of a local area network later spawned UNIX, the first successfully adopted Networking Operating System (NOS) used by academia and scientific community alike thanks to its portability and availability. The separation of host entities as a distributed and localized full-fledge computing nodes when developing the Internet has to some extent hindered mass adoption of distributed virtualized environments today. The Internet as of current composes of components of high centrality – routers, DNS, root name servers. Such centrality can add to the overhead of network management while rendering connected guest devices much more vulnerable to attacks that target the central devices, an example being the recent DDoS attack on DNS Service provider Dyn that took down a long list of high profile websites including GitHub, Twitter, Reddit, and Netflix.

2. OVERVIEW AND STATE OF THE ART

2.1 Security in 21st century

The Internet today can be highly insecure in face of new technologies like the Internet of Things (IoT) due to exposed vulnerabilities of its unsecured traffic and session layer protocols. The unprecedented DDoS attack on Dyn's DNS servers late in October 2016 was in fact attributed to a botnet consisted of hundreds of thousands of IoT devices. Initiated by a malware called Mirai, 100,000 security cameras with networking capability were enlisted to direct bot traffic to Dyn's servers, congesting them with up to 50x normal traffic volumes across a large number of IP addresses. [3] A fundamental insecurity lies in the authentication mechanism underpinning the HTTP protocol. This vulnerability comes before the application and presentation layers that sit atop the session layer, where by exploiting the TCP handshaking mechanism that freely allows for one-way handshake initiations, malicious requests can be essentially indistinguishable from valid traffic, regardless of whether the data transfer is encrypted (i.e. with TLS/SSL) on the application layer. In Dyn's analysis of its attack, it was indeed crippled with the ability to discern legitimate traffic, which could sometimes spark in the event of unexpected demand, from attack traffic originated from all geographies [3]. The attack was further exacerbated with a storm of legitimate retry activity as recursive servers attempted to refresh their caches.

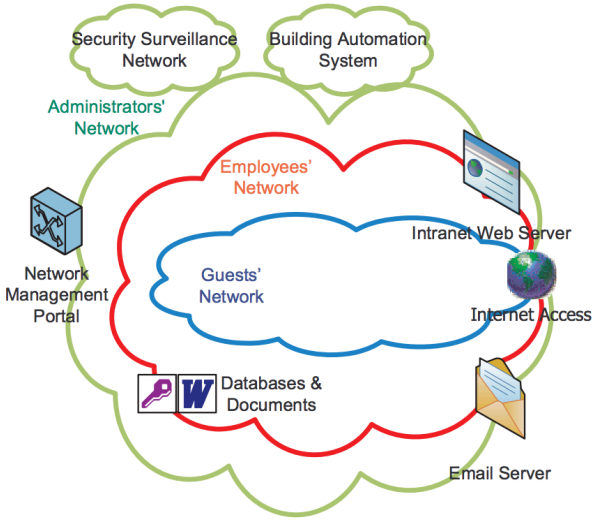
The attack on Dyn showed that even if we inspect our traffic, we are often hit with an uncertainty with the legitimacy of incoming and outgoing traffic, due to the public nature of our Internet. Because routers in a network have to be able to freely contact others to route packets using the OSPF routing protocol, most local area networks inevitably expose ports to the public. Any medium that does not inspect its traffic runs the risk of carrying out new attack surfaces: if a packet is transported without verifying its tag for security, we may well be transporting insecure and harmful data all along. Network virtualization could bring the benefit of dissociating infrastructure from security requirements while establishing a central policy for security and risk management level [4].

2.2 Comparable Virtualization Technologies

Virtual LAN (VLAN) – sharing of multiple logical local area networks on a physical network at the data link layer of OSI (L2 constructs). VLAN partitions the same broadcast domain into

isolated network segments. Only physical routers in LAN configurations traditionally provided such a service. *Frame coloring*: 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 [2].

Virtual Service Networks (VSN) – generalization of VLANs to the scale of multiple network instances. When defining multiple network instances that share physical resources, VSN requires that the networks are properly isolated and segmented in functionality, access permissions, etc.



