MINT 709 – CAPSTONE PROJECT REPORT

Software Defined Networking (SDN) Based Solution for Data Center Construct

Project Mentor: Noonari Juned
Name: Nooruddin Khorasi
Course: Masters of Science in Internetworking

ABSTRACT

In past few years, Software Defined Networking (SDN) has been a popular topic for discussion and debate in computing world. It is highly sought after evolving architecture for delivering cost-effective, dynamic, programmable and extensible network platform which can deliver and deploy new services in minutes of time. SDN is getting widespread acceptance from community in all areas of computing right from application development, storage needs and network services in data centers. SDN allows development of agile, centrally managed and vendor-neutral network architecture by decoupling control functions from underlying network infrastructure and managing the whole network from a single, extensible and centralized software-based controller.

The purpose/intent of this project report is to traverse through the rich history of Data Centers, explore it's evolution to modern day virtualized, multitenant and cloud Data Centers and document the impact and changes SDN is driving through the Data Center Constructs since its inception.

This project report is divided into three major parts. The first part of the project is an in-depth explanation of traditional Data Center Construct, its primary components and overall working mechanism. It also discusses the evolution of Data Center Constructs to multitenant, cloud and virtualized Data Center. The second part dives into core fundamentals of SDN framework and how it addresses the challenges and problems of Modern Day Data Centers. It also showcases the challenges, possible solutions and use cases of Software-Defined Data Center (SDDC). The part three showcases research work undertaken by SDN community while addressing the challenges in widespread acceptance of SDDC.

Keywords: Data Center, Software Defined Networking (SDN), Cloud computing, Network Virtualization, Network Functions Virtualization, Open Flow, Software-Defined Data Center (SDDC), Multi-Tenancy.

TABLE OF CONTENTS

TABLE OF FIGURES

1 DATA CENTER INFRASTRUCTURE, COMPONENTS, WORKING MECHANISM AND EVOLUTION TO MODERN DAY VIRTUALIZED DATA CENTER.

1.1 DATA CENTER IN A NUTSHELL

Data Center is a centralized infrastructure used to house computing, storage and networking systems for an organization's IT needs. Depending on business requirements of an organization, Data Centers can take up to one rack or extend to a huge facility. Data Centers can be in-house or may be outsourced to third-party organization who then owns the equipment and provide access to it as a service. Data Centers hosted by third party vendors are prescribed to support less applications and serve more users while in-house Data Centers are meant to host huge number of applications and serve lesser number of users. Figure 1 depicts a typical Data Center designed by layered approach. Right from its inception way back in 1940s, primary function of a Data Center is to store, manage and disseminate critical data of an organization and provide reliable and timely access to it. Data Center is brain power of any organization as it is through Data Center all services inclusive of web hosting, e-commerce, SaaS and social networking are offered.

Figure 1: A layered, traditional data center setup on white space or physical facility. Source: https://gigaom.com/2010/09/19/now-online-yahoos-chicken-coop-inspired-green-data-center/

Two primary accreditation organizations that classify Data Center tiers and benchmarking are Uptime Institute (UI) and Telecommunications Industry Association (TIA). ISO/IEC is responsible for international standards pertaining to Data Centers while CENELC and ANSI are responsible for European and American Standards. Both have four Tiers to classify Data Center in terms of availability and reliability of Data Center Infrastructure. All these four Tier however justifies that Data Centers should have enough redundancy and backup of systems and components to beat high-cost downtime and maintenance problems [1]. As per Datacenter Dynamics report, TIA has decided to drop its Tier system and update the ANSI/TIA-942/TIA-942A. TIA will collaborate with UI to come up with new benchmarking tools that will business improve the design and operations of data centers in their existing facilities [2].

The following Tables summarized the UI benchmarking standards.

UPTIME INSTITUTE DATACENTER INFRASTRUCTUER TIER CLASSIFICATION

	Tier I	Tier II	Tier III	Tier IV
Active Capacity Components to Support the IT Load	N	N+1	N+1	N After any Failure
Distribution Paths	1	1	1 Active and 1 Alternate	2 Simultaneously Active
Concurrently Maintainable	No	No	Yes	Yes
Fault Tolerance	No	No	No	Yes
Compartmentalization	No	No	No	Yes
Continuous Cooling	No	No	No	Yes

Source: http://www.gpxglobal.net/wp-content/uploads/2012/10/TIERSTANDARD_Topology_120801.pdf

1.2 EVOLUTION OF DATA CENTER

The history of Datacenters can be dated back to early 1940s when US Army developed ENIAC to calculate artillery firing tables for Ballistic Research Laboratory. This huge machine can be thought of as Datacenter of today's world with minimal capabilities as it was a full featured computing facility with necessary storage and infrastructure in place with power backup.[1] This machine occupied a huge floor space, required technicians for its proper functioning and did approximately 5000 numerical operations per second. Apart from computing power, this machine used ten-position ring counters to store the digits for performing arithmetical operations.[2] In 1960s, the development of minicomputers and invention of solid-state devices significantly reduced the size, cost and power consumption of computing hardware. The storage memory of computer started being replaced from vacuum tubes to solid state memory. Wide commercial acceptance brought IBM System series mainframes into the market and soon

[1] http://en.wikipedia.org/wiki/History_of_computing_hardware
[2] http://en.wikipedia.org/wiki/ENIAC

mainframes were being used not only by government and military agencies but also by many commercial organizations. This drift required the systems to be more powerful and to have more utilization of resources. IBM teamed up with American Airlines to automate a reservation system called SABRE using two IBM 7090 mainframes in their data center in New York. This system was a huge success to have the capacity of processing huge number of transactions and about 83,000 daily phone calls.[3] CDC 6600 was also one of the major innovation in 1960s and was by far the most modern representation of today's data center.

Early 1970s introduced world's first microprocessor, the 4004 developed by Intel and the term microcomputer started making more sense when most of the processing power was built into a small microprocessor chip. Datacenters were still used for bookkeeping duties and so organizations started developing disaster recovery plan as it won't hamper the overall business operations. Development of Xerox Alto brought a paradigm shift in computing world as the mainframe computers were now being replaced with graphical interfaced, bit-mapped high resolution screens and more internal and external storage. The mainframes computers required cooling. With introduction of minicomputers, air-cooled computers became widespread. Late 1970s brought an important discovery of ARCnet to the computing world. Initially designed using token-ring architecture to allow as much as 255 computers to interconnect and provide 2.5 Mbps of data rate, the LAN protocol broke new grounds and thousands of ARCnet installations went commercial. This decade also brought the concept of virtualization for datacenters to wide acceptance by introduction of IBM VM/370 OS. Now mainframes had exceptional resource utilization and could handle multiple applications and execute multiple tasks. Owning datacenters would still cost a huge fortune to companies as the resources required for floor space, cooling and operational management were too much [3].

