Networking in the Era of Virtualization

Compute virtualization has changed IT's expectations regarding the efficiency, cost, and provisioning speeds of new applications and services. By decoupling the operating system from the underlying hardware, compute virtualization provides an extremely flexible operational model for virtual machines that lets IT treat a collection of physical servers as a generic pool of resources.

Compute virtualization provides a template for how to think about IT infrastructure more broadly. That is, all data center infrastructure, including networking, should provide the same properties as compute virtualization. If that goal can be achieved, it will unlock a new era of computing more significant than the transformation from mainframe to client-server.

The advantages of this shift bear repeating: once infrastructure is fully virtualized, any application can run anywhere, allowing for operational savings through automation, capital savings through consolidation and hardware independence, and market efficiencies through infrastructure outsourcing models such as hosting and "bursting."

Unfortunately, data center infrastructure is not yet fully virtualized. While the industry has decoupled operating systems from servers, it has not yet decoupled the operating system from the physical network. Anyone who has built a multi-tenant cloud is aware of the practical limitations of traditional networking in virtualized environments. These include high operational and capital burdens on the data center operators, who run fewer workloads, operate less efficiently, and have fewer vendor choices than they would if their network was virtualized.

Fortunately, the industry is moving quickly towards fully virtualized networks. Nicira works with the operators of many large virtualized data centers and understands the scope of the problems posed by legacy networks. The company is working with the architects of these data centers to develop and deploy a new approach to networking, a distributed virtual network infrastructure (DVNI), which addresses the shortcomings of traditional architectures and brings the benefits of virtualization to the network.

To understand the concept of a DVNI and how it provides a complete virtual networking solution within a virtualized data center, it's helpful first to examine why network virtualization is needed and why traditional networking approaches can't support virtualization.

nicira

The Need for Network Virtualization

A fully virtualized data center should maintain all of the properties IT has come to expect from virtualization. Unfortunately, achieving these goals is extremely difficult with traditional network designs, and is complicated by the interplay of multiple technologies. Often a limitation can be addressed with a point solution that then has a deleterious effect elsewhere in the architecture. For example, mobility limitations can sometimes be handled by running a router in a VM, but this reduces the virtual network capacity to the throughput of a single virtual machine, which is impractical in most production environments.

The following are common issues that arise when using traditional approaches to networking in virtualized data centers. While some of these problems have available solutions, in aggregate they are considered the primary hurdles to building modern virtual data centers. They include:

Reliance on hardware for provisioning

With compute virtualization, any VM can run on any standard server, and the provisioning and management of that VM can be done fully automatically. Unfortunately, due to the way their virtual data centers are constructed, many organizations don't enjoy this freedom. Rather, the creation of isolated networks for VMs (and their associated network policy) is done by configuring hardware, often manually and through vendor-specific APIs. There are multiple problems with this approach – data center operations are now tied to a particular vendor's hardware and, if configured manually, rely on an error-prone process. In addition, if the network is enforcing the network policy, it is difficult to replace or upgrade the network hardware in a non-disruptive way.

Lack of address space virtualization

Commonly, VMs reside in the same IP address space as the physical network. For example, a VM's first hop gateway is configured to be a physical router somewhere in the network. There are two classes of problem that arise under such configurations.

First, VMs that share the same L2 network are confined to that physical subnet, limiting mobility and VM placement. In many of the largest virtualized data centers, the inability to place a VM anywhere within the data center (or between data centers) makes it very difficult to accommodate tenants as they grow. Either all of a tenant's VMs cannot be on the same IP subnet, a tenant cannot grow, or all the VMs must be moved to a pod with enough space, resulting in downtime and fragmentation of compute resources.

Second, because the same forwarding tables are being shared (whether L2 or L3), it is difficult to support overlapping IP addresses. Multi-tenant environments, in particular ones that host third-party workloads, ideally should be able to support any IP address desired by the customer. This allows a much simpler migration path to the virtualized data center, as the customer premise can be bridged into a virtual hosting environment without re-numbering IP addresses. Today, virtual routing and forwarding (VRF) table limits and the need to manage NAT configuration make it cumbersome to support overlapping IP addresses or impossible to do at scale.