Virtual Private Networks (VPN) – connect multiple sites using tunnels over public network. Because VPN's hardware spans L3, L2, L1 technologies, its customizability is limited to L4 and above, i.e. transport layer technologies like TCP. Can be distinguished between Intranet vs. Extranets.

Active and Programmable Networks – the Internet is increasingly heterogeneous and dynamic, and thus demand for an adaptive and resource restrictive network arises. Active networks allow users to inject customized programs into nodes of the network [5]. It offers applications the opportunity to benefit from processing and storage in network nodes such that customized computations can be performed at strategic locations, with the potential to improve communication performance and to automate the deployment of new protocols and services. 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 [2]. In order to accommodate coexisting networks or multiple 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. Thanks to the independent programmability of virtual networks, any kind of network architecture can theoretically be built over a programmable virtual network.

3. EXISTING NETWORKING CONCEPTS

3.1 Quality of Service (QoS)

To ensure an organization or enterprise's success, communications network must be provisioned to certain technical standards. Quality of Service refers to both standardized measures (transmission rates, error rates, latency, throughput, uptime, jitter, etc. loss characteristics) of a network's performance and set of techniques (i.e. traffic shaping: packet prioritization, bandwidth throttling, policy-based application classification, etc.) to manage network resources to achieve optimized performances. Typically applied to high-bandwidth

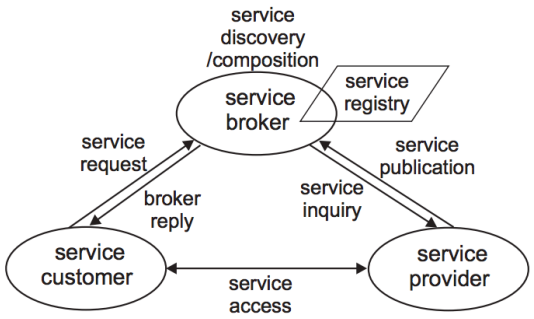
multimedia streaming traffic such as video on demand, IPTV, VoIP, QoS provides priority to networks and guarantees parameters of high performance. The need for QoS arises from that transmission of multimedia content dependably using “best effort” protocols such as UDP can result in a asasfdfs amount of data loss while TCP could incur much overhead.

3.2 Differentiated Services & Integrated Services (DiffServ, Int-Serv)

Differentiated Services is a QoS forwarding mechanism that is used for a number of mission-critical applications and to provide end-to-end QoS. Its main goal is to **classify** traffic into groups of delivery priorities and guarantees. DiffServ represents a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for such classification and management of network traffic. Integrated Services refers to an architecture that specifies the elements to guarantee QoS. It is based mainly on reserving resources per session and limit demand to the capacity than can be handled by the network [6].

3.3 Service Oriented Architecture (SOA)

Service-Oriented Architecture (SOA) encourages individual units of logic to exist autonomously yet not isolated from each other. SOA provides an effective solution to coordinating computational resources across heterogeneous systems to support various application requirements. It is an architecture within which all functions are defined as independent services with invokable interfaces that can be called in defined sequences to form business processes.



As a paradigm for organizing and utilizing services and capabilities that may be under the control of different ownership domains. SOA enables virtualization of various computing resources in form of self-contained services and provides a flexible interaction mechanism among services. These platform-independent services can be described, published, located, orchestrated, and programmed through standard interfaces and messaging protocols, abstracting away implementations of their functions. A key feature of SOA is loosely-coupled interaction among heterogeneous systems in the architecture [7]. SOA enables more flexible and reusable services that may be reconfigured and augmented more swiftly than traditional system construction; thus can accelerate time-to-business objective and result in better business agility. SOA also provides a standard way to represent and interact with application functionalities thus improving interoperability and integration across heterogeneous systems.

Overall, the consolidation of security and privacy services like authorization, authentication, encryption, firewalls and anti-malware programs is the most significant attribute of SOA networking. This kind of consolidation cuts down the intricacy of network management and the potential risk of network vulnerabilities.

3.4 Infrastructure as a Service (IaaS)

Infrastructure as a Service is a model of SOA used to define cloud computing along with other

standards. It aims to offer highly scalable resources in the form of services that can be adjusted on-demand, abstracting the user from details of physical infrastructure.

