Enabling and Testing Network Functions Virtualization (NFV) to Ensure Carrier-Grade Delivery
Table of Contents

Server Virtualization as a Building Block for NFV ............................................. 4
NFV Driven by ETSI ISG .................................................................................. 6
SDN and NFV ................................................................................................. 9
  How are SDN and NFV Related? .................................................................. 11
  Integrating Virtualization, Networking, Orchestration, & Management .... 12
Real-World NFV .............................................................................................. 12
Testing NFV to Ensure Carrier-Grade Service Delivery ............................... 13
  Test Scenarios ............................................................................................ 15
    Example of Testing L2/3 Virtualized Functions ..................................... 16
    Example of Testing L4-7 Virtualized Functions ..................................... 17
    Example of Inserting Testing into the Virtualized System .................. 18
Ixia Virtualization Test Solutions ................................................................. 19
  IxVM .......................................................................................................... 19
  IxVM – Network ....................................................................................... 19
  IxVM – Load .............................................................................................. 19
RackSim ......................................................................................................... 20
IxChariot ......................................................................................................... 21
Phantom Virtualization Tap ........................................................................... 21
SDN-Related Features ................................................................................... 21
Server Virtualization as a Building Block for NFV

The term “virtualization” means many things to many people. In the realm of compute, many people think back to companies like IBM, Bell Labs, GE, and others who were some of the first to enable multiple processes to run on a server or mainframe. Most of the benefit was based on time-sharing a computer’s processing to enable multiple programs or multiple users. This was based on dividing up the memory and other resources like CPU per process and/or per user. Modern server virtualization still leverages many of those basic concepts. This evolution progressed to provide virtualization at the user- or workspace-level.

A major breakthrough occurred in the late 80’s when Insignia Solutions demonstrated a software emulator called SoftPC. It allowed users to run DOS applications on their UNIX workstations. This provided significant cost savings over having to purchase a dedicated PC to run MS-DOS. They also released a Mac version and added the ability to run Windows in addition to DOS. With the success of Insignia Solutions, many other companies began to get in the game, including Apple who released Virtual PC and VMWare who has become the leader in server virtualization. Running a virtual machine (VM) on top of an existing operating system is known as a Type-2 Hypervisor. A Type-1 Hypervisor does not require a host operating system (OS) and is much more efficient in terms of the use of the underlying hardware resources.

Since releasing ESX Server in 2001, VMware has grown to be the largest provider of virtualization software. Microsoft, who acquired Virtual PC maker Connectix in 2003 and also released Microsoft Virtual Server 2005, has been growing market share. Their current offering is Hyper-V. Citrix, who acquired XenSource in 2007 and renamed the product to XenServer, has also built strong market share.

KVM (Kernel-based Virtual Machine) is open-source virtualization software that turns Linux into a hypervisor. The kernel component of KVM is included in mainline Linux as of 2.6.20. KVM was developed by a startup company called Qumranet, which was acquired by Red Hat in 2008. KVM is available in Red Hat Enterprise Linux. Oracle is also becoming a player in the virtualization market with the introduction of Xen-based OVM (Open Virtualization Format) in 2007.

Initially, a key driver of this technology was purely cost savings realized by increased server use. What developed over time are many features that have transformed operations and created abundant revenue-generating service offerings.

Since server virtualization has been introduced to this space, users now have access to key features including:

- Virtual machine creation – partitioning a physical server into multiple VMs
- Physical to virtual migration – converting a physical server to a virtual server
- High availability – ability to start, stop, and restart a VM on another server in case of a hardware failure or maintenance
- Live migration – ability to move a running VM from one server to another without downtime
- Template/image library – ability to create and save VM templates to enable the rapid provisioning of new VMs
While the features listed above are impressive, they are considered core requirements while these additional features differentiate vendor offerings:

- **Storage live migration** – moving VMs between storage arrays without downtime
- **Storage array integration** – offloading storage tasks from host servers to the storage array
- **Advanced memory management** – using memory ballooning technology and transparent page sharing
- **High availability and fault tolerance** – continuous availability of VMs during hardware failures
- **Chargeback** – accurate measurement of resource use by department or business unit
- **Storage DRA** – automated monitoring and load balancing of storage to avoid storage bottlenecks
- **Share nothing migration** – ability to move a powered-up VM between servers without the need for shared storage
- **Heterogeneous hypervisor management** – management of hypervisors from other vendors
- **Automated site recovery** – leverages storage area network (SAN)-based replication to keep VM files and configuration data at backup location
- **Network services** – network optimization techniques to reduce network congestion