Network services tied to hardware design cycle

If the physical network implements the network services, then the ability to add new services is limited by the availability of new hardware and IT's ability to deploy it. Hardware feature development cycles, which tend to be limited by ASIC design, can be anywhere from 18 months to 4 years. Because of these long development times, organizations that operate the largest virtual data centers don't rely on the physical hardware for virtual network provisioning or virtual network services. Instead they are using software-based services at the edge, which allows them to take advantage of faster software development cycles for offering new services.

Scale

Virtualization is putting pressure on traditional networking's ability to handle scale. The canonical example of this is VLANs. Traditional VLANs are limited to 4,096 independent segments. Many large clouds have far more customers than this, requiring them to segment their operations to use multiple, non-overlapping networks each supporting its own VLAN space. In addition to this VLAN limitation, there are many other areas in which traditional networking cannot handle the scaling requirements of modern data centers.

For example, a typical multi-tenant virtual data center hosts between 20 and 80 VMs per hypervisor. This high level of consolidation requires that the physical network (if the network is not virtualized) manage addresses and policy for many more hosts than in traditional data centers. As a result, MAC and ACL table exhaustion is a real issue in large data centers. Even moderately-sized virtualized data centers require higher-end (and thus more expensive) switches to accommodate larger L2 table sizes. A preferable solution is to not expose the virtual addressees or policy to the physical network, which would reduce the demands on table sizes significantly.

Integrating network services

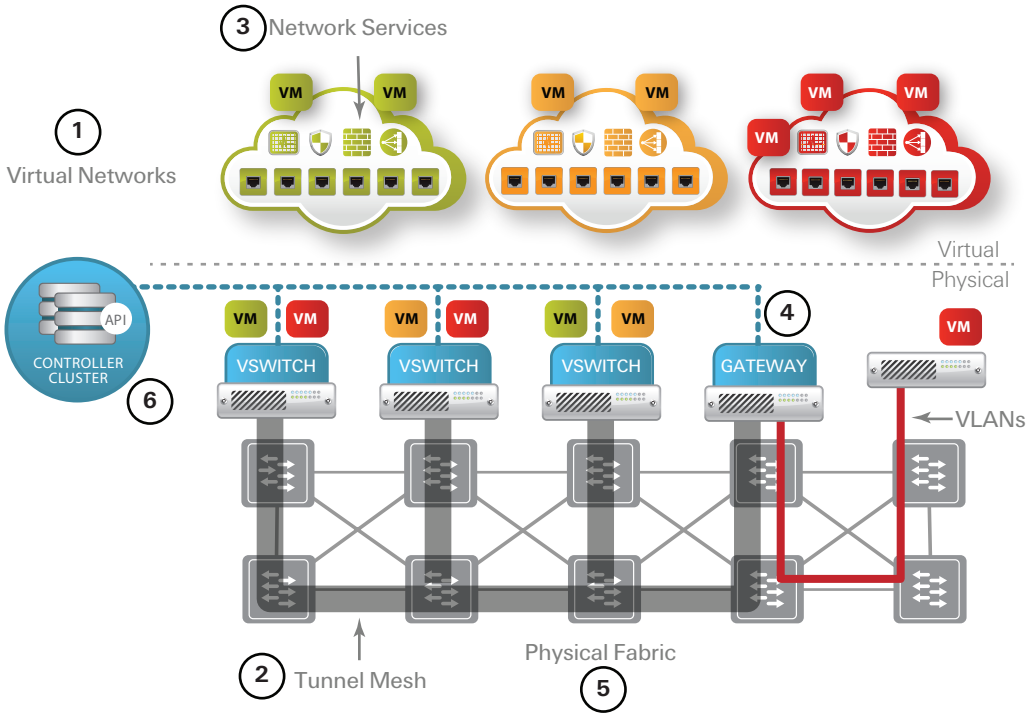
Services such as load balancing, firewalling, and WAN optimization are an integral part of networking and therefore must be available in some form for workloads running in virtual data centers. There are two issues that must be addressed when integrating virtual services into a virtual data center. The first is ensuring that the traffic is routed to the appropriate services. A common way to do this is to use VLANs; however, this results in all of the shortcomings discussed above, and requires L2 adjacency between the VM and the middle-box providing the service.