3.5 Tunneling and Encapsulation [8]

Tunnels (often using encapsulation techniques) provide virtual (logical) links to connect network devices that are not physically adjacent. For example, tunnels may be used to create the illusion for some protocols running on a network device that this device has a direct connection to another device even when no physical link between the two devices exists. Some popular technologies include generic routing encapsulation (GRE) tunnels, Internet Protocol security (IPsec) tunnels, GPRS Tunnelling Protocol (GTP) tunnels, and MPLS label switched path (LSP) tunnels. Essentially, tunnels are overlay links and form the fundamental building blocks of overlay networks (further defined below).

3.6 Distributed Computing

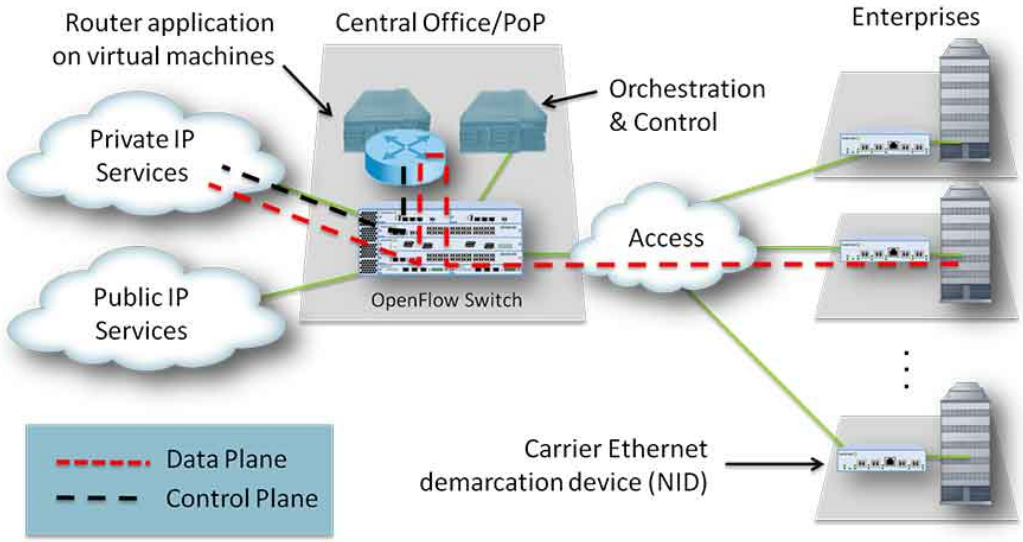
A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The network will become a moldable infrastructure. Compared to traditional physical counterparts, it has the benefits of being scalable, economical, and more resilient & performant.

4. COMMERCIAL NV TECHNOLOGIES

4.1 Network Functions Virtualization (NFV)

Network functions virtualization (NFV) offers a new way to design, deploy and manage networking services. NFV decouples the network functions, such as network address translation (NAT), firewalling, intrusion detection, domain name service (DNS), and caching, etc., from proprietary hardware appliances so they can run in software.

NFV is designed to consolidate and deliver the networking components needed to support a fully virtualized infrastructure – including virtual servers, storage, and even other networks. It utilizes standard IT virtualization technologies that run on high-volume service, switch and storage hardware to virtualize network functions. It is applicable to any data plane processing or control plane function in both wired and wireless network infrastructures.