1980s brought microcomputers into the picture with the introduction of IBM PC. Organizations had to worry less about operational requirements and the cost incurred by that as the small sized IBM PC became widely accepted. With more and more installments of IBM PC, organizations had to figure out a way to control all their IT resources. During this decade, Sun Microsystems came up with network file system protocol to access network files from a client PC. All these acceptance made organization question the need of virtualization and started abandoning its implementation for multitasking and better resource utilization. With more adaption of IBM PC and advent of UNIX, vendors started coming up with more sophisticated OS with multitasking and timesharing incorporated within them to facilitate client-server networking. More and more vendors started programming their own version of UNIX. But what was common to all these versions of UNIX is their capability to run multiple applications within single instance of OS [4].

1990s is well-known for its impact in the world of computing and telecommunications. This period also known as 'dot-com bubble' brought drastic and most important changes affecting

[3] http://en.wikipedia.org/wiki/Sabre_%28computer_system%29

every bit of computing world including datacenters. As more and more Internet based companies were founded, each had a demand for sophisticated network operations and fast connectivity to establish themselves on rapidly growing Internet. Big companies started incorporating facilities called Internet Data Centers to manage large scale operations. What was called microcomputers a decade back was now called servers and was mounted on walls as new and cheap networking equipment started to become industry standards. This was a peak time when numerous hosting companies started to incubate to offer reliable and cheap hosting services to small companies which could not afford to establish their own data centers. This widely accepted traditional data centers were called *silos* with each silo dedicated for use by a particular application. This limited the data centers to physical space as a new application would require a new silo to be established and allocation of new storage to be considered. Thus traditional data centers were tightly bounded by physical space and number applications that can be hosted troubling hosting companies in expanding to more clients. Data centers now offered more reliability and increased performance but would still lack best possible resource utilization and efficient, secure service [5].

The problem expanding within limited physical space and very low resource utilization brought virtualization again into action in early 2000s. With traditional data center in place, IT Operations was divided into two main authorities. The servers itself allowed for application-based decisions for developers and programmers while the data center hardware operations and management remained an enterprise-wide decision. As more and more desktop servers started being part of data centers, managing the data center environment became painful. To accommodate Cross operability and software compatibility across various systems, tools like Virtual PC, VMWare workstation, etc. were developed. Virtualized Data Centers started coming to existence by pooling the storage, network and computing resources from various siloed data centers into a centralized resource which could be reallocated based on the organizational and application needs. However, the siloed datacenters that makes up the virtualized data centers still had to be managed separately. And with virtualized datacenters, resource utilization improved drastically but had inimical impacts on networking and storage components. Organizations had to devise data center designs that brings less detrimental effects on resources and is also easier to manage leaving the resource utilization intact [6].

Virtualized data centers had its own shortcomings. Consider an environment where resources are shared by applications with different and ever changing workloads. If utilization of a shared resource by an application increases, there is an immediate shortage of resource for some other application using the same resource. This could break Service Level agreements stating 100% server and resource availability. One solution was to perform resource allocation programmatically which led to software defined data centers of today's computing world. But the solution is not robust enough as businesses these days have hybrid applications operating within a network. Some may be real-time compute based applications and some way require simple but not on demand tasks to be performed. This required software-based data center infrastructure and not just software-based resource allocation. A complete automation

framework was needed to automate applications within and between data centers. Software defined data centers was now leading to an infrastructure completely controlled by software right from its resource allocation to automated configuration, monitoring, deployment and provisioning. It is thought to have a centralized hub or controller to control, manage and monitor a network of data centers. It is thought to have automated control over hardware of data centers right from cooling and power to storage elements [7].

1.3 DATA CENTER COMPONENTS

No matter how Data Centers are designed, the key components of modern day Data Center can be aggregated to following [8]:

1.3.1 PHYSICAL RAISED FLOOR OR WHITE SPACE:

The actual usable space in physical facility to house IT equipment. Efficient use of white space to accommodate physical equipment and growing business is highly emphasized. It is also very important to consider the operational environment control parameters set on computing devices by manufacturer and have the physical facility maintain its standards in sync with those parameters. Raised Floor Data Centers are common as it addresses easy underfloor cabling, handles heat and cooling loads and allows proper channeling of hot exhaust air to CRAC unit. Raised Access Floors (RAF) were initially used to run the cables and connectors inches long into the mainframes. Data Centers were offered cooling by freezing the physical facility. Later the concept of air cooling was adopted and underfloor air was used to offer data center cooling and RAF got widespread acceptance in Data Center Design [9]. To avoid using RAF, organizations used chilled water lines through the wall behind the data centers or underground pipelines. But using chilled water lines came with completely different overhead of leaking water and inconvenience of having ports in ceiling.

Raised access floor was the only key point that mostly differentiated the benchmarking standards laid down by TIA and UI. However, advancement in Technology standards have brought new ways to eradicate RAF to solve data center issues. For example, aisle containment and air chillers took over RAF cooling methods. Cable designs have drastically improved to remove the RAF completely from design documents. Water pipes can now be easily run overhead to limit RAF issue for underfloor piping.

1.3.2 RACKS:

Data Center equipment like switches, routers, servers and other rack mounted systems are usually hold fast on physical space using 19-inch standard racks as shown in Figure 2. The equipment are designed in height multiples of 1.752 inches which is also called as one rack unit or simply U. The standard 19-inch rack is widely used and accepted by computing and telecommunication industry because of its efficiency in using floor space by dense stacking of equipment. A standard rack is designed to take around 42 1U equipment. Racks may be also equipped with rack-level air or liquid cooling, management unit, in-built KVM and power distribution. The racks can be even modularized further by using chassis that can hold servers in

itself. These chassis comes with power supply, back panels and management unit. A single chassis can hold around 16 servers. However, this implies more densely spaced hardware in single rack and eventually drastic increase in per rack power consumption. Also, to stay within limits of storage Bandwidth it may not be possible to load the entire rack with equipment [10].

Figure 2 A standard 19-inch Rack

1.3.3 NETWORKING DEVICES:

The core of data center design is the networking equipment stacked on racks or in white space. The most common networking devices found in any data center are Gateways, Switches, Routers, Hub, Firewall, Modem, NIC, Wireless NIC, Proxies, Wireless Access Points, Channel Service Unit and Data Service Unit.

Servers are major equipment found in Data Center setup. They could be a software application serving clients, actual physical server, a group of servers forming server farms, rack server or blade server providing with high performance and compact size. Mainframes have been part of Data Centers for a long time but only IBM System-Z machines are now popular amongst this category.