The second issue is managing to scale the available bandwidth and service capacity as needed by the virtual network. If a virtual network's traffic is being routed to a single choke point (say, a virtual appliance acting as a load balancer), it is possible for the bandwidth requirements to exceed the capacity of that single point, in which case the service itself must also be virtualized to provide scale-out capacity that grows with the needs of the virtual network.

Distributed Virtual Network Infrastructure (DVNI)

Rather than try to work around the shortcomings of traditional networks, some of the more aggressive virtual data center architects and operators have turned to alternate approaches that are more suited to the problem space. While the architecture presented here is unique to Nicira, this general approach is being adopted throughout the industry by network vendors, hypervisor vendors, cloud management system vendors, and data center architects themselves.

With a Distributed Virtual Network Infrastructure (DVNI), the virtual network is implemented along the network edge (generally within the vswitch) and abstracted from the physical network via a dynamic mesh of L3 tunnels. A good DVNI solution will provide virtual networks that support any MAC address, any IP address, any network service model (for example L2, L3, multicast, etc.), and higher level services such as NAT and load balancing. And it will support all of this over any physical network, such as a simple IP fabric, without introducing choke-points or requiring any special support from the network hardware. A good DVNI solution should also allow VMs to be placed or migrated anywhere within or between data centers.



1. The virtual network

The virtual network is the basic abstraction maintained by the DVNI architecture. It is what VMs connect to, in order to communicate, and what administrators or cloud management systems use to monitor and manage the network. A virtual network is fully decoupled from the underlying topology. That means that the network services supported at the virtual network layer do not need to be supported by hardware, and the topology and configuration of the physical network remains unchanged even during changes in the physical network topology (for example, during VM migration).

To the administrator or orchestration system, virtual networks have the same operational model as VMs: they can be created, grown or shrunk dynamically; they can each have different configurations; and they can be moved anywhere within a data center or between data centers without disrupting the virtual view. The virtual network is implemented at the edge in a distributed manner, meaning that the edge only implements the functions relevant to its directly-attached VMs or physical machines (this is described in more detail later in the paper).

2. The tunnel mesh

All virtual traffic is transported over the physical network using L2 in L3 tunnels. The encapsulation hides the internal traffic from the physical infrastructure, relieving network devices from having to handle the additional addresses and dynamic complexities of the virtual networks. Tunnels are generally terminated in the vswitch for virtual workloads, and top-of-rack switches or gateways for physical workloads.

The particular tunneling protocol is unimportant to the architecture. There are many encapsulation formats, including GRE, NVGRE, VXLAN, and CAPWAP. For maximum compatibility, a network virtualization solution should support as many as possible.

The topology of the tunnels is not fixed within the architecture. Scaling is often a factor in determining how the tunnels are arranged. vswitches, for example, can support tens of thousands of tunnels without affecting performance and therefore can support a full mesh (every hypervisor connected to every other hypervisor) even in very large networks. Hardware, on the other hand, often has tunnel limits, in which case a multi-hop topology must be implemented.

In addition to carrying virtual L2 frames, tunnels carry any additional information needed to implement the virtual network abstraction. This includes information specifying the virtual network, the state of the virtual network (for example, what lookups have already been done), and any other information needed to reconstruct the virtual environment.

With respect to performance, modern tunnel implementations within the hypervisor vswitch can operate at full 10Gbps line rate with low CPU overhead (40% of a single core in the case of Nicira's solution).

3. Network services

DVNI pushes lookup within the virtual network to the edge; however, where that computation actually occurs isn't fixed. For example, the first hop vswitch may perform the L2, L3, and ACL lookups of the virtual network. Or, that lookup may be implemented partially on the first hop vswitch, and partially on the last hop vswitch. In some cases, it may be preferable not to do the computation within the hypervisor at all. In this case, the network services are implemented as standalone appliances which are stitched into the tunnel mesh. There are multiple methods of implementing network services; the appliances themselves can offer stand-alone services, or they can be generic compute pools that can run any number of virtualized services.