4.2 Software-defined Networking (SDN)

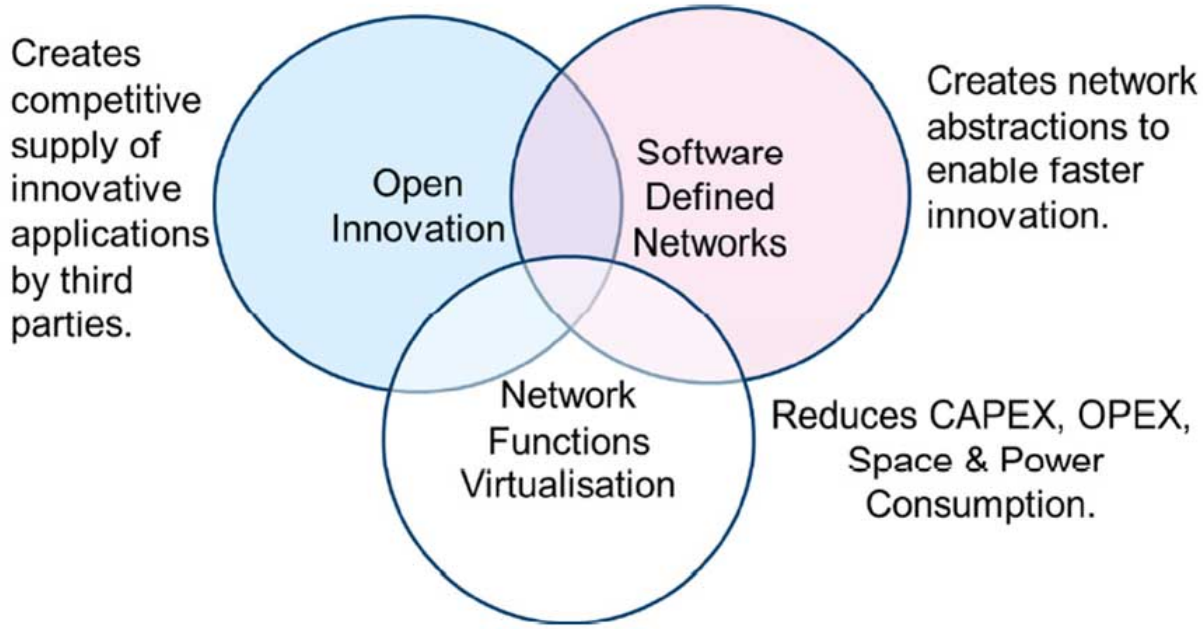
Software-defined Networking (SDN) separates the control and data planes of the network to control the topology and behavior independently of data forwarding, providing flexibility not found in traditional network devices. Compared to traditional networking methods, SDN emphasizes on a centralized control system that oversees creation and access to virtual networks software-defined. A core router centrally decides where data can go via its control plane and helps transport the data via data plane. SDN breaks the link between control and routing and distributes it in a way that decentralizes this optimal decision making process, therefore such distributed control is faster and more efficient.

SDN systems are better capable of using centralized policy control to configure the network in which systems policy, such as security or performance requirements, are defined and declared in a manner meaningful for application owners, as opposed to infrastructure owners. For example, an application owner is concerned with declaring what type of controls are placed on the network, as opposed to how they are done. There has traditionally been a gap between the intent of the application owners and infrastructure owners. If the desires are not expressed clearly, they are often misinterpreted by the infrastructure owners. Another example is whether or not a user acceptance test (UAT) version of a multi-tier application ought to have access to “live” data in a customer database in order to test it under production scenarios. We realize that it is difficult to construct a realistic test replica of a customer database, so although that may be undesirable, it may be necessary to resort to connect to live data during off-hours to validate the UAT system before switching into production. Having an automated policy system to manage this will make it much simpler for application and test teams to administer. Examples of SDN systems include Cisco's ACI, as well as the open source Group Policy project under the OpenDaylight or OpenStack's Neutron projects.

4.3 OpenFlow

As SDN started to gain more prominence it became clear that standardization was needed. The Open Networking Forum (ONF) was organized to formalize a singular approach for controllers to communicate with network elements. Named OpenFlow, this approach was proposed by researchers at Stanford University to enable flexible control of a switch’s data plane. It defines both a model for how traffic is organized into flows, and how those flows can be controlled as needed. OpenFlow enables programmability on commodity hardware using FPGA-based routers and switches with competitive performance [9]. The separation of the controller makes it possible for researchers to experiment with new protocols by simply programming the controller remotely, which can be a simple PC [8], and has lay technical ground for SDN.

The following exhibits an overlapping yet complimentary relationship between SDN vs. NFV, while there is a trend moving towards open source unifying the two with additional features.