### Summary of leaders in server virtualization

<table>
<thead>
<tr>
<th>Company/Product</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware – ESXi, vCloud, vSphere, VCenter (and others)</td>
<td>Most mature virtualization product line, very feature rich, solutions for enterprise and cloud although a high priced solution</td>
</tr>
<tr>
<td>Red Hat – KVM in Red Hat Enterprise Virtualization</td>
<td>Open-source, feature rich</td>
</tr>
<tr>
<td>Citrix – XenServer, XenDesktop, CloudPlatform (and others)</td>
<td>Strong for cloud infrastructure and desktop virtualization; feature rich, good usability, highly scalable</td>
</tr>
<tr>
<td>Microsoft – Hyper-V aka Windows Server Virtualization in Windows Server 2012 R2</td>
<td>Although quickly gaining momentum and becoming feature rich, there is still some feature gap compared to leaders; high priced</td>
</tr>
<tr>
<td>Oracle – VM Manager, VM Server</td>
<td>Good usability and pricing, catching up on features</td>
</tr>
</tbody>
</table>
Server virtualization has enabled:

- Server consolidation and improved management
- Service agility – ability to move, add, change services as needed
- Service programmability – ability to program the delivery of services through APIs
- Reduction in overall cost by optimizing server use and management costs
- Service scalability and elasticity
- A “pay as you go” model
- Cloud-based service offerings including private cloud and public cloud, with options that include:
  - IaaS – Infrastructure as a Service
  - PaaS – Platform as a Service
  - SaaS – Software as a Service

Through this innovation, many companies have launched successful commercial offerings including Google, Amazon, Oracle Cloud, Salesforce.com, Microsoft Azure, and many others.

**NFV Driven by ETSI ISG**

Network Functions Virtualization (NFV) is a paradigm shift in networking as we know it, moving network functions like switching, routing, firewall, BRAS, and many others from hardware appliances to running on virtualized x86-based server platforms.

The move from hardware-based appliances to software is not new. Cisco developed the Nexus 1000v virtual switch, Brocade acquired the virtual router Vyatta, F5 release their “Virtual Edition”, CheckPoint provides a security gateway virtual edition, and many more. The move to virtual functions is also providing companies like VMware to natively integrate functions into the hypervisor and start-ups like PLUMgrid to offer overlay services that can stitch in their own functions, like routing, NAT, and DCHP or third-party functions like the F5 virtual load balancer.

The reason this has become so important is because of the formation of the ETSI ISG group for NFV. In 2012, 13 leading network operators banded together and produced a white paper outlining the need for virtualized functions as well as the integrated orchestration and management. The ETSI ISG for NFV has grown to over 150 participating companies. This shift toward software-based appliances will increase the ability to automate and program services, enabling providers to better respond to the changing needs of their customers. The group has recently released an updated white paper and several technical documents.

Network operators have a constant challenge to roll out new services and upgrade and maintain networks while keeping CAPEX and OPEX under control. Some of the pain points they face trying to deploy new services include:

- Variety of vendors and proprietary platforms to integrate
- Space, power, and cooling for all the equipment
- Skill set needed to design, test, deploy, and support networks
- Rapid product cycles requiring field upgrades

This shift toward software-based appliances will increase the ability to automate and program services, enabling providers to better respond to the changing needs of their customers.
NFV proposes that network functions, like routing, firewall, load balancer, and DPI would run on standard servers instead of expensive, proprietary appliances. By leveraging standard server virtualization technology, NFV implements network functions on software across a range of industry-standard server hardware. This provides the ability to easily instantiate, move, or change services without the need to install new equipment (assuming there are server resources available).

Operators are very serious about moving in this direction. The ETSI Industry Specification Group (ISG) for NFV published a white paper in October 2012 at the SDN congress in Darmstadt, Germany. More than 50 of the world’s leading operators are members of this group and contributing towards its efforts.

