cloudscaling

SDN & Cloud Networking

Introduction

Cloud computing offers a truly revolutionary paradigm shift in the model for how IT services are consumed and managed. Much more than a set of technologies or even an outsourcing model, cloud computing provides a new way for IT to "do business." This white paper explores how to deliver agile, cloud scale networking for private and hybrid clouds.

As a technical guide, we make an attempt to strike a balance between a purely academic resource, a pragmatic reference, and enough business information to allow a broad range of readers to derive value. Therefore, the target audience for this document is assumed to be network architects, cloud engineers, CTOs, and technical business leaders looking to understand how to deploy agile and scalable private cloud networks.

Some information here is intentionally simplified to reach a broader audience, but there is more than enough detail to point experienced network engineers and architects in the right direction when trying to understand how to deploy cloud-scale IaaS networks.

Executive Summary

Cloud networking is not a trivial task. The Internet itself is network infrastructure designed to run at massive scale. In its history and evolution, a large number of challenges were overcome to run a global network of nearly ten billion connected devices. Modern data centers designed to provide next generation private cloud offerings face challenges similar to building the Internet itself. This is because even private clouds must ultimately be designed to scale to a reasonable size and to be flexibly configurable.

Building a data center network that can efficiently manage thousands of networked virtual servers still means using techniques similar to those used in large ISP backbones and the Internet. In fact, many of the largest cloud providers, such as Amazon, Google, Yahoo!, and Facebook, prefer Layer 3 (L3) networking techniques over typical Layer 2 (L2) networking techniques because of their proven scalability. L2 networking models, prevalent in enterprise data centers, are known to have significant scaling problems.

In addition to the need to network massive numbers of physical servers together, the rise of hardware virtualization and distributed systems means that the number of connected devices in a data center, particularly those of Infrastructure-as-a-Service (IaaS) clouds, is growing exponentially. With modern clouds supporting distributed applications easily seeing average VM densities in the range of 10:1, a 10,000 server data center needs to support 100,000 VMs — and VM density is only increasing, not decreasing. Only L3 networking techniques are designed for this scale.

While L3 networking techniques are excellent from the point of view of an enterprise building a new private cloud, they limit how cloud users can use their virtual networking between servers. For example, Amazon's EC2 Classic Networking does not easily allow the use of broadcast traffic or allow customers to pick their own IP address range. Many use cases, including support of legacy applications, require this functionality, which is provided by allowing each customer to have their own L2 network(s).

An ideal arrangement, therefore, is to allow enterprises to build and operate their physical networks using L3 networking techniques, while allowing customers to use L2 for their own purposes. This is now possible using data center network virtualization via Software-Defined Networking (SDN). As the technology and methodology matures, this will become the prevalent model for building agile, cloud scale networks.

Cloud Computing and the Network

Like industrialized automobile robotics factories, bigger is better for private cloud systems. Larger clouds can achieve economies of scale that are impossible to achieve otherwise. Everything from space to power, cooling, servers, storage, networks, and labor can be effectively designed to maximize efficiency. The networks inside these data centers are no exception.

Cloud networks, much like the Internet, have special concerns for two key groups: centralized IT and cloud end users. Real private clouds must be able to achieve efficiency and scalability in their networks. An increase of just one percent in efficiency can provide significant savings. Scalability directly impacts an enterprise's ability to grow effectively in the face of demand while keeping operational costs low.

Meanwhile, cloud users find themselves with two kinds of applications they must support on their private cloud infrastructure: legacy (client-server or mainframe) and cloud-ready. New cloud-ready applications can be designed for the L3 networking of elastic clouds to maximize their own scalability and elasticity. Simultaneously, however, cloud users have huge numbers of legacy applications that make assumptions about operating in the traditional L2 networking environments found in client-server-based data centers today.

Because today's clouds mainly support either—but not both—types of applications, any complete solution for cloud networking needs to address these two very separate concerns:

1. Scalability and efficiency for scale-out cloud applications
2. Support for legacy enterprise applications

For example, Amazon's Elastic Compute Cloud (EC2), the market leader, was largely designed for new cloud applications and optimized for scalability and efficiency. By using a Layer 3 (L3) networking design, they are able to grow to a massive size. Amazon's success in this regard is already well known, with millions of virtual servers running on 500,000+ physical servers today.

Other clouds focus on enterprise support for legacy applications. These are typically VMware-based vCloud providers whose architectures are designed for supporting legacy client-server applications. Their networks are a Layer 2 (L2) networking design, which simplifies cloud user issues with legacy applications but increases complexity, cost, and scalability for your private cloud.

The need for both solutions can be inferred from Amazon's Virtual Private Cloud (VPC) service, which provides a L2 capability on top of their existing L3 network. In essence, delivering the best of both worlds by addressing the two needs simultaneously[1].

[1] Google Compute Engine is known to use similar techniques for their networking model.

SDN and Network Virtualization

As the foundation of the Internet, Layer 3 (L3) oriented network designs can scale to massive size. In contrast, Layer 2 (L2) network designs are known to have significant scalability issues. Most enterprise data centers are oriented on L2 design principles, while most Internet Service Providers (ISPs) and large cloud data centers use the L3 model.

A number of techniques and technologies attempt to solve some of the scalability, performance, and security issues in L2 network designs. For example, VLANs are a mechanism used to provide both scalability and security. Use of VLANs allows for isolation between network segments without using routing (L3). Newer techniques, such as RBridges and SEATTLE, attempt to provide additional mechanisms for scaling L2 oriented networks. Unfortunately, these techniques continue a time-honored incremental enhancement tradition of being 'bolted-on' to the existing L2 methodologies.

An emerging technique that offers far more promise is that of Software-Defined Networking (SDN), a combination of network virtualization and network programmability.