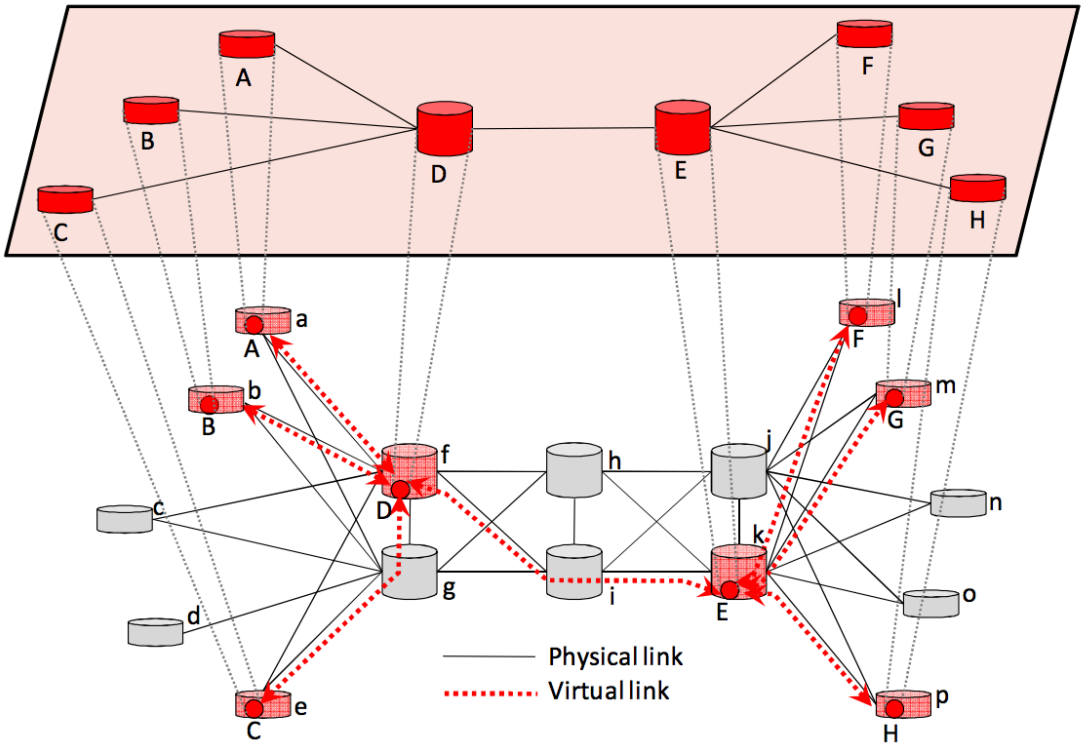


5. INDUSTRY APPROACHES

5.1 Overlay Networks

Application layer virtual networks that are built upon existing network infrastructure using tunneling or encapsulation. This is used when there is a need to implement new services without changes in infrastructure. As an incremental upgrade, it is used to induce innovation in ossified Internet 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 and availability of Internet routing, enabling multicasting, providing QoS guarantees, protecting from denial of service attacks, and for content distribution, file sharing and even in storage systems. Overlays have also been used as testbeds (e.g., PlanetLab) to design and evaluate new architectures. In addition, highly popular and widely used peer-to-peer networks are also overlays in the application layer. However, in their seminal paper on network virtualization, Anderson et al. 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.