The ISG for NFV have created the following working groups to lead this effort:

- Architecture of the Virtualization Infrastructure
- Management & Orchestration
- Performance & Portability
- Reliability & Availability
- Software Architecture
- Technical Steering Committee
- Security

The architecture model proposed looks like this:

![Figure 1. High Level NFV Framework](image-url)
**Hardware Resources:** The proposal is to run on COTS (commercial off-the-shelf) hardware like Dell or HP servers with standard x86-based computing architecture. One concern is the speed of running functions in software; to address this CPU technology continues to advance. An example is the Intel Xeon 5600 once supported just 55Mpps, and now the new Xeon E5 v2 is nearly 5x faster at 250Mpps. This kind of speed is proving to be fast enough for networking applications.

**Virtualization Layer:** The virtualization layer is a hypervisor that provides virtualization of the underlying compute resources and enables many features like fast start/stop of the VM, snapshot, VM migration, and many others. The leading hypervisors are VMware (ESXi), KVM/QEMU, Citrix (ZenServer), Microsoft (Hyper-V). VMware has the leading market share in large enterprises, but due to the licensing cost, many providers are moving toward KVM.

**Virtual Layer (Virtual Compute, Storage, Network):** The virtual layer largely refers to the VM created by the hypervisor. This VM will have CPU, memory, and storage allocated to it. The VM will have its own operating system and software.

**Virtual Functions:** Virtual functions are traditional networking functions that now run in software instead of on vendor-specific appliances.

**Management and Orchestration:** The management at the highest layer is referred to as the Operations Support System/Business Support System (OSS/BSS). The orchestration layer needs to interact with the server and network infrastructure, and in most situations uses OpenStack (some may prefer the Citrix flavor CloudStack or VMware vCenter/vCloud Director). SDN controllers are typically also part of the architecture.

In the image below the ETSI ISG is also proposing architecture for service chaining and nesting, leveraging many virtual network functions:

![End-to-End Network Service](image)

**Figure 2. Examples of an end-to-end network service with VNFs and nested forwarding graphs**
Service Chaining: When multiple network functions are strung together, they are typically inter-connected by the vSwitch. Functions can also be nested instead of linear. The diagram above shows a more complex solution where functions can be stitched together even if they are in different physical locations. This can be achieved using various networking virtualization solutions including overlay (VXLAN, NVGRE, STT), SDN/OpenFlow, or other mechanisms.

This ETSI ISG is moving quickly to define the requirements driving vendor support. While this innovation is very exciting, it has to deliver on many key attributes to be widely adopted by the operators. Some of these include:

- Portability
- Interoperability
- Performance (understanding trade-offs)
- Migration from and co-existence with legacy services
- Management and orchestration
- Automation
- Security and resiliency
- Stability

When moving functions from physical appliances to virtual instances, there can be many trade-offs and design decisions that need to be made. Server resources like memory and CPU will be allocated to each function and it is important to optimize these resources for the application. NVF will likely change the network design, moving to a more distributed model for functions like firewall and DPI. Each of these will require test cycles to determine if moving to a virtualized function will meet the requirement of the service.

Ixia is an active participant in the ETSI ISG for NFV. We are working with the leading equipment manufacturers and service providers on the development of NFV and SDN. Ixia has a natural fit in this space since we continue to be the leading vendor for testing network technology and have offered virtualized solutions for several years. Ixia’s virtualized test technology, branded as the IxVM platform, enables testing to extend into virtualized servers. The IxVM platform supports Ixia L2-3 test application IxNetwork, as well as L4-7 test application IxLoad.

SDN and NFV

Current networking equipment runs on closed platforms that have specialized hardware with vendor-specific operating systems that run standards-based protocols. It requires years to enable a new protocol feature, which must make it through a standards body and then go through implementation by each vendor. Also, the primary way network devices are configured and managed is via command line interface (CLI). Networking technology has not kept pace with server technology in the data center and now there is a move toward COTS hardware and programmatic interfaces as seen in the server world.