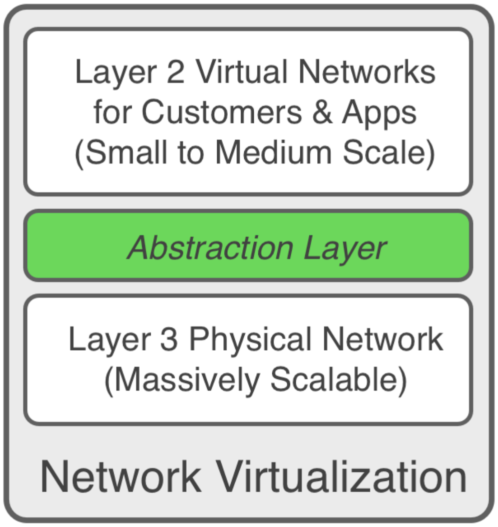
SDN Explained

Software-Defined Networking (SDN) separates the traditional network into a control plane and a data plane for scalability and manageability. Functionally, SDN is really a two-part solution:

1. Network virtualization (implemented in the data plane with an underlay/overlay) and
2. Network programmability (implemented in the control plane).

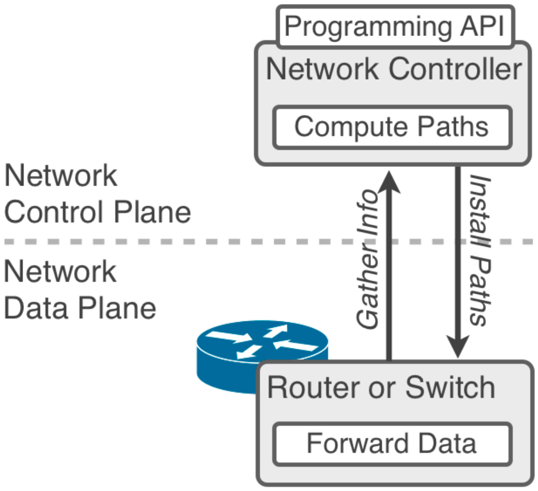
Frequently, SDN is used to describe both, but this can cause confusion. Although the two are related, it is possible to have a programmable ("software-defined") network separate from network virtualization. It is also possible to have network virtualization without programmability. However, the true benefits of SDN are seen when programmability and network virtualization are tied together.

Network virtualization comes into play as an abstraction layer created between the physical cloud network and the cloud user virtual network(s) that provides a separation of concerns. Your private cloud can efficiently operate a physical L3 network topology as the "underlay", while providing a traditional L2 enterprise network data center view for cloud users as an "overlay". Additionally, network virtualization allows customers to pick between the L2 and L3 models. Customers with a new cloud-ready application can choose to use an L3 network design for that application while continuing to use the less scalable L2 networking for legacy applications.



Network virtualization can be achieved through a number of approaches including tunneling and flow control. Most modern SDN vendors are focused on tunneling L2 overlay traffic over a scale-out L3 underlay network. This is sometimes called "L2oL3". The advantage of this approach is that it uses well understood network designs for both the cloud provider and cloud users.

SDN also focuses on providing programmability of network configuration, particularly to configure the virtualized L2 network layer that cloud users see in a typical private cloud. This programmability is enabled by the abstraction of networking intelligence from the distributed networking hardware and into a network controller or SDN controller. This creates a clear separation of the network control plane and the network data plane.



SDN programmability is typically implemented as a rich REST-based API for managing your virtual networks. In some cloud operating systems, such as OpenStack, this layer is standardized using Neutron, while in others, the SDN solution is managed separately. Since private clouds are dynamic environments, a rich API that allows for granular control and management of virtualized networking is critical so that cloud users have the on-demand experience they expect when building, deploying, and managing their applications.

Programmability is key to enabling network scalability of your cloud. Large clouds like Amazon EC2, Google Compute Engine, and Microsoft Azure are highly automated systems. Each new customer deployment requires that networking is allocated appropriately and on demand. Well designed SDN solutions provide an API that can integrate to network provisioning and scheduling systems.

In the following sections, we will discuss L3 and L2 network designs as deployed in cloud provider data centers today, followed by a deeper dive on how network virtualization via tunneled overlays works.

SDN Future

While the early work in SDN has focused on re-inventing traditional enterprise networking, the current development is focusing on managing networking as a service through northbound APIs along with operational use cases for automated roll out of higher level network services. With the convergence of virtualized compute, storage and now networking in modern data centers, SDN needs to address the needs of cloud users (e.g. application developers for networking on-demand), IT administrators (for server virtualization, orchestration and multi-tenancy) and network security administrators (for network virtualization and secure network service delivery).