4. Gateways

A gateway is an "on and off" ramp into the virtual world. That is, a gateway is the point at which a packet that was traversing a virtual network gets placed on the physical wire and is out of the control of the DVNI solution. Gateways can map a packet to a physical port, or to another VPN or encapsulation technology such as VLAN or MPLS. Good DVNI implementations will support load balancing and failure across multiple gateways to ensure that a single point of failure has not been introduced into the system.

5. Physical fabric

The only requirement that a DVNI solution has of the physical network is IP connectivity. Therefore it is compatible with any L2 or L3 data center network design. However, to avoid constraining VM placement or performance bottlenecks, the network should be only lightly oversubscribed and each element in the network should be able to enforce QoS decisions based on standard packet markings.

6. Controller cluster

The controller(s) manage the state at the edge of the network and expose an API for provisioning, configuration, and monitoring the virtual networks. A robust DVNI solution will implement the controllers in a distributed manner, allowing for non-disruptive failure of nodes within the cluster.

DVNI, SDN, and Fabric

DVNI aligns well with two dominant trends in computing, the move towards a data center fabric and software-defined networking (SDN). Each is described in more detail below, along with how they relate to DVNI.

Software-defined networking

SDN is a term used for networks in which the control plane is decoupled from the data plane. The rationale for SDN is that it is difficult for system builders to implement complex control functions in a purely distributed manner, which is what traditional networking requires.

With SDN the system builder can choose the distribution model that best fits the problem at hand. For example, a tightly coupled, homogenous cluster of servers is far more amenable to implementing sophisticated control than the pure distribution model required by traditional networking. Therefore, with SDN, a system builder can draw on distributed system techniques developed over the last decade within companies such as Google, Amazon and Facebook and apply them to networking to build both very scalable and very sophisticated network functionality.

In the context of DVNI, SDN is used for managing the complex state mapping that happens at the edge of a DVNI network.

The move towards fabric

Data center networks are converging on simple, non-oversubscribed fabrics that deliver consistent latency between any two ports. There are many competing technologies for building large fabrics. A common approach used by many of the largest data centers in the world is to build L3 networks (using, for example, OSPF) with a non-oversubscribed physical topology. The traditional network vendors also have fabric offerings that provide various features, such as flat L2 with any VLAN on any port, although they generally are built around proprietary protocols.

DVNI separates the concern of forwarding in a dense physical topology (the fabric), and the management of the virtual abstraction, which contains the network configuration and policy. This separation allows each to evolve independently using the architectural approach that is best suited for the problem space. For example, a DVNI solution can be implemented over an existing data center network non-disruptively today, without changing any of the hardware or configuration. As needs change, the physical network could then be upgraded to a newer fabric design without requiring any change to the virtual network layer. Similarly, because the virtual network layer is implemented primarily in software, new features and services can be introduced quickly without requiring any change or upgrade to the fabric.

The Anatomy of a Virtual Data Center

While approaches to building virtualized data centers vary, most conform roughly to the same approach. From the perspective of network function, the primary components of a virtualized data center are:

Physical network (fabric)

The physical network is simply the networking gear that provides data center connectivity. In virtualized environments, it is important that the network not constrain workload placement, otherwise it is difficult to achieve high compute efficiencies. To achieve this, many organizations are moving towards a network fabric architecture in the data center. A fabric is effectively a network that has little or no oversubscription, has similar latency between any two ports, and often is managed as a single entity. Very effective and economical fabrics can be constructed using L3 and a non-oversubscribed multi-stage physical topology, such as a CLOS network. While, many vendors provide proprietary fabrics, organizations with the largest data centers are trending towards building simple L3 fabrics using only standard protocols such as OSPF or ISIS.

vSwitch

The vswitch is a soft switch within each hypervisor. It manages switching on the server between virtual machines, and can enforce network policies, such as basic quality of service (QoS) and filtering on packets sent from the server to the physical network. Vswitches are often configured by a controller which manages the network policy globally and updates the policy as VMs join, leave, or move within the network.