Cabling is one important factor to connect all these network devices in such a way that Data Center can accommodate growing business needs. Connectivity should be as modular as possible to achieve high availability and prevent degraded network performance during downtime and maintenance. Cat5E and Fiber Optic Cables are widely used for data

communication and power devices. Data centers these days prefer overhead cabling as under floor cabling can have cooling issues if it lacks enough space for air flow [11].

One of the biggest concerns in designing a Data Center infrastructure is to keep in mind tight capacity and redundancy requirements. Network redundancy is extremely important when dealing with downtime and maintenance. It is usually achieved by leasing from providers, supporting highly redundant network connectivity or by leasing from multiple providers. Network redundancy is of two types – Active-Active or Active-Passive. As their name suggests, Active-Active redundancy has both end circuits up and running while Active-Passive turns up the passive circuit in case of active circuit failure. Active-Active is widely used and usually implemented by using BGP4 protocol. Latency is also an important issue while considering network provision. Routing model used by provider and SONET technology used by provider all add up to factors used for calculating the latency. SLA (Service Level Agreement) dictates these factors and should be properly negotiated while planning bandwidth requirements.

Network Interface Cards (NICs) are another important components in networking enabling physical connection to network. They are present onboard and have a 6-byte long unique MAC address that switches use in making forwarding decisions.

Switches are common layer-2 devices used for segmenting the network. They primarily setup a Source Address Table that provides a list of port and corresponding MAC addresses that can be reached through that port. Initial Broadcast packets are sent to learn about a new port and MAC address combination. Switches operate on Layer-2 of OSI model and have different switching methods based on which they are distinguished. Cut-through, Store-and-Forward, Fragment-Free are common switching methods found.

Routers or Layer-3 Switches are commonly seen in Data Center Infrastructure. Operating on Layer-3 of OSI model, primary function of Routers is to perform routing using IP address. It assigns IP subnets to local ports and reads the destination IP address in incoming packets to perform routing.

Load Balancers are used to balance high density traffic among different hosts in the network.

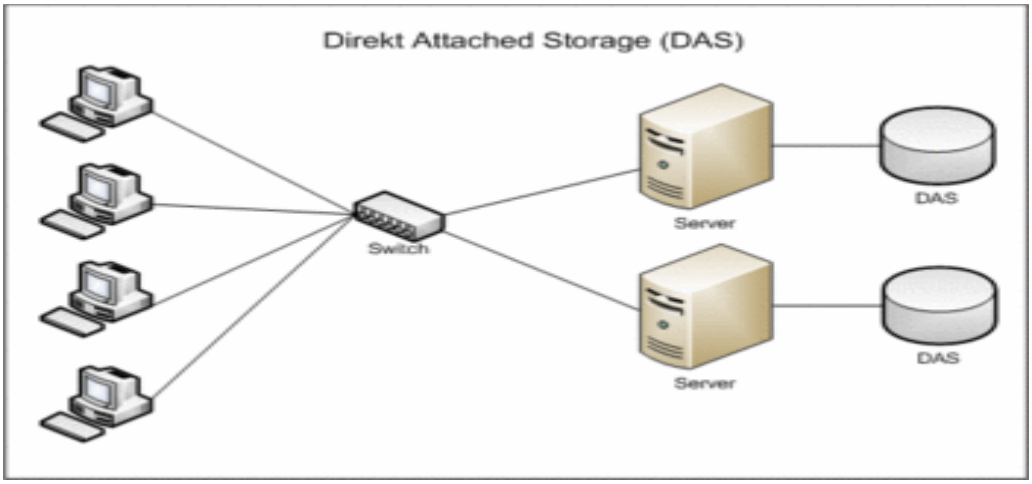
Firewalls are primarily used in Data Centers to prevent unauthorized access from any host based on address of source or destination. They can software firewall installed on the system to be protected or a completely separate network firewall which will then restrict access to network. Firewalls are also used for URL filtering, spam and virus filtering, restricting data transfers and IDP.

1.3.4 STORAGE TECHNOLOGIES:

Providing effective storage and access to organization's data is becoming crucial with the introduction of BigData. Data is also stored in many different forms. The three primary form of storage forms known are Direct-Attached Storage (DAS), Network Attached Storage (NAS) and Storage Area Network (SAN). While DAS is server-based solution, NAS is file-oriented and SAN

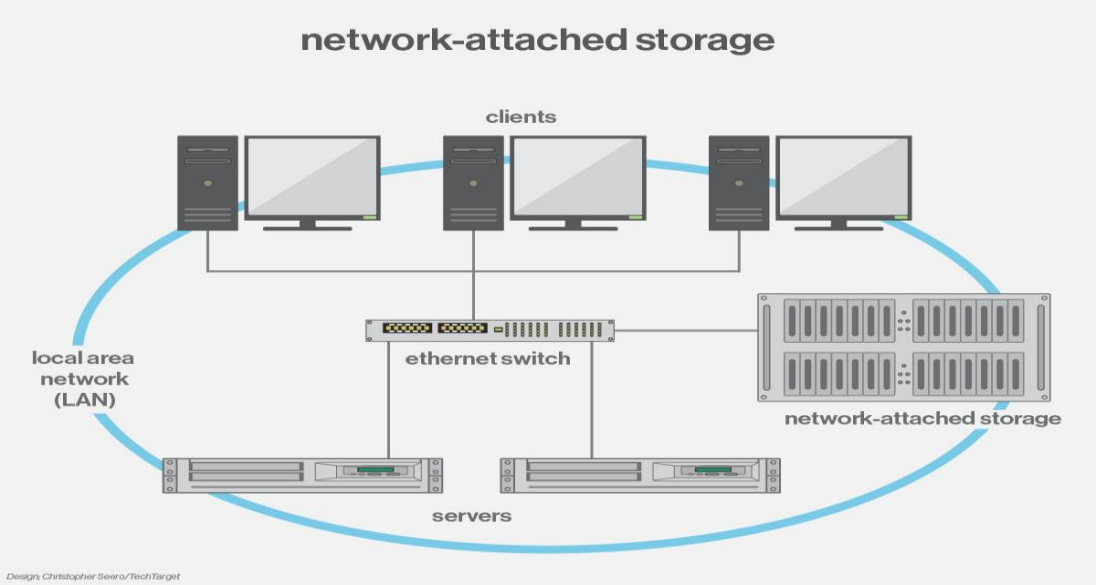
provides file-level access to data. SAN is popular choice in Fibre Channel configuration with widespread acceptance of iSCSI and FCoE [12].

DAS is traditional storage method where storage devices have a directly connected path to individual server using SCSI, SATA or SAS interfaces. Storage can only be accessed through server. Access from network is provided using a controller. Performance is really good as data can be directly accessed from the server instead of traversing through the network. Though server crash can zero-out access to storage, introduction of virtualization has again made DAS a popular choice.