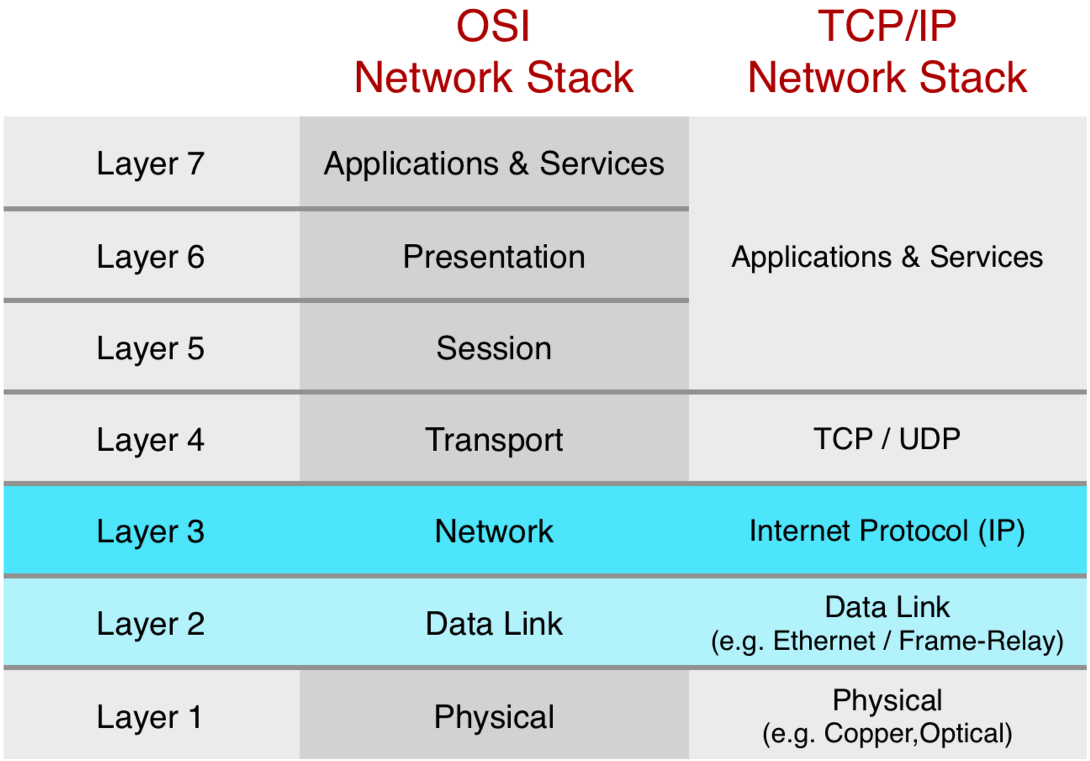
The challenges are in building a robust, scale-out control plane that supports dynamic service automation and application enablement like Network Function Virtualization (NFV) and Service Chaining while seamlessly working with existing network hardware infrastructure. These services, which are beyond the scope of this document, will only amplify the agility and operational efficiency benefits delivered by SDN today.

Three Cloud Networking Approaches

There are three major approaches to building private cloud networks today: L3, L2, and network virtualization. We will first provide an overview, then look at each of the three techniques, how they are used, their pros and cons, and where cloud networking is headed in the future.

Layer 3 Routing Vs. Layer 2 Switching

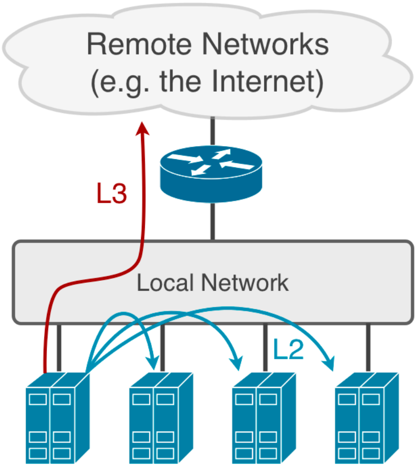
If you are not familiar with the TCP/IP 'stack', we recommend you familiarize yourself with the basics. TCP/IP and other networking models (e.g. OSI Reference Model) are 'stacks' where each layer of the stack provides different functionality. The following diagram shows both OSI and the TCP/IP stacks side-by-side:



	OSI Network Stack	TCP/IP Network Stack
Layer 7	Applications & Services	Applications & Services
Layer 6	Presentation	
Layer 5	Session	
Layer 4	Transport	TCP / UDP
Layer 3	Network	Internet Protocol (IP)
Layer 2	Data Link	Data Link (e.g. Ethernet / Frame-Relay)
Layer 1	Physical	Physical (e.g. Copper,Optical)

The two layers of the stack we care about are Layer 2 and Layer 3. In most modern data centers, Layer 2 (the Data Link Layer) is Ethernet. There are a variety of alternatives to Ethernet, but most are used in Wide Area Networks (WANs), not Local Area Networks (LANs) where Ethernet is king. Directly above Layer 2 is Layer 3, the Internet Protocol (IP) Layer. Every computer on the Internet uses an IP address which allows you to communicate with that computer. When we discuss moving data at Layer 2 in Ethernet-based environments, we talk about 'switching', while in Layer 3, we talk about 'routing'. As implied by the diagram above, switching (L2) and routing (L3) are not entirely independent. One depends on the other.

In this white paper, when we say an 'L2 oriented' network design, we mean one in which the focus of effort is on scalability, performance and security for the switching layer. When we say an 'L3 oriented' network design, we mean the focus of effort is on the routing layer.



Layer 2 and Layer 3

The reason to use an L2 approach instead of an L3 approach is greater simplicity and usability. At the lower end, an L2 network design is much, much simpler and easier to use than an L3 network design. With Ethernet, each device has a unique Ethernet address. When moving between physical locations in the same data center or campus, the Ethernet address tells where that device can be found. This means that the IP address doesn't have to be changed when the location is changed. With IP, however, most devices must have updated IP addresses when their locations are changed.

On the other hand, L2 network designs simply don't work well in large data centers. Most modern private clouds will grow such that L2 network designs will be challenged to scale. The need for every switch to understand the location of every Ethernet address (device) on the network make scaling L2 extremely difficult, if not impossible. This is why L2 is not the fundamental design model for the Internet.

L2 network designs have a number of 'bolt-on' technologies and protocols that attempt to allow them to scale up. Although there have been varying degrees of success, L2 still does not scale as well as desired. Perhaps more importantly, these L2 scaling techniques do not sufficiently acknowledge fundamental customer requirements: L2 networks are usually designed around a single application or customer and hence individually don't need to be very large. VLANs that provide isolation and protocols like Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP) all attempt to manage very large multi-tenant networks, in which each individual network is manageable, but the aggregate is not.

In this regard, SDN network virtualization fits the requirements since it is designed so that each tenant receives its own set of L2 network domains, while the underlying physical layer is scaled based on L3, not L2 networking.

Cloud Networking: The Layer 3 Approach

The Internet is designed using L3 networking, and this is also the approach used inside large cloud providers such as Amazon, Google and Facebook. As the proven approach for cloud network infrastructure, L3 can operate at arbitrary scale. In other words, when built properly, it should be possible to build a L3 network of any size.