SDN is decoupling of the control plane and data plane while enabling it to be directly programmable. This is a move away from distributed protocols (like BGP and OSPF) and more toward centralized control. Some of the initial work on SDN that was done at Stanford evolved into the definition of the OpenFlow protocol, which defines how an OpenFlow controller can program an OpenFlow-enabled switch. SDN has now split into many different protocol options and a few deployment models.
OpenFlow is the leading SDN protocol at this point, with support by more than twenty vendors including: Alcatel-Lucent, Big Switch, Brocade, Broadcom, Centec, Ciena, Cisco, Dell, Extreme, Freescale, IBM, HP, Huawei, Intel, Juniper, Metaswitch, NEC, Noviflow, NTT, Tallac, and VMware (OVS). In more of a pure OpenFlow model, referred to as an “underlay” implementation, all the devices would need to support OpenFlow and the connectivity end-to-end would be provisioned by an OpenFlow-based controller.

In an “overlay” implementation only the edge is programmable and the edge devices use tunnels (VXLAN, GRE, NVGRE, STT, etc.) to provide connectivity end-to-end over an existing network.

There are also “hybrid” deployments that can describe SDN being enabled on existing routing/switching products and/or being interworked with traditional routing/switching networks.

SDN has evolved into many different solutions being labeled as “SDN”. A few examples are:

- **VMware (formerly Nicira) NSX Solution** – this primarily enables the programmability at the vSwitch and establishes tunnels for end-to-end network connectivity. This has been a successful solution for virtualization and they have had customers such as AT&T, Fidelity, and Rackspace. They use propriety STT tunnels vSwitch to vSwitch and now also support VXLAN for tunnels to some hardware vendors (supporting VTEP).

- **NEC Controller and Programmable Flow Devices** – these are more of a pure OpenFlow-based solution, providing the ability to provision paths in the network. They have several success stories including Genesis hosting, which uses the technology to enable cloud hosting services.

- **Juniper Networks Contrail** – Is an overlay solution, which uses MPLS over GRE tunneling in the data-plane and XMPP for control plane signaling.

As you can see from the examples above there are many different options and deployment choices. There are also many different devices that support SDN, falling into three main categories:

- **Software Switches** – examples like OpenVSwitch (OVS) from VMware/Nicira or Switch Light from Big Switch are typically pretty full featured and low cost, yet due to their need to run in software may have limited forwarding performances. Also, if they are terminating tunnels, like STT or VXLAN, there can be a significant performance penalty. These are primarily used in an overlay deployment.

- **Existing Routing/Switching Platforms with Added SDN Protocol Support** – these include products from vendors like IBM, Brocade, HP, and others. Here you typically get limited features and sometimes-limited performance. These are useful in underlay or hybrid scenarios.

- **Hardware-Based Pure SDN Devices** – there are not many on the market yet, but companies like Netronome, Noviflow, Intel, and others are producing hardware that is optimized for SDN. These are typically more expensive than standard Broadcom-based switching hardware. There is one other class here - commodity switches like the Pronto by Pica8, which provide the ability to load and run third-party SDN software. They can be limited in feature-set, including the lack of traditional features (like SNMP).

OpenFlow is the leading SDN protocol at this point, with support by more than twenty vendors including: Alcatel-Lucent, Big Switch, Brocade, Broadcom, Centec, Ciena, Cisco, Dell, Extreme, Freescale, IBM, HP, Huawei, Intel, Juniper, Metaswitch, NEC, Noviflow, NTT, Tallac, and VMware (OVS).
Implementation of OpenFlow have progressed to v1.3, yet it still lacks a lot of functionality as well as operational features. Some of these limitations include:

- Lack of a standards-based North-Bound API (work on this is started now within ONF)
- Lack of OAM (some work is started in ONF in the config-management workgroup)
- Lack of management (no equivalent of SNMP, Ping, Trace, etc.)
- Complexity due to many optional features and pipelining (multi-table) implementations

How are SDN and NFV Related?

A good visual representation of the relationship between SDN and NFV looks like this:

![Figure 3. Visual representation of relationships](image)

The end goal is to optimize existing services or build new revenue-generating services.