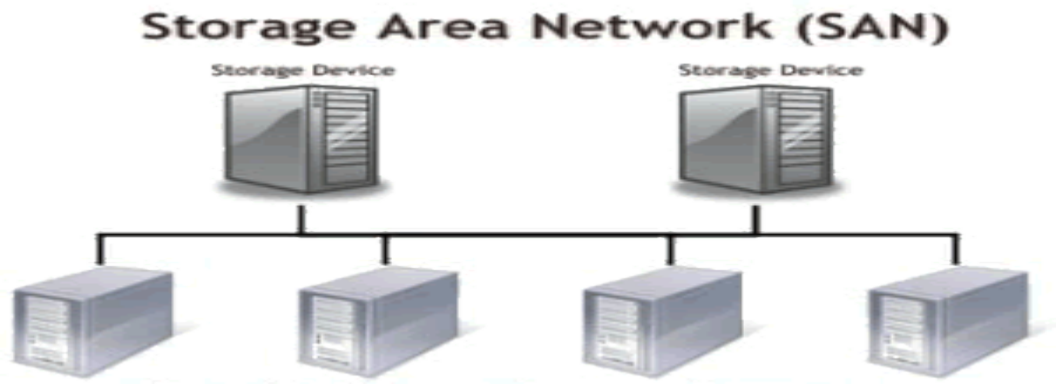
Source: http://winfwiki.wi-fom.de/images/thumb/e/eb/DAS.gif/350px-DAS.gif

NAS is another form of Storage in Data Center providing file-level access to data. The storage systems are directly attached to LAN and can be shared by heterogeneous set of servers as far as they run IP protocol. It provides excellent redundancy of data as storage is shared and access through network. They are configured using web utility and is considered to be an independent node on network with its own IP address [13].



Source: http://cdn.ttgtmedia.com/rms/onlineImages/network_attached_storage.jpg

SAN represents a subnetwork wherein storage systems are pulled from user network and make up their own independent network. Multiple servers can then access shared storage network by sending block-access request. It constitutes of cables, bus adapters and switches and switches are directly connected to storage devices. Fibre Channel SANs are widely used but are expensive and has high operational cost. iSCSI was designed to overcome this by encapsulating SCSI packets into IP packets [14].



Source: http://www.netvpro.com/images/storage-area-network.gif

1.3.5 SUPPORT INFRASTRUCTURE:

Equipment required to support Data Center operations and high infrastructure-wide availability. This includes UPS systems, batteries, generators, ventilation and exhaust systems, computer room AC and physical security devices like CCTV cameras and biometrics.

1.3.6 OPERATIONS STAFF:

Sufficient Manpower is required to ensure controlled-access, proper management, maintenance and round the clock availability of Data Center systems.

1.4 DATA CENTER: WORKING MECHANISM

1.4.1 CLASSIC THREE-TIER DATA CENTER ARCHITECTURE:

Since late 90s Data Center networking was dominantly built upon three-tier network. Dividing network into hierarchy promotes modularity and helps in designing more reliable network with greater performance, easy administration, scalability, high availability and less maintenance cost. Data Centers are designed to meet high Data volume transfer and aggregation and switching was made easy using three-tier network by specifying switching responsibilities with each layer. Data transfer between client and server is done using Ethernet, InfiniBand is used for server-server communication and Fibre Channel for data storage traffic [15].

Figure 3 shows a traditional three-tier architecture used in Data Center for decades. Let's dig into the working mechanism of traditional Data Center using this architecture.

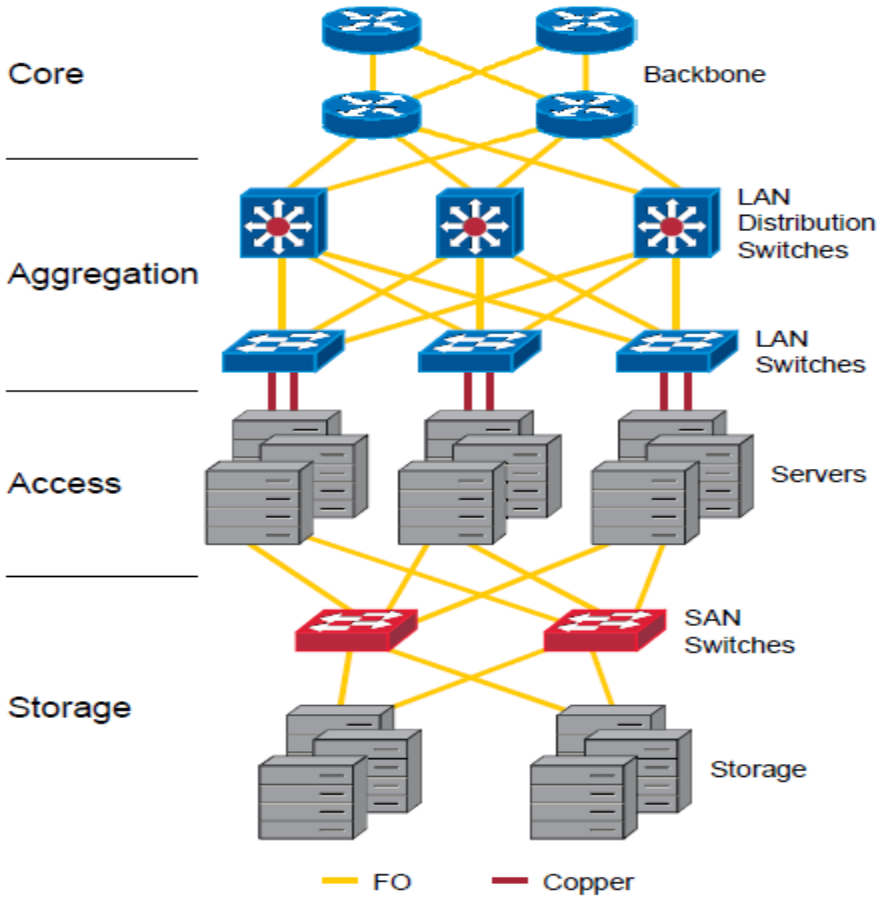


Figure 3 Traditional Three-Tier DC Architecture

ACCESS LAYER:

This layer is primarily responsible for connecting end devices, terminals and edge switches to rest of the network. It defines rules on which devices could connect to the network and how each of these devices will connect to each other. Other primary devices could be hubs, WAP, bridges and routers. It allows users to use services provided by other two layers. Access Layer allows only few set of systems to connect to LANs by filtering MAC addresses. It improves the performance by allowing each node to have its own collision domain. High percentage of Bandwidth usage is also possible by having a single network handle all data and allowing load balancing by moving traffic to a different network [16].