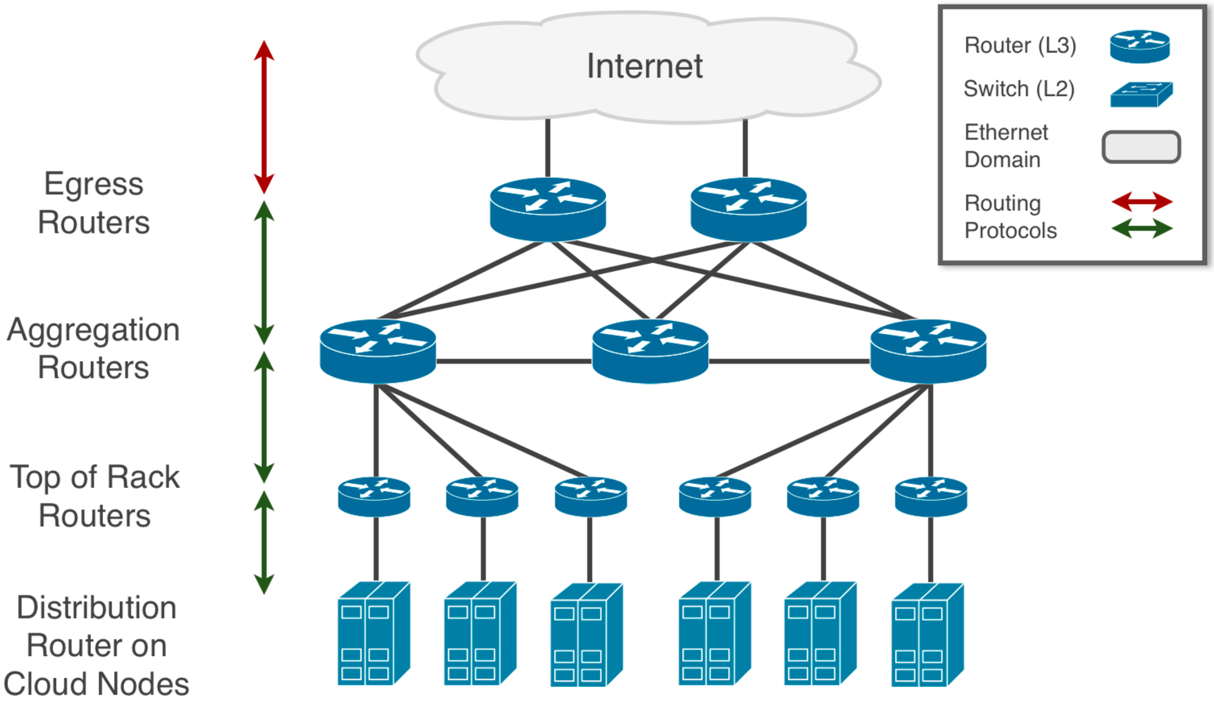
To allow for any size network, L3 networking hides all of the details of the network depending on the topological location. By aggregating route information, using a hierarchical addressing scheme, and only storing what is necessary to understand whether data is local or remote, each system in the L3 network can make relatively simple decisions on a small dataset[2]. In contrast, L2 networking requires that core systems know a lot about every system in the network (e.g. ARP tables contain all the Ethernet MAC addresses).

If we attempted to use L2 networking to run the Internet today, the backbone routers of ISPs would have to hold hundreds of millions, and perhaps many billions of entries in their routing/ARP tables and make decisions across massive datasets. Not only would hardware capable of efficiently handling such data be prohibitively expensive, but the ever increasing size of the Internet itself would eventually make it impossible to keep up.

L3 networking techniques solve this problem elegantly.

L3 in Cloud Data Centers

The following diagram shows how L3 networking works in practice in cloud data centers today. Each node or system in the infrastructure acts as an L3 router and has only enough information to make a local routing decision, routing the data up to the next tier of router as necessary.



[2] The routing table.

The most obvious difference in an L3 routed network topology for a cloud data center is that its routing protocols may be run all the way down to each individual server (cloud node). Each virtual server is directly connected only to its default gateway—in this case, the physical cloud node—and there are no other virtual servers on the same L2 network as the virtual server. This is why broadcast traffic is impossible, multicast networking is hard, and why applications that make assumptions about servers being physically co-located "on the same network" can become confused[3].

L3 Pros and Cons

There are pros and cons to the L3 approach and, while there could be a vigorous debate about whether this approach is better or worse, we know for certain that L3 routing has several key advantages:

- **Manageable networking:** Route aggregation allows L3 to scale. Because each router needs only a subset of all network information to make a decision where to send data, large networks are extremely manageable.
- **Efficient network utilization and improved bandwidth:** Equal Cost Multi-Pathing (ECMP) means that data traveling from one point to another can use multiple paths simultaneously, allowing for very efficient network utilization and greater amounts of aggregate bandwidth[4].
- **Increased utilization and efficiency:** Shortest-Path First (SPF) routing means that in larger networks data will take the shortest and quickest path to its destination, resulting in increased utilization and efficiency.

The primary disadvantage of L3 networking is that locality matters and broadcast traffic is not supported. Unlike L2 Ethernet, it's not possible to simply move a device from one L3 network to another without adjustments. Moving a server requires changing its IP address to an address on the new network. Clarifying the second disadvantage, broadcast networking which is typically used for discovery is inherent to L2 networking and hence does not work in an L3 environment. Similarly, multicast traffic can also be challenging, although it is possible to provide multicast in an L3 network. Despite these disadvantages, L3 is clearly the technique of choice for large scale networking.