Services can be built directly using NFV without SDN and services can be built with SDN without NFV. However, the two are very complementary. SDN can be used to provision the network connectivity to the virtual function. This enables the ability to build end-to-end services with network connectivity and enhanced functions that have been virtualized. This works for both “overlay” (using SDN at the edges over non-SDN networks) and “underlay” (using a fully SDN-enabled network) architectures.
Real-World NFV

Is this technology real, or is it many years away?

- In 2013 AT&T launched the Domain 2.0 project. As a part of this the “beachhead” project is to virtualize the evolved packet core (EPC). They have selected Affirmed Networks for this project and will start deploying this year. At a recent event they announced they will use SDN and NFV to create a “user-defined network cloud.” This will be a multi-service, multi-tenant platform that “taps into NFV and SDN to perform a broad variety of network functions and services”

- British Telecom (BT) has several projects underway including:
  - Tested and successfully deployed a virtualized BRAS (vBRAS) solution
  - Tested and deployed virtualization of IPSec tunnel termination for Wi-Fi and LTE services as part of their FON offering
  - Tested virtualizing content distribution networks (CDNs) using Akamai Aura HyperCache
Colt has several projects underway to leverage NFV and SDN

- Internal projects include L3 CPE virtualization, and WAN SDN in the Data Center fabric
- External projects include “Colt Live” offering for the media sector, “elastic bandwidth” offering for infrequent applications, “Service Console and Dashboard” for enhanced self-service and reporting, “WAN as a service” driven by self-directed moves/adds/changes to a service

Deutsche Telecom (DT) also has several projects including:

- Tested and deployed the new Terastream Architecture in Croatia (based on NFV and SDN), which virtualizes DHCP, IPv4, and VPNS. Built based on ALU, Cisco, and Combis.
- Tested using a distributed BRAS/MBG function performing authentication with SDN
- Project to move the set-top-box to the cloud and provide a dongle to customers to access services from the cloud

NEC has developed several virtualized solutions including: virtual EPC, virtual BRAS, and virtual CGNAT

- The vEPC solution became available in 2013 and has been commercially deployed and is in several other trials

These are just a few examples. A recent survey from Infonetics (Infonetics Research 2013: “SDN and NFV Strategies: Global Service Provider Survey,” July, 2013) indicates that 82% of providers are beginning to evaluate NFV and SDN.

Testing NFV to Ensure Carrier-Grade Service Delivery

When evaluating performance of a virtual function, there are four key types of work-loads:

- **Data Plane** – These are tasks associated with packet handling that involve input/output operations and read/write memory operations.
  - In terms of NFV, this will vary by function. A virtualized router, for example, will need to perform route look-ups, forwarding packets, adding/removing headers and other features enabled on the virtual router.
- **Control Plane** – These are tasks related to protocol exchanges including setup, session management, and termination.
  - For a virtualized router this would include protocols such as BGP and OSPF. A virtualized BRAS would have PPP session management and RADIUS authentication among other functions. Control plane is normally more resource intense on CPU resources and lower on I/O and R/W operations.
- **Signal Processing** – These are tasks associated with digital processing and are expected to be highly resource intense on CPU and delay-sensitive. An example is the FFT decoding and encoding in the C-RAN Base Band Unit.
- **Storage** – These are all workloads associated with reading and writing to disk storage. There are non-intensive operations like logging and more intense operations like a network probe that requires a high volume of disk write operations.

When testing, there are many factors that impact performance. It is important to establish performance baselines before adding complexity and determine what may be creating performance bottlenecks through isolation. A step-wise process would involve:
1. Run tests on bare metal (without virtualization) first to identify any bottlenecks introduced by the hardware, the host OS, or the software implementation of the virtualized function. Iterate through changes (changing only one thing at a time) to determine the performance impact. Some tuning usually available from the hardware and host OS, and sometimes the function itself.
   a. Features that will affect the performance when running on bare metal:
      • Processor architecture
      • Extended instruction set (req. for crypto)
      • Clock rate
      • Size of data caches
      • Memory access speed
      • Memory latency
      • Inter-processor bus bandwidth
      • Number of cores on a processor
      • Large page support
      • I/O page support
      • TLB cache with large page support
      • I/O TLB cache with large page support
      • Size of TLB caches
      • Interrupt affinity
      • Layered memory cache
      • Deterministic allocation of threads in CPU
      • Deterministic memory allocation
      • Independent memory structures per thread
      • Inter-thread communications through memory pipeline structures
      • CPU isolation
      • DMA
      • Direct I/O access to processor cache
      • Flow affinity/steering by I/O devices
      • NIC acceleration capabilities