5.2 XEN

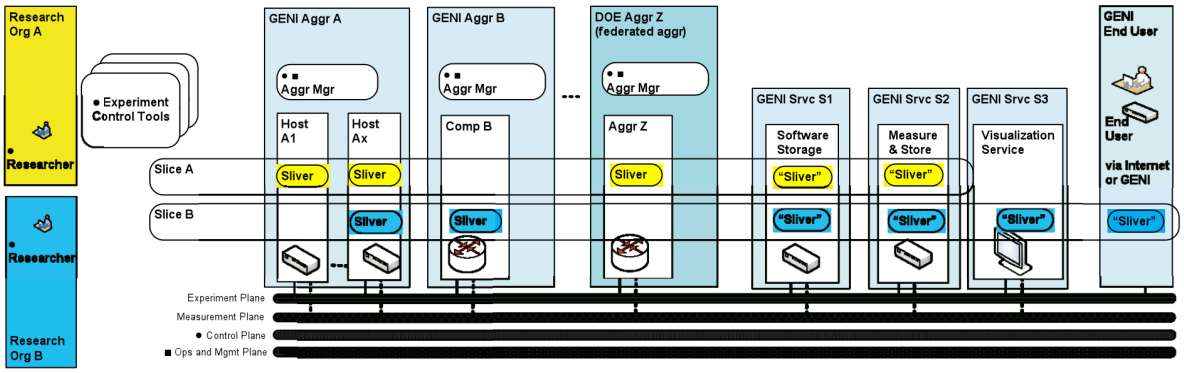
As the leading open source virtualization platform, Xen is an x86 based hypervisor that allows multiple operating systems to run on the same hardware. It is capable of virtualizing operating systems as virtual routers for use in virtual networks. In a traditional Virtual Machine Monitor (VMM), the hardware is emulated. In contrast to this technique, Xen uses paravirtualization to host a guest OS. This means a similar but not identical software interface is provided for the hardware. Subsequently some slight modifications of the OS are necessary. But the relative low costs to port an OS to Xen have made it a popular choice for deploying virtual networks. This results in a system nearly as efficient as a native system [9]. Compared to "normal" virtualization of a traditional operating system, Xen can provide performance isolation that ensures a VM's performance cannot impact the performance of another one.

6. RESEARCH FRONTIERS

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.

6.1 GENI

Open large-scale experimental facility for researchers: deploy and evaluate new network architectures for users: carry real traffic connects to Internet to reach external sites Supports slices of resources partitioned in space and time.



The Global Environment for Network Innovations (GENI) 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 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.

6.2 New Generation Network (NwGN)

Pioneered by University of Tokyo's NetworkVirtualizaiton Research Lab, a new generation network at least two generations ahead may be built upon the advanced and innovative use of network virtualization as an architecture basis. *Elastic Networking:* While this gears it towards a new paradigm "Network as a Service" and upscales its cloud usability, it proposes a meta-architecture to enable the synchronization of multiple network architectures to operate user and application specific logical networks simultaneously and securely [10].

6.3 FIND

FIND (Future Internet Design) is an initiative recently proposed by National Science Foundation (NSF) of the U.S. for the research community to collectively conceive what would become the future of networking – a redesigned Internet from scratch. Two key issues such a new network hopes to emphasize on from the start are fundamental security and availability.

7. FUTURE OF THE INTERNET

Instead of patching “band-aids” to the Internet therein after catastrophic events of disruption and malicious efforts, the research community has long been calling for a new Internet reinvented from the group up. While it may take time for the injection of a new Internet to replace existing infrastructure, global efforts from numerous research communities have all acknowledged the importance of paralleled network virtualization as the foundation to a dynamic and error–proof Internet. Therefore, one key research direction today is to find a viable global connectivity-enabling framework that propels the testing of NV’s plausibility with real users.

References

1. D. Costa, **History of Network Operating Systems**, Techwalla. Leaf Group, 2014.

2. N.M. Mosharaf, R. Boutaba, **A survey of network virtualization**, Elesevier B.V. Computer Networks, vol. 54, 2009

3. S. Hilton, **Dyn Analysis Summary Of Friday October 21 Attack**, Dyn Blog, 2016.

4. B. Germain, **Network Virtualization to Enhance Visibility and Containment**, IEEE Communications Society, 2016

5. D. Tennenhouse, D. Wetherall, **Towards an Active Network Architecture**, In proc. of Multimedia Computing and Networking 96, 1996.

6. T. Abbas, IntServ & DiffServ, **Advanced High Performance Network**, 2013.

7. Q. Duan, Y. Yan, A. Vasilako, **A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing**, IEEE Transactions on Network and Service Management, vol. 9, no. 4, pp. 373-376, 2012.

8. A. Wang, M. Iyer, R. Dutta, G. Rouskas, I. Baldine, **Network Virtualization: Technologies, Perspectives, and Frontiers**, Journal of Lightwave Technology, vol. 31, no. 4, pp. 526-530, 2013

9. K. Rausch, **Network Virtualization – An Overview**, Seminar FI & IITM SS, Network Architectures and Services, 2011.

10. University of Tokyo, **Network Virtualization as Architecture and Its Applications**, IETF 2009-ISOC Aki NAKAO, 2009.