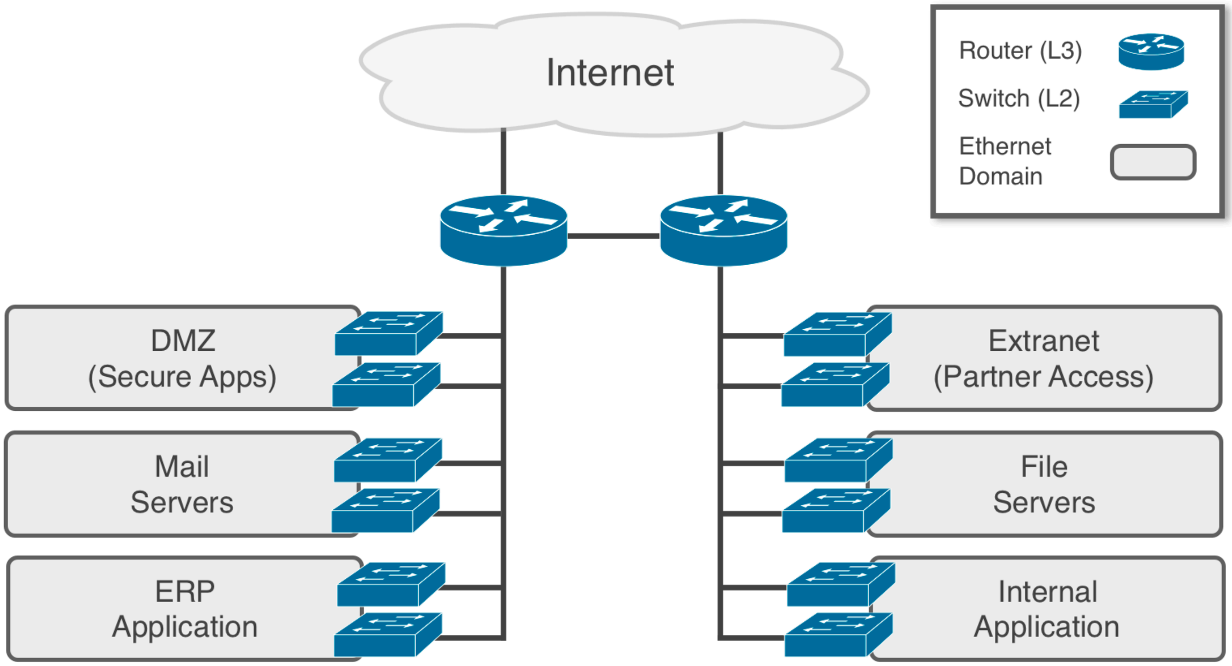
[3] For example, apps that broadcast to discover each other.

[4] In fact, in our testing of Link Aggregation (LAG) vs ECMP, we found that while 2 LAGed NICs delivered a 30% boost in bandwidth, 2 NICs using ECMP delivered a full 100% increase in performance.

Cloud Networking: The Layer 2 Approach

Unlike L3 networking, L2 switching is very easy to understand and use at a small scale. Simply attach any two servers to the same L2 network segment and they can find each other instantly, without involvement from routers. Inside most data centers, L2 networking is the dominant method for connecting devices. While not able to operate at cloud scale, L2 does work well for small and medium deployments. When operating less than 1,000 servers together (and most applications and data center needs are far less than 1,000 servers), L2 networking is easy to use, configure, deploy and manage. As previously mentioned, many modern applications make assumptions about being able to use L2 protocols for discovery and easy networking.

It should be apparent at this point that, at the application level, L2 is a desirable networking technique. From a customer perspective, it's easiest to simply have lots of L2 networks, one for each application or functional area of a data center. The following diagram depicts how L2 networks are created inside enterprise data centers today:



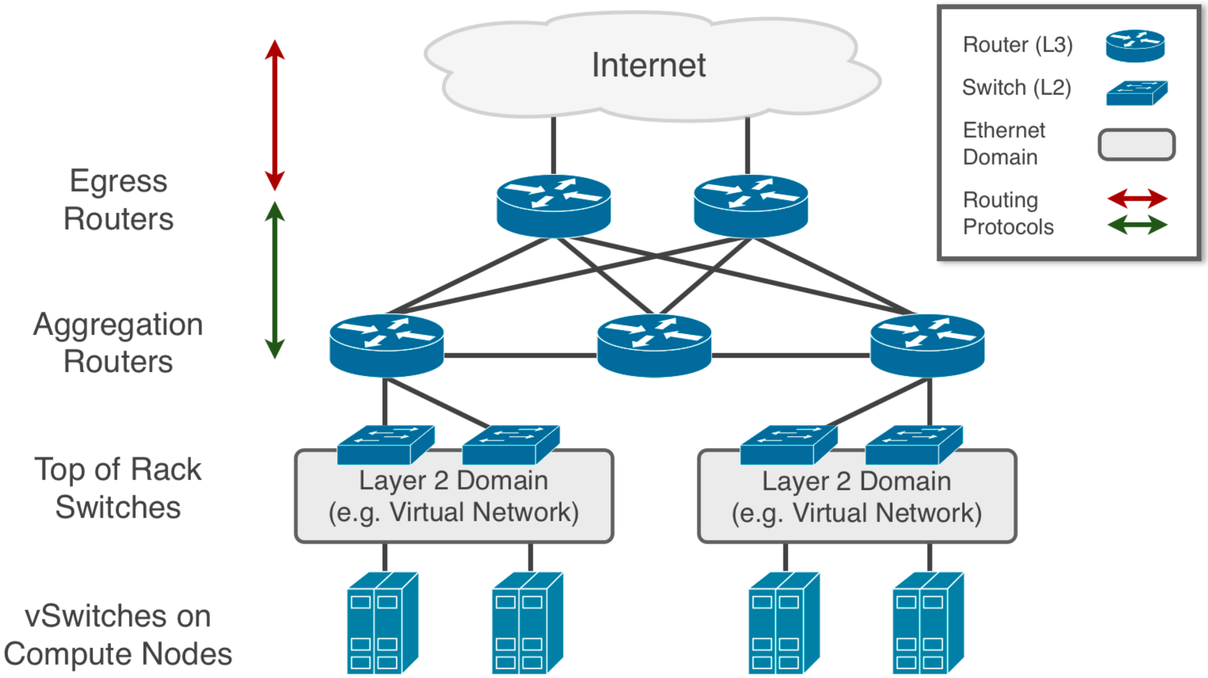
Unfortunately, L2 networking has significant complexity and scaling issues once you reach a certain size.