Access Layer allows features like port security, PoE, STP, Virtual Access Lists, QoS classification, Layer 2 switching, inspecting ARP packets, etc. Switch manufacturers allows control access features for access layer like user authentication on ports. High availability and low operating cost and power consumption can be achieved by using low latency switches in access layer.

Access Layer can be programmed using three different strategies viz. Layer 2 (loop-free), Layer 2(looped) and Layer 3.

Looped Access Layer design implemented using square and triangle topologies is shown in Figure 4. Primary reasons of using this model in access layer is adding new servers in particular VLAN easily and allowing excellent redundancy throughout looped topologies. Also active stand-by service modules require Layer 2 adjacency for high cluster availability.

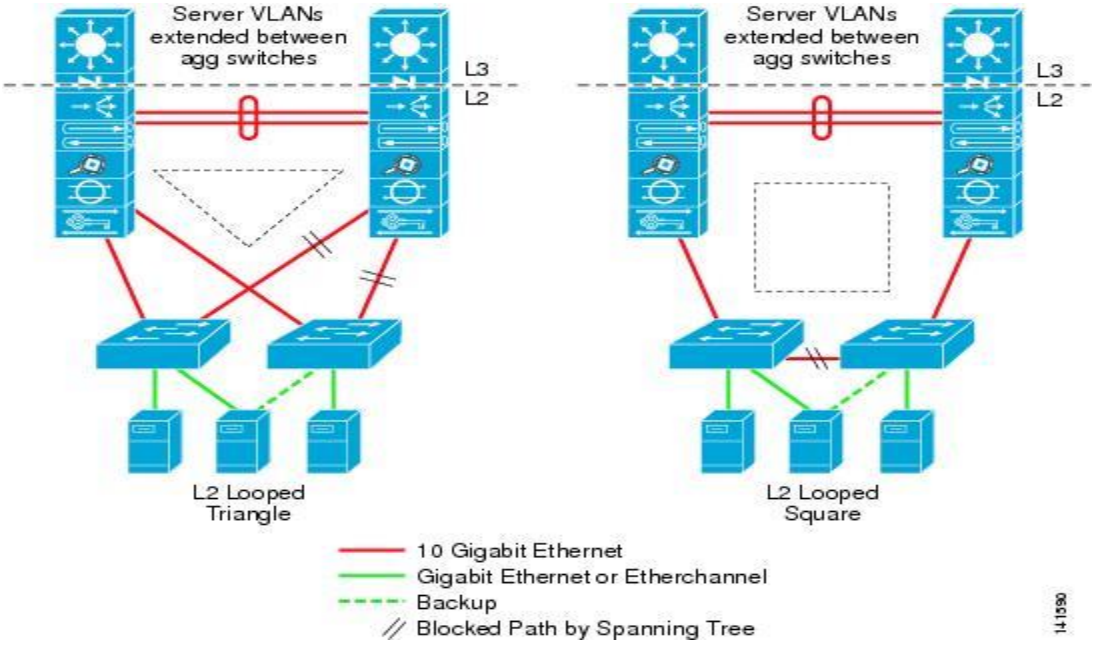


Figure 4 Looped Access Layer Model (Cisco.com)

Loop-free Access Layer design implemented using U and inverted-U topologies is shown in Figure 5. This model is primarily used to have all uplinks active at the same time or when difficulty in implementing STP protocols. However, VLAN extension is only supported with inverted-U topology. Considering it

doesn't have loop detection inbuilt, it is highly recommended to run STP to prevent or detect possible loops.

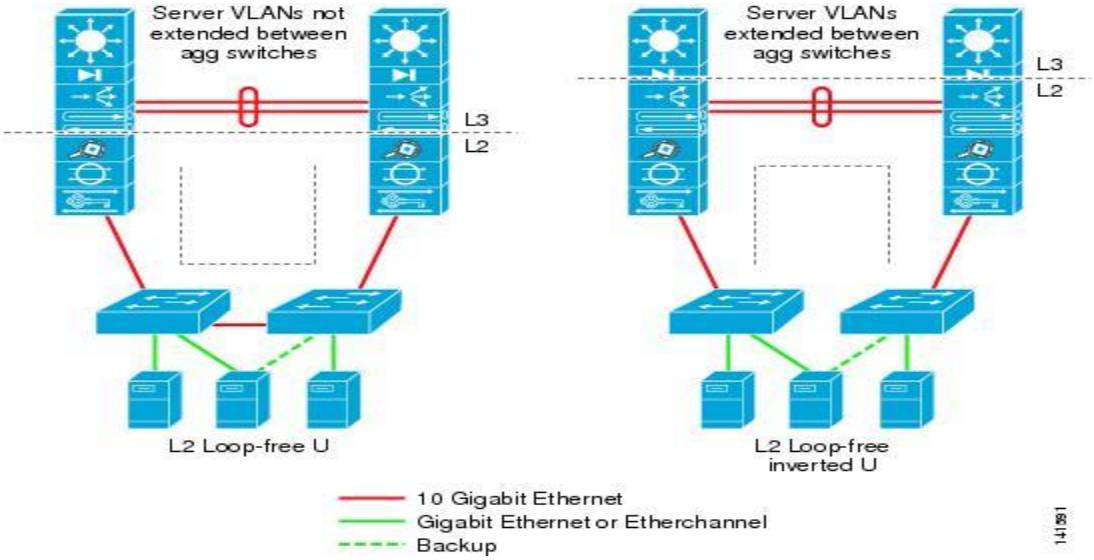


Figure 5 Loop-Free Access Layer Model (Cisco.com)

Layer-3 design model is possible in Access Layer as shown in Figure 6. It allows limiting broadcast domain size so that servers getting affected with certain broadcast level can be protected. It allows all uplinks to be active and load balances traffic up to ECMP maximum without using complex routing algorithms. As with the loop-free model, it is recommended to use STP to prevent loops. Also, VLAN extension across the DC is not possible [17].