Virtual network (VN)
A virtual network is an isolated connectivity domain that ties together a set of VMs. Often these are implemented using VLANs (one per virtual network), ACLs, or per-VN overlays.

Network services
Network services such as firewalls and load balancers are generally available as either dedicated hardware appliances or virtual machines. These services are often integrated into a virtual environment using VLANs or tunnels.

Orchestration system
The orchestration system is responsible for orchestrating all of the resources in a virtualized data center, including compute, storage and networking resources. OpenStack is one example of an orchestration system which is gaining considerable momentum. In relation to networking, an orchestration system can be used to configure vswitches or the physical network directly, or to communicate with a controller that manages the network portion.

Bare-metal workloads
Often VMs have to be integrated with workloads running on bare metal. There are many reasons for this; for example, integration with legacy environments, and the desire to provide a higher-performance configuration than is offered by virtualization. The most common use cases for integrating physical workloads with a virtualized environment are tying together physical and virtual hosting in a multi-tenant cloud, bridging a customer LAN into a virtual cloud, and sharing a physical server that's not suited for virtualization.

Implementing DVNI

Moving to a virtualized network infrastructure should be as non-disruptive as possible. When evaluating DVNI solutions, it's important to consider the following issues:

Resiliency
DVNI introduces new components into the network, the controllers, the service nodes, and the gateways. A reasonable and common concern is whether or not these change the failure characteristics of the data center. If implemented correctly, DVNI provides high availability throughout the architecture. The controllers, for example, should be implemented as a fully distributed system that can handle arbitrary failures of any non-majority of nodes. For components on the data path, it is both important that a single failure does not disrupt operations, and the data path handles convergence after failure quickly. For this reason, both gateways and service nodes should be implemented in a fashion that failure results in quick failover on the data path.

Enforcing QoS within the fabric
In a DVNI context, the intelligence and first line of enforcement for QoS policy resides at the edge. From that vantage point, it is possible to enforce virtual guaranteed minimum and maximum bandwidth allocations for the virtual network, and any prioritization during contention at the access link. To deal with QoS in the fabric, a DVNI solution can mark packets using standard header bits on the outer tunnel, allowing the physical fabric to enforce priorities for any link under contention.

Load balancing within the fabric
To provide full bisectional bandwidth, a network fabric generally load balances network flows over multiple paths (for example, in L3 fabrics, ECMP is used). In a DVNI architecture, virtual network flows are tunneling within much more static flows, and thus the per-flow fidelity is lost to the load balancing mechanism. As a result, it is important that the solution expose the flow information in the outer header. This can be done by modifying any of the header bits that can be used by the ECMP hash algorithm in the fabric. Common headers used for this purpose are the transport source port or the GRE key.

Security
A correct implementation of DVNI should not change the security model of a virtualized environment, and should provide security guarantees not offered by traditional networks. The basic security model of DVNI matches that of compute virtualization, which is trust consolidation in the hypervisor. The vswitches, which run in the hypervisor, are assumed to be trusted entities and they enforce all security policy, including address isolation, QoS, and higher level

security services. The vswitches themselves are strongly authenticated with the controllers, which are typically run on an out-of-band network, so they are not addressable from the guest operating systems. In addition, a DVNI architecture hides the physical network from the guest VM, thereby removing it as an attack target.

Troubleshooting and monitoring

Virtualization has a direct impact on troubleshooting and monitoring of a DVNI implementation. Instead of having to monitor a single physical network, an operator must now monitor the physical network plus all of the virtual networks. There are three areas in which a DVNI architecture can aid the operator.

First, each virtual network should implement a standard management interface, such as SNMP, so that the operations and management toolset used by the administrator can be applied to both the physical and virtual domains. Second, with the tunnel overlay, it is possible to monitor the end-to-end health of the fabric. This provides a level of visibility into the fabric that doesn't exist today. Finally, a DVNI implementation should provide the ability to map from the virtual world to the physical world and back. This cross-layer visibility allows operators to map from a virtual problem to a physical source, presuming one exists.