L2 in Cloud Data Centers

In contrast to L3, L2 networking means that each host knows how to talk to local hosts on the same network segment. A designated router (aka default router or default gateway) is usually assigned for the host to send all data that is not local.

It is convenient for an application distributed across multiple servers to have all the servers on the same L2 network segment. This makes discovery easier, reduces the chances of an impact from a router failure, and ensures that security devices such as firewalls are not interfering. In fact, there is an implicit contract between network operators and their customers. When an L2 network block (e.g. 192.168.0.0/24) is assigned to a customer, it is assumed that the customer can do what they want with the assigned network. This includes adding new machines or changing IP addresses. The only requirement in this contract is that the node sends all non-local network traffic to their default router.

The following diagram depicts an L2 oriented networking topology:



This is a simplified diagram, but it shows how L2 switching in a virtual network hides the physical network topology. For each L2 domain all of the devices appear to be local, meaning here the physical top-of-rack switch (ToR) and the virtual switch (vSW) in the compute node hypervisors are invisible to the virtual machines. This simplifies networking. It also means that if a virtual machine is moved from one cloud node to a different cloud node in the same rack, it just works. On the other hand, if a virtual machine had to move between these two L2 domains, its IP address would have to change and it would now be routed (L3), requiring updates to ensure it can continue to speak with the other devices on its old network.

L2 Pros and Cons

L2 networking (primarily Ethernet) has significant advantages over L3 (IP) in its simplified addressing and ease of use.

The shortcomings for large scale networks, however, are glaring. In particular, L2's flat[5] addressing, which makes simplified addressing possible also makes having large scale networks difficult. IP's hierarchical addressing, in comparison, allows simplifying the problem through route aggregation.

[5] Please be aware that OpenStack incorrectly uses "flat" to describe a routed L3 network. Traditionally, "flat" refers to an L2 networking model. In this white paper, we use "flat" in the traditional manner only, referring to L2 networking and ignoring OpenStack's flat networking model.

While L2 has a flat address scheme, it also enforces a very strict tree topology where each switch must look for loops. This is the problem that Spanning Tree Protocol (STP) was designed to solve. Unfortunately, STP must shut down 50% of links in order to avoid loops[6], which means that ECMP is not possible and half of your available bandwidth is lost. It also means that in L2 deployments most traffic takes the longest route possible, moving from the bottom of the tree, to the top of the tree and back down.

For example, in a large L2 data center network using Gigabit Ethernet, it is not unusual to see 'core links' between switches at the top of the tree that are 10 or 100 times the size of 'edge links' (to hosts). This makes core links expensive as they are usually 'fat pipes' such as 10Gig, 40Gig, and 100Gig Ethernet because all traffic must transit these links between core switches. This puts an undue burden on those switches, making them more expensive, and making L2 difficult to manage and scale for large networks.

Because L2 switching is inherently hard to scale, most attempts to make it more scalable appear to be 'bolt-ons' or require the addition of L3 networking techniques. For example, RBridges (formerly TRILL) is an attempt to provide better topology and L2 switching information between switches using IS-IS. Other companies' attempts to make L2 networking scale, such as SEATTLE and Cisco's Layer 2 Multipath Protocol (L2MP), share the same characteristic of stretching L2 networks to behave like L3 routed networks. An ideal solution, of course, is the combination of L3's scaling properties with L2's ease of use.

Despite its issues, a number of public and private clouds today do attempt to use L2 networking to provide each customer their own L2 network (aka 'virtual network'). Most of these attempts work sufficiently well at small or medium size, but begin to fall apart rather quickly once significant size is achieved. The primary exception here is those public clouds, such as Amazon EC2 and Google Compute Engine[7], who are running their L2 overlay networks over an L3 underlay, achieving the best of both worlds. This is the emerging de facto best practice for the future.

Summarizing L2 VS. L3

This table summarizes the two approaches:

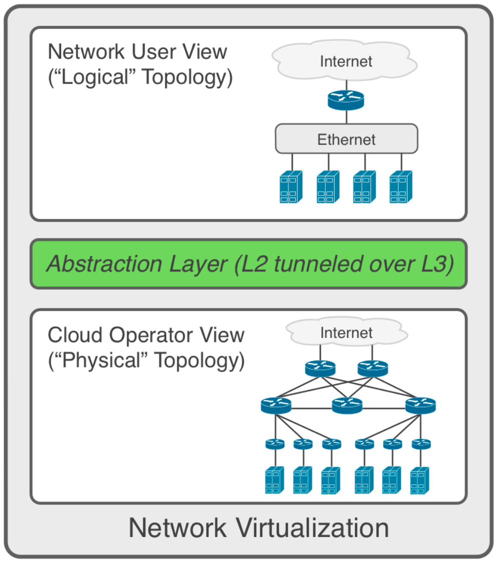
L3	L2
Fast convergence times	Fast convergence times only when tuned properly
Locality matters	Simple to configure
Use all available bandwidth (via ECMP) using multiple paths	Uses half of available bandwidth and most traffic takes a long route
Proven scalability	Tree topology works at small to medium scale only
The Internet is L3 oriented	A typical enterprise data center is L2 oriented

[6] MLAG partially addresses this problem but is not a standard like STP and has its own shortcomings.