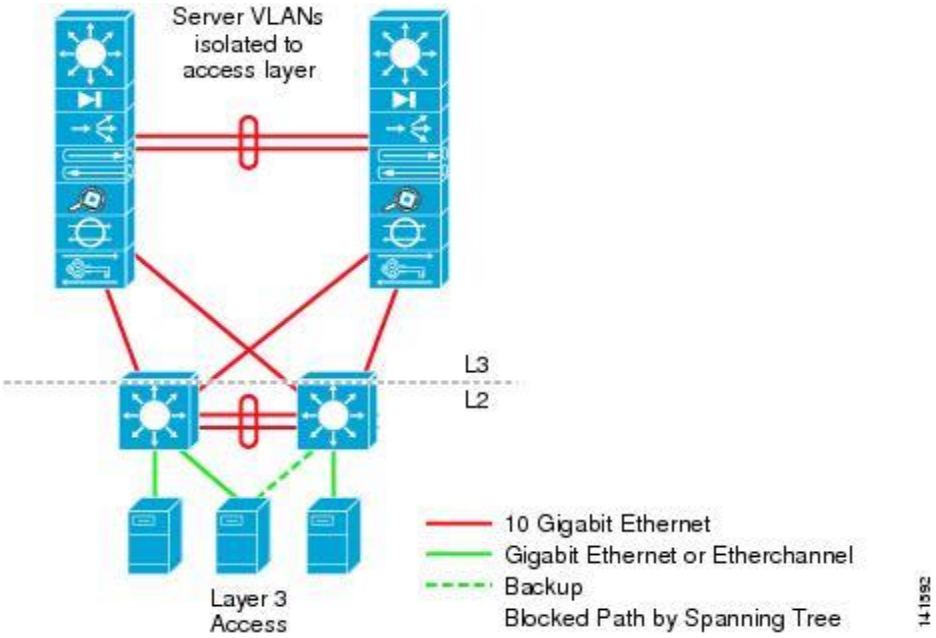


Figure 6 Layer-3 Access Layer Model (Cisco.com)

AGGREGATION/DISTRIBUTION LAYER:

Aggregation Layer or Distribution Layer manages all sessions that traverses through the data center. Aggregation switches are different from access layer switches wherein they are required to have high forwarding rates and top speed switching fabric. Firewalling, load balancing are primary features of this layer. This layer requires keen attention while designing DC infrastructure as it deals with features like port density and CPU processing which has its own implications. It allows aggregating of edge switch cabling into high speed links thus reducing cabling and ease network management. Considering large aggregation from thousands of users, this layer usually deals with MAC tables of extremely large size and is designed to provide low latency.

Thus aggregation layer can be thought of providing with LAN/WAN aggregation, ACLs to provide security, redundancy, load balancing, route aggregation and summarization, etc. It performs routing between VLANs in access layer before transmitting data to core layer. Switches in this layer are programmed in pairs as to provide redundancy because these switches have to deal with utmost load from the network. As huge traffic is involved in this layer from variety sources, QoS is extremely important to prioritize between traffic.

The traffic flow in aggregation layer is of two different types. The North-South traffic flows from core layer to access layer when client initiates resource access to web server farm. The East-West traffic flows between access layers or between the servers in web farm primarily for replicating data or backing it up. It is possible to have such flow even in multitier application viz. from web request to application and from application to database. Security and scalability is achieved by using load balancers and firewalls. East-West traffic is extremely crucial and difficult to manage because of frequent backups and updates across the servers.

CORE LAYER:

In terms of dense traffic flow, this layer is quite equivalent to aggregation layer. It deals with connection between devices in aggregation layer and thus high redundancy is also of prime importance here. To support aggregation layer with sufficient bandwidth, this layer should be able to perform aggregation of links. With extreme traffic load on core switches, they should be provided with sufficient cooling. The core switches allows traffic passing without 1Gbps or 10Gpbs limits and functions as both routing and switching engines [18]. This layer is primarily concerned with reliable packet delivery and speed across the network. It is the only layer to allow access to internet resources.

1.4.2 TRANSMISSION PROTOCOLS

Once everything is wired up or the infrastructure and devices are in place, they need to communicate with each other with standard networking protocols. They can be connection-oriented or connectionless. The protocols forms a stack (generally called TCP/IP protocol stack) where each layer is associated with protocols devices use to communicate with each other. Generally, protocols act upon layer to layer communication in OSI or TCP/IP model.

TCP/IP model was designed based on the generic OSI model and hence the protocol stack of each model relates to each other. Basic relationship between OSI model layers, TCP/IP layers, protocols used for each layer and devices operating at each layer is shown in Figure 7.

L#	Device Type	OSI Layer	TCP/IP Org.	TCP/IP New	Protocols	PDU
7	Gateway	Application	Application	Application	HTTP, FTP, POP,SMTP, DNS, RIP	Data
6		Presentation				Data
5		Session				Data
4		Transport	Transport	Transport	TCP/UDP	Segments
3	Router	Network	Internet	Network	IP, ARP, ICMP, IGMP	Packets
2	Switch/Bridge	Data Link	Link	Data Link	Ethernet, Token Ring	Frames
1	Hubs/Repeater	Physical		Physical		Bits

Figure 7 OSI & TCP/IP Protocol Stack.
Source: http://jaredheinrichs.com/wp-content/uploads/2013/10/osi-cheat-sheet-01.png

Let's dive into different protocols layer by layer.

1.4.2.1 PHYSICAL LAYER/DATA LINK LAYER:

These physical layer deals with data transmission over physical medium be it electrical or optical. The Data Link Layer deals with error control, framing and flow control. Major protocols involved at these two layers are IEEE 802.3, IEEE 802.2 and PPP. IEEE 802.3 is best known protocol in this layers. It is primarily used in LAN applications where DLL is sub divided into two layers viz. Logical Link Control (LLC) and Media Access Control (MAC). It specifies and documents characteristics of Ethernet and physical media. Possible media are 10BASE-2, 10BASE-5, 10BASE-F, 10BASE-T and 10BASE-36 where 10 is possible transmission speed of 10Mbps, BASE refers to baseband signalling, F refers to fiber optic cables, T refers to twisted-pair and 2, 5, 36 refers to coaxial cable length. Other variants are 100BASE-T and GigabitEthernet.

1.4.2.2 NETWORK LAYER:

This Layer is based on IP-based packet forwarding. It forwards the outgoing packets to appropriate link layer based on next-hop. It directs the incoming packet payload to upper layer. Popular protocols in this layer are IPv4, IPv6, ICMP, IGMP and IPSec (for securing IP Packets). The packets are checked for integrity and errors in DLL and this layer would use routing tables to send information further in network. This layer is responsible for inter-network connectivity while DLL is primarily responsible for local devices. Network layer does datagram encapsulation, routing, logical addressing, error handling and fragmentation and reassembly.

1.4.2.3 TRANSPORT LAYER:

This layer is equivalent to transport layer of OSI Model but sometimes also deals with high-level functions of level 5-7 of OSI model. It allows software processes and applications to communicate with each other. It divides large data into chunks, keeps track of arriving data and defragments it at the other end. It manages the data transmission rate and handles transmission issues using connection-oriented (TCP) and connection-less protocols (UDP). Thus primary tasks

of this layer could be application process addressing, flow control, connection establishment and termination, sending acknowledgements, multiplexing and de-multiplexing and segmentation/reassembly.