2. In the second phase, progress to the virtualized environment. Once virtualized, there will be an initial configuration of processor (CPU) and memory allocated to the VM. Again, iterate by only changing a single variable at a time to determine the performance impact.
   a. Features that will affect the performance when running virtualized environment (some of these may be supported and configured at the guest OS level)
      • Processor architecture
      • Extended instruction set (req. for crypto)
      • Clock rate
      • Size of data caches
      • Memory access speed
      • Memory latency
      • Inter-processor bus bandwidth
      • Number of cores on a processor
      • Large page support
• I/O page support
• TLB cache with large page support
• I/O TLB cache with large page support
• Size of TLB caches
• Interrupt affinity
• Layered memory cache
• Deterministic allocation of threads in CPU
• Deterministic memory allocation
• Independent memory structures per thread
• Inter-thread communications through memory pipeline structures
• CPU isolation
• Polling mode drivers
• DMA
• Direct I/O access to processor cache
• Flow affinity/steering by I/O devices
• NIC acceleration capabilities
• Instructions to reduce the number of VM exits under certain common operations
• Second-level address translation services
• Second-level address translation services for large pages
• Second-level address translation services for I/O
• Second-level address translation services large I/O pages
• I/O interrupt remapping
• Extension of processor caches with new fields to avoid cache eviction with VM exits
• Extension of processor TLB caches with new fields to avoid TLB flushes
• Independent TX/RX queues for virtual machines
• SR-IOV

When testing, it is important to define the performance objectives and determine if they can be achieved.

Test Scenarios

When testing a virtualized function, there are two primary approaches:

1. Map the function to physical NIC interfaces and test with traditional hardware-based test systems
2. Test the function in the virtualized system by inserting testing into the virtualized system (primarily through running on a VM). This can also include a combination of virtual and physical test ports
Example of Testing L2/3 Virtualized Functions

Figure 5. L2/3 test example

Testing L2/3 functions that are mapped to physical NIC interfaces has many of the same attributes of traditional testing:

- Forwarding performance (loss, latency, throughput)
- Protocol performance and scale
- Multi-protocol/multi-dimensional testing
Example of Testing L4-7 Virtualized Functions

Testing L4-7 functions that are mapped to physical NIC interfaces has many of the same attributes of traditional testing:

- Application performance (loss, latency, throughput)
- Session scalability
- Threat/security testing
Example of Inserting Testing into the Virtualized System

Test cases can also require a combination of physical and virtual ports for certain test scenarios.

Figure 7. Inserting test example

Note – test cases can also require a combination of physical and virtual ports for certain test scenarios.

Testing virtually by inserting test interfaces into the virtualized server:

• Test the vSwitch for performance
• Test each of the virtual appliances
• Test virtual appliance chaining

Additional variables that apply to testing a virtualized function include:

• Determining the optimal resources (CPU/memory) allocated to the virtual appliance to meet the performance requirement
• Instantiation of a service – how fast
• Termination of a service
• On-the-fly changes to a service
• Moving (VM migration) of a service
• Reliability of a service
• Service isolation (affected by other VMs or services on the server)
IxVM

IxVM virtual ports provide a software-based version of a traditional Ixia hardware port that can be deployed in a virtualized environment. The IxVM platform requires no specialized Ixia hardware and can run on general-purpose, off-the-shelf servers.

![IxVM Diagram](image)

**Figure 8. IxVM provides a software-based version of traditional Ixia hardware ports**

The IxVM platform supports the two mainstay Ixia applications, IxNetwork and IxLoad. The capabilities and workflow of these applications are the same on the IxVM ports as they are on hardware ports. The platform is an ideal solution for our existing customers looking to test their products early in the development cycle and also for customers looking to build their SDN and NFV strategies. IxVM provides the perfect test platform to meet these needs as it operates in a pure software environment and does not require and custom Ixia hardware.