[7] Actually, GCE uses a flow-based model rather than a tunneling overlay which is beyond the scope of this document to discuss.

Cloud Networking: The Network Virtualization Approach

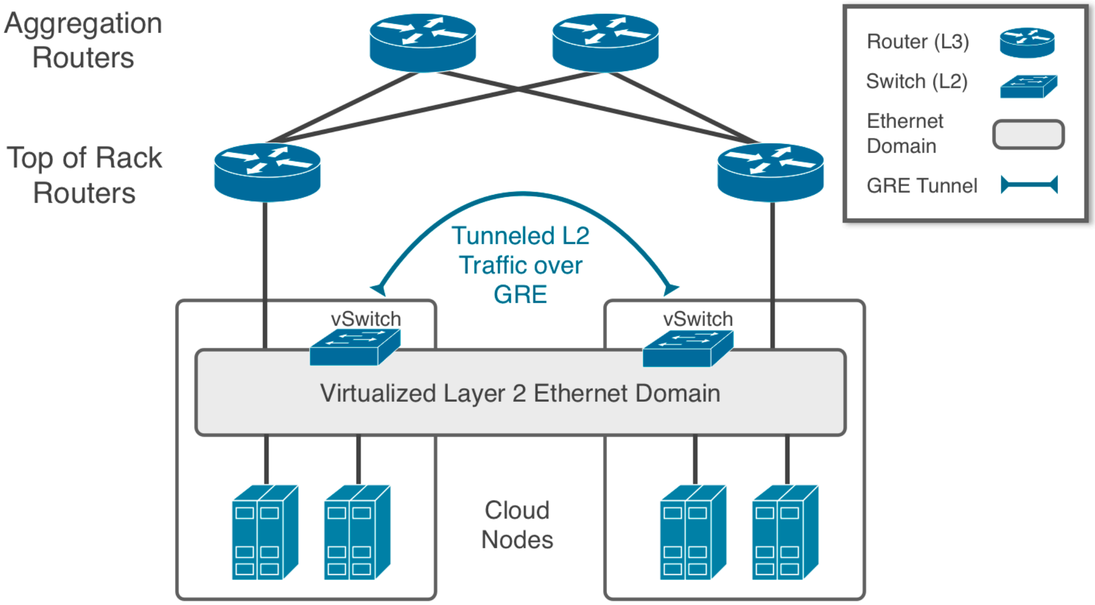
Network virtualization attempts to meld the L2 and L3 approaches. In fact, some refer to it as "L2 over L3" or "tunneling", instead of the normal "L3 over L2"[8].



With this approach, we provide a layer of abstraction between the physical and virtual network topologies. Network customers may have as many L2 networks as they like, wired however they like, while the underlying physical networking is run as a cloud scale L3 network. This allows the best of both worlds. The tradeoff is some additional complexity and concerns around performance, most of which are being addressed today.

The diagram to the right shows how network virtualization works. A key characteristic of true network virtualization is the abstraction layer between the 'physical' and the 'virtual'. This is akin to the hypervisor in a virtualized server. The hypervisor acts as an abstraction, hiding the physical hardware and providing a virtualized set of hardware for the guest operating systems.

This is still a bit abstract. Let's take a look at a more detailed network topology that shows how L2 over L3 actually works. The following diagram shows how each cloud user's vSwitch can provide a dedicated GRE tunnel to every other cloud user, thereby creating a fully virtualized (and dedicated) L2 domain, much like a VLAN:



In the diagram, note that, invisibly to all of the customer virtual servers, traffic that is not on the local cloud node (shown as a white box) is tunneled over L3 using the GRE protocol.

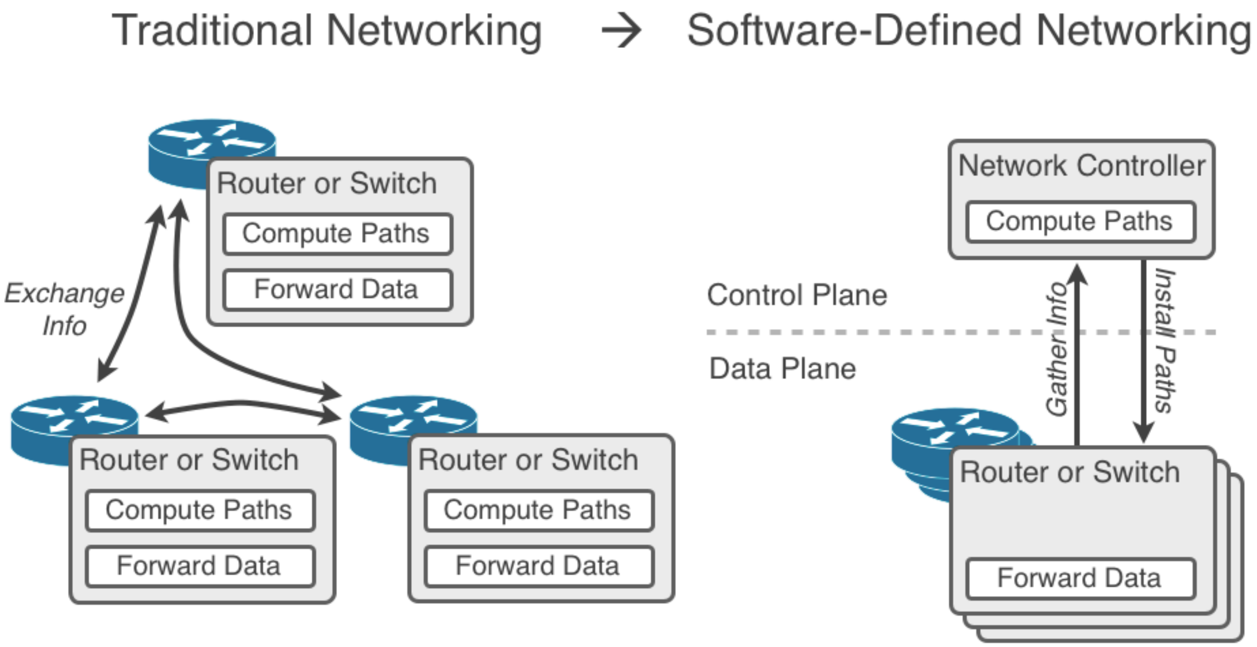
[8] Layer-3 is 'over' Layer-2 in the TCP/IP stack.