1.4.2.4 APPLICATION LAYER:

This layer constitutes of what is session layer, presentation layer and application layer in OSI Model. The primary protocols in this layer are HTTP, SMTP, POP, DHCP, FTP and Telnet. It allows exchange of application data over the network. Programmed libraries and APIs are used to encapsulate the data and present it in a desired format. Well known IANA port numbers are used by applications while deployment. It deals with all the important session management, half-duplex/Full-duplex communication and cryptographic functions for presenting data.

Thus in short the data flow starts with the transmitter application forwarding data to application layer which in turn goes down layer by layer and at each layer the data is headed and trailed by additional information and encapsulated into desired format for that layer. The information then passes through the physical media and the whole process is reversed on the receiver side where the data is unwind by removing headers and trailers passing it to application layer for use.

1.4.3 SHORTCOMINGS OF TRADITIONAL, NON-VIRTUALIZED DATA CENTER.

Although the classic three-tier model is still being used by many companies, it is still not a viable solution for today's data center dealing with BigData. The top reason for this is that setting up such infrastructure and managing it in long run is extremely expensive and tedious. Traffic travels all the way to aggregation and core layers going through over-subscribed congested planes and takes time in microseconds (which is a lot) using traditional vendor solutions. Having such long turn around in East-West traffic between applications and SAN storages incurs lot of funds running down the water.

The problem just doubles when virtual machines run on servers. Now virtual servers can be anywhere on network and limits IP addressing. Maximum VLANs that a network manager can create will be 4096 and thus it will require dividing of virtualization clusters into even more smaller chunks. Managing such smaller clusters is hectic as performing a simple task of moving a VM to less loaded server can be tedious. Vendors comes with solution for VLAN Migration that are locked to their proprietary devices and brings in extra complexity with configurations required.

Making changes to such hierarchical infrastructure is not easy. One may end up adding more and more aggregation switches and overloading ports in core switches. When applications requires more throughput, one can trunk multiple Ethernet connections to it if the application is running in the same rack. Even STP puts limits to number of connections for a switch creating high latency network. Also, when added networking features like firewall and load balancing comes into picture, this hierarchical model fails to address the issues. Firewalls can be configured on virtual servers but when that server needs to talk to storage systems, problem arises as data streams over storage devices cannot be filtered due to performance issues and

high latency. Intent specific firewalls can be used but that just adds more and more physical devices to network increasing the cost and maintenance complexities [19].

Excess hardware and management cost involved with that is just one issue of enterprise data center. Some other important factors of traditional data center that led to the idea of Virtualization are [20]:

- Periodic backup of servers is cumbersome and tedious. For every single backup, operations team have to worry about availability of backup server and maintaining current data on the images backed up.
- Excessive hardware devices in data center prevents organization from their noble cause towards green IT. Reducing power and cooling in such rigid infrastructure is extremely costly and difficult.
- With non-virtualized servers in place, testing is a nightmare. Any issues during test phase can result in corrupt deployment.
- With traditional servers, organization is locked-in to a particular vendor it decides to go with. This brings in almost no flexibility and choice when organization wants the best of every vendor in the infrastructure.
- Recovering from physical ad-hoc disasters in data center space can be tragic with data centers functioning only with hardware devices with no software application control.
- In a non-virtualized data center, one server is destined to perform a single data center function. Having a single server partitioned for serving as email server, database server and application server is not possible.
- Traditional data centers cannot scale to business demands easily.
- Monitoring and reporting only-hardware systems is tough.
- Resource utilization is significantly low in traditional data centers.

In conclusion, traditional data center infrastructure with its rigid structure still fails to deliver flexibility, high availability, great performance, low latency and tight-budget provisioning.

1.5 RISE TO VIRTUALIZED, MULTI-TENANT DATA CENTER:

As seen in previous section, Data Center Infrastructure was lacking better resource utilization and was draining available physical space by excessive use of hardware devices. It triggered the innovation of Virtualization that converts the hardware devices into software resources. It also reduces the final cost organizations end up spending on their data center resources. When much of the data center functions are handled using software applications, the power and cooling infrastructure becomes less cumbersome and can be managed effectively. Let's dive into the history and causes that changed the traditional data center infrastructure into robust and scalable solution using Virtualization.

1.5.1 HISTORY AND NEED FOR VIRTUALIZATION:

Virtualization came into picture in 1950s when programmers panicked when their code could not use more than the main memory. In 1959, the Atlas Team brought in the concept of Paging

wherein the application would use virtual memory address of both main memory and auxiliary memory and would quickly decide on which data could use the faster main memory in comparison to the auxiliary memory. Soon in 1970s, IBM released System/370 which virtualized mainframes. It had a Control program or Hypervisor that would allow creation of VMs and resource sharing. Now a single mainframe was able to run multiple instances of Operating Systems replacing the older time-sharing method. Thus these virtualization techniques proved the use of virtualization in efficient storage use, high availability of resources, lesser physical hardware and transparency to end user as the virtual environment to them appears as if having total control over physical hardware. In 1980s, Insignia Solutions came up with SoftPC which took VM concept to another level by allowing DOS to execute in Unix environment. Soon Apple Inc. created Virtual PC and VMWare came into existence with its VMWare workstation. All these product lines added to widespread market acceptance of Virtualization. But still x86 virtualization faced many problems like low resource utilization, high infrastructure costs, cumbersome management and insufficiency with failover and recoveries from disasters. In 1999 and from there on, VMWare pioneered the Virtualization techniques and became market leaders in delivering top-class virtualization products. Virtual Machines started inspiring innovation in technologies that covered all areas of a Data Center viz. Network, Storage and Servers as seen in Figure 8. However when virtualization was applied to aspects of Data Center there were number of challenges to be tackled. It was possible to minimize the hardware use by virtualization but it was really difficult to figure out amount of physical hardware necessary to address the virtual workload. Also, not many applications were supported to be executed on Virtual Server. Not many vendors provided virtualization and thus organization were locked-in to specific vendors. Addressing the VM sprawl was challenging and a major issue. With more VMs in place sharing common Storage Area Network, the I/O operations were extremely high and a single problem with SAN affected all VMs [21]. Let's now explore how virtualization is supported over the time with applications, servers, network, storage media and application infrastructure services in Data Centers of modern world addressing the above issues.