**IxVM – Network**

IxNetwork with IxVM ports gives our customers on the virtual platform access to all the protocol emulation available on our hardware platform. With support for protocols like BGP, MPLS VPNs and OpenFlow, IxNetwork, and traffic generation capabilities (up to 10G), IxVM helps users validate the performance and scale of their SDN/NFV deployments.

**IxVM – Load**

IxLoad with IxVM ports gives our customers the ability to generate true stateful application traffic like HTTP, DNS, and SMTP at high loads. The NFV initiative is driving the needs for NEMs to virtualize functions like Firewalls and SLBs. With support for hundreds of stateful applications, IxLoad VM can help address the needs to validate and test these network functions at scale.

The IxVM platform is an ideal solution for our existing customers looking to test their products early in the development cycle and also for customers looking to build their SDN and NFV strategies.
RackSim

Server and storage virtualization technology has revolutionized the data center in recent years. With the introduction of SDN, cloud service providers (CSPs) can now leverage even greater benefits from their data centers. New categories of data center switching/inter-connect devices are also emerging. These devices are capable of connecting servers, virtual assets, and networks, as well as storage devices.

Ixia RackSim is a "first-of-its-kind" test solution that emulates such a virtualized data center environment with all of its complexities. A single RackSim appliance can simulate a rack full of servers/hosts with various hypervisor flavors.

The emulated hypervisors can be set up to host simulated virtual machines with VM events, in addition to generating traffic from the simulated VM. Using RackSim, a user can verify the functionality of their data center assets, test-drive massive configurations involving virtualized assets, and model a variety of environments.

A single RackSim appliance can simulate a rack full of servers/hosts with various hypervisor flavors.

Figure 9. RackSim simulates a rack full of servers/hosts with various hypervisor flavors

The emulated hypervisors can be set up to host simulated virtual machines with VM events, in addition to generating traffic from the simulated VM.

Using RackSim, a user can verify the functionality of their data center assets, test-drive massive configurations involving virtualized assets, and model a variety of environments.
The key highlights of the RackSim solution include:

- Emulation of a large number of virtualized server hosts running various hypervisors like VMware and KVM/QEMU
- Realistic provisioning of the VMs
- Support for third-party VM Managers
- Support for realistic VM events like deploy, start, stop, destroy, and migrate
- True subscriber emulation per VM with support for application traffic flows
- VM migration scenarios at high scale to verify disaster recovery effectiveness

**IxChariot**

IxChariot is the industry’s leading test tool for simulating real-world applications to predict device and system performance under realistic load conditions. Comprised of the IxChariot Console, Performance Endpoints, and IxProfile, the IxChariot product family offers thorough network performance assessment and device testing by simulating hundreds of protocols across thousands of network endpoints. IxChariot provides the ability to confidently assess the performance characteristics of any application running on wired and wireless networks. The IxChariot endpoints work on almost any operating system and are Ixia’s most portable independent software platform.

**Phantom Virtualization Tap**

Like a magnifying glass, the landmark Phantom Virtualization Tap exposes the true scope of your virtualized infrastructure. You see every bit of traffic passing between VMs on hypervisor stacks, as well as between VMs residing on separate blades in the same chassis. It taps traffic between your VMs on a physical server and enables rigorous, best practices network management and control.

The Phantom Virtualization Tap integrates directly into the hypervisor kernel, enabling full access to the entire network stack without the performance penalty of running your vSwitch in promiscuous mode. Elimination of promiscuous mode improves virtual machine and physical resources-relief that translates into improved performance, greater throughput, and resource optimization. You avoid the loss of important network-layer errors that may be cleaned off before sharing (and thus be invisible to other monitoring resources), when those very errors are key to successful troubleshooting of your issue.

**SDN-Related Features**

SDN is another great example of Ixia’s drive to be on the bleeding edge of networking technologies and innovation. We were on the charter Testing and Interop working group since its inception. We already have commercial solutions like IxANVL and IxNetwork that allow our customers to validate conformance and performance of the latest OpenFlow standards.

For more information on NFV and NFV test solutions go to:

http://www.ixiacom.com/nfv

Note: much of the NFV technical information in this document is from ETSI ISG documents, including ETSI NFV Architecture and NFV Performance & Portability Best Practices, located at http://portal.etsi.org/