SDN In Cloud Data Centers

Software-Defined Networking (SDN) is the future for building private and hybrid clouds. It is the only solution that can provide the best of both worlds. As mentioned previously, you can think of SDN as two components: the control plane and the data plane. The control plane is focused on network programmability and management. The data plane is focused on delivering a robust network virtualization capability.

The Cloud Networking Control Plane

The creation of a control plane is instrumental to Software-Defined Networking. With traditional networking, the network intelligence is distributed across all networking devices. Routers and switches exchange network information with each other and they compute traffic paths to forward data on to the right destination. This system and its protocols have incrementally evolved over the past few decades, resulting in a network fabric that is hardware-defined, rigid and expensive to operate. SDN specifically extracts the creation of network paths from the distributed network devices and into a centralized controller. This enables the network to be re-programmed on demand by users and applications to create and manage network traffic paths.



The Cloud Networking Data Plane

The simplified data plane is left with the role of receiving network path configurations from the central controller and forwarding traffic on the network. This separation of the control plane from the data plane facilitates the programmatic configuration of L2 traffic over L3 physical networks.

This network virtualization in the data plane is being converged on by many. Microsoft has written white papers detailing this SDN technique from the perspective of a large scale data center operator, while folks like Juniper and VMware have been working on the programmatic and management aspects. For historical information, we recommend reading Microsoft's original white paper on Valiant Load Balancing, and its update on 'VL2', both precursors to more modern SDN techniques.

There are a number of ways to implement the "data plane" of network virtualization, and although it is early days, we'll certainly see more. Examples include:

- Virtual Extensible LAN (VXLAN)
- Generic Routing Encapsulation (GRE)
- Network Virtualization using GRE (NVGRE)
- OpenFlow + VXLAN/GRE/NVGRE tunneling

All of these network virtualization techniques share one commonality: isolating or tunneling L2 traffic over L3 networks using an overlay encapsulation protocol. This simple but powerful approach allows a private cloud to have a highly scalable network using L3 while giving each customer its own set of simple to use and easily configured L2 networks. In addition, because customers are completely isolated from one another, it's possible to run overlapping IP address ranges and provide advanced features such as quality of service (QoS), distributed firewalls, network function virtualization (NFV) and service chaining.

VXLAN

Virtual Extensible LAN (VXLAN) is an overlay encapsulation protocol for network virtualization that uses a VLAN-like approach to encapsulate L2 Ethernet frames within L3 UDP packets. VXLAN is included as a standard part of the Linux kernel and, given that it's supported as the data plane or transport protocol for many vendor's SDN solutions, it's as close to a standard as you can get. It scales up to support 16 million logical L2 networks running above L3 IP networks with support for multicast.

GRE/NVGRE

Generic Routing Encapsulation (GRE) is a tunneling protocol that encapsulates a wide variety of network protocols within virtual point to point links over an L3 IP network. It works by wrapping the original IP packet with an outer IP packet that is then removed at the GRE tunnel endpoint. GRE also has the advantage of supporting multicast between networks. The GRE protocol is a proven approach, having long been used in conjunction with PPTP (Point-to Point Tunneling protocol) to create VPNs (Virtual Private Networks). Network Virtualization using Generic Routing Encapsulation (NVGRE) is a competing approach to VXLAN that is very similar but leverages the GRE protocol.

OpenFlow + Tunneling

OpenFlow is a communications protocol that enables network controllers in the control plane to configure network routers or switches in the data plane. OpenFlow can be loaded as firmware onto physical switches. By running the same protocol for configuration and overlay tunneling on both physical and virtual switches it is also possible to run mixed cloud environments in which the cloud user has both virtual and physical servers. This will be an important concept in the future, as some intensive workloads, such as databases and file servers, may not be virtualized.

Summary

Enterprise infrastructure teams need network scalability and cloud users need the ease and familiarity of simple Ethernet networking. SDN delivers this. It is truly virtualization in that it provides a clean abstraction layer creating a separation of concerns between the underlying physical network and the cloud user's virtual networks.

We believe SDN is the path forward for modern data centers. It simplifies networking for all and avoids complex bolt-on technologies. Best of all, it plays inherently to the strengths of commodity systems. Instead of buying increasingly expensive networking gear, it is much less expensive to scale-out using L3 networking techniques on commodity equipment and then use the abstraction layers to hide it all beneath.

Contact Us

To learn more about how Cloudscaling's products and services can help you deploy and manage private elastic cloud capabilities, just give us a call at +1-415-508-3270, email us at sales@cloudscaling.com or visit our website at www.cloudscaling.com.

ABOUT CLOUDSCALING

Cloudscaling is the leader in elastic cloud infrastructure. The company's core product, Open Cloud System (OCS), is the world's most advanced OpenStack cloud infrastructure system. OCS is designed to meet the requirements of next-generation dynamic applications, delivering the agility, performance and economic benefits of leading cloud providers, but deployable in the customer's data center and under the IT team's control. Cloudscaling is backed by Trinity Ventures and headquartered in San Francisco. For more information, please visit www.cloudscaling.com.

Cloudscaling
45 Belden Place
San Francisco, CA, 94104
Main: +1-877-636-8589
International: +1-415-508-3270

Cloudscaling is the trusted source for information on OpenStack and together with the community is making OpenStack more production-grade. For more information, please visit www.openstack.org.