Because each packet is encapsulated by a trusted entity (the vswitch), another advantage DVNI offers for management is that any packet within the network can be traced back to the original sender, regardless of whether the sending operating system changed addresses or is sending forged packets. In order to fully realize a virtual pipeline, a good DVNI implementation will have enough context within the tunnel header to map the packet back to the sending VM.

The Benefits of a DVNI Approach

Because it decouples the virtual networks from the underlying hardware, DVNI delivers a host of benefits, including:

Full address virtualization

DVNI fully decouples the virtual and physical address spaces. This allows VMs unrestricted placement and mobility anywhere in the world. (Mobility may be limited by the end-hosts' ability to handle latency within the virtual network. For example, many L2 protocol stacks are built assuming LAN-level latencies and do not operate correctly over long distances). It also allows overlapping addresses between virtual networks, the ability to bridge or migrate physical networks into virtual networks without renumbering IP addresses, and it allows multiple addressing mechanisms (for example IPv6) to run over standard IPv4 gear.

Full L2/L3 virtualization

With DVNI, it is possible to faithfully reproduce the physical network service models such as L2, L3 with support for broadcast, multicast, flooding, and standard ACL and policy controls. The virtual service model is independent of the underlying network, allowing for multiple virtual networks to support different service models (e.g., some L2 and some L3) and for the service model to be distinct from the underlying networks (e.g., L2 virtual service model over an L3 network). It is also possible to insert L4-7 services such as NAT, stateful firewalling, and load balancing into the virtual network without requiring any additional hardware.

Services as software

Implementing the virtual networking functionality at the edge of the network allows services to evolve quickly with market needs by leveraging the rapid innovation and release cycle for software. Further, if implemented correctly, these services can be added, grown and configured dynamically without requiring any manual configuration or rewiring of the physical infrastructure.

Scale and performance

A DVNI solution can be overlaid on any data center fabric design without imposing any additional scaling hurdles, such as table state explosion or additional control overhead. Because DVNI implements everything at the edge in a distributed fashion, there is little to no data path performance degradation, and no additional choke points.

The primary challenge with scale for a DVNI implementation lies within the controllers. They are responsible for managing the state mapping at the edge, between the virtual and physical address space. There may be thousands of forwarding entries and tunnels at thousands of end points. In order to truly scale and provide resilience to failure, the controllers should support distribution of the work needed to control all of the vswitches.

Decoupling from physical hardware

Under DVNI, any virtual network should be able to run on any physical hardware. From an operational viewpoint, this allows the physical network to be upgraded or replaced without significant disruption to the virtual networks. As previously noted, it also allows for the introduction of new services on software time scales. From a capital viewpoint, this decoupling allows a data center operator to evaluate physical network solutions on price-performance rather than value-added features implemented in hardware.

Achieving Network Virtualization

Like computing, networking must evolve a virtualization layer that decouples workloads from the physical network. Until this happens, the full potential of compute virtualization will remain unrealized. Traditional networking approaches are not well suited for this task.

Distributed Virtual Network Infrastructure (DVNI) provides a network virtualization architecture that addresses the shortcomings of traditional network approaches, providing a host of virtualization benefits, such as isolation, mobility, scalability, dynamic provisioning without restriction, and hardware independence. As a result, DVNI is taking hold in the world's largest virtualized data centers.

About Nicira

Nicira is the network virtualization company. Nicira's Network Virtualization Platform (NVP) enables the dynamic creation of virtual network infrastructure and services that are completely decoupled and independent from the physical network hardware. Innovative companies such as AT&T, eBay, Fidelity Investments, NTT and Rackspace are using Nicira NVP to accelerate service delivery from weeks to minutes and dramatically reduce complexity and cost. For more information, please visit www.nicira.com

Nicira 3460 West Bayshore Road Palo Alto, CA 94303 U.S.A.
Tel +1.650.473.9777 Fax +1.650.739.0997 info@nicira.com www.nicira.com