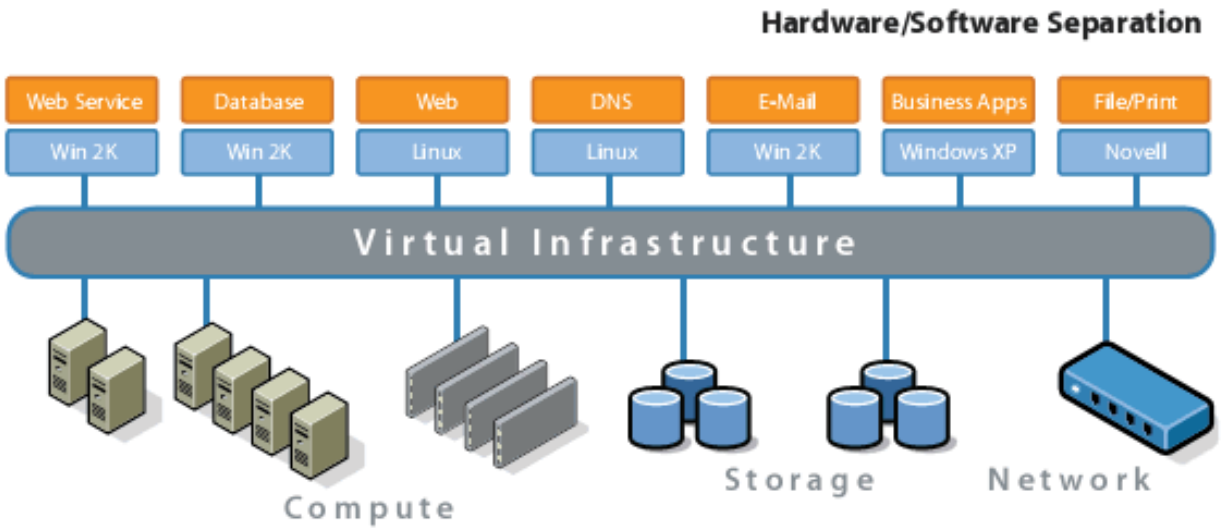


Figure 8 Virtual Infrastructure.
Source: http://capitalhead.com/media/1933/vmware_virtualinfrastructure.png

1.5.2 Network Virtualization

First thing that comes to mind when networking a Data Center is Ethernet. Since 1983 when Ethernet was standardized to IEEE 802.3, it has evolved from data rate of 10 Mbps to 100 Gbps. Ethernet frames have been transferred both over wireless media and wired media (using standard coaxial, twisted-pair and optical fiber cabling standards) for years.

As depicted in Figure 9, Network topologies have evolved from serving a campus community to geographically-spread organizations. Servers which were directly connected to core switches are being replaced now a whole new 3-layer network setup to address server proliferation.

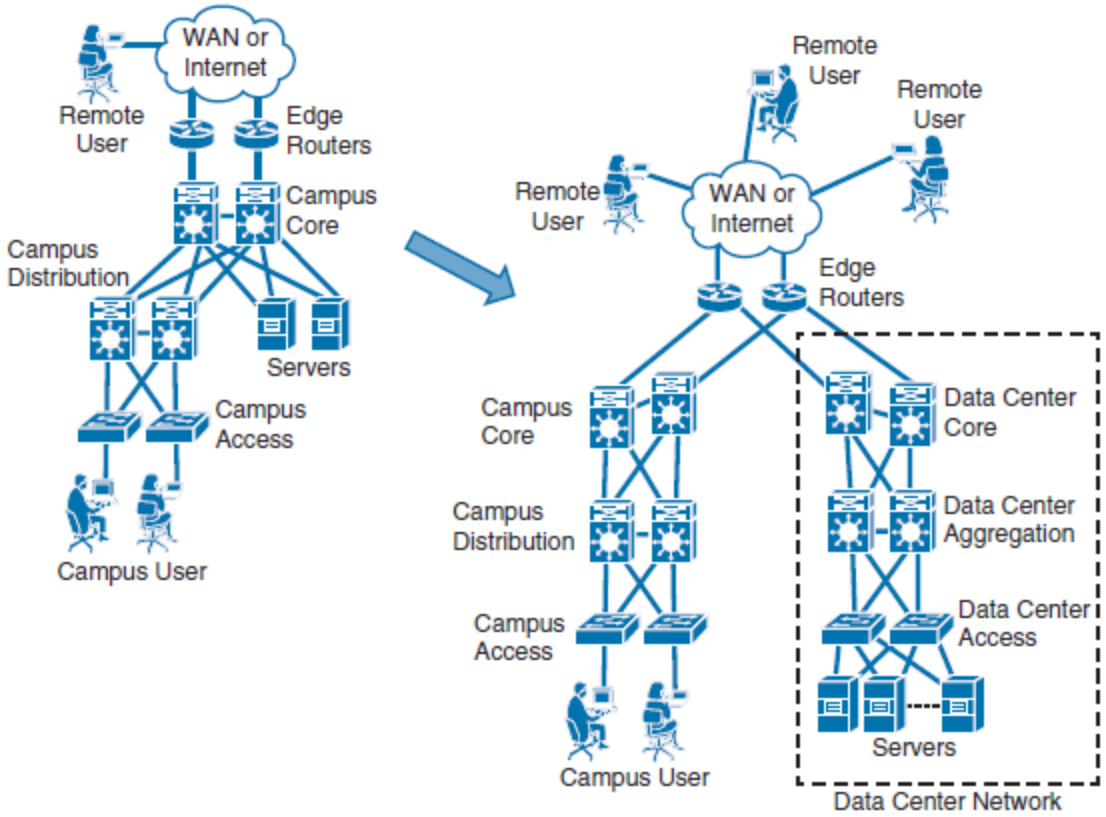


Figure 9 Network Topologies: Then and Now
Source: Data Center Virtualization Fundamentals by Cisco.

With emerging Ethernet speed, the network had to address more number of devices and scale with growing business demands. The obvious solution was to partition networks into multiple domains leading to technologies delivering network virtualization. Two important techniques leading to network virtualization are VLAN and VRF.

Network virtualization is abstraction of logical network from underlying physical network resources. There are two different network virtualization techniques. External network virtualization deals with aggregation and isolation of virtual networks from physical network using technologies like VLAN. Internal network virtualization deals with emulation of physical network with software by configuring software containers in systems.

Virtualization of network allows provisioning of application's virtual network by decoupling network services and configurations from physical network and automate it with software applications. It reproduces L2-L7 network services at software virtualization layer hosting the hypervisor and virtual switch. Thus it provides with logical routing, logical switching, logical load balancing and logical firewall services. Network is provisioned using APIs at virtual switch's software layer. It also allows taking periodic backups of data center architecture along with the VM backups. Physical network is now only responsible for applying packet forwarding between hypervisors. Thus network virtualization solves the complex problem of manual deployment of network features, policies and services.

Network Virtualization however needs to address some complexities that comes with Virtual Machines. A network upgrade might require to use 1Gb to 10Gb Ethernet to meet the demands of fast VM provisioning. Broadcast traffic is blocked through VLANs so applications with such requirements may need a workaround. Switch memory and processors have to be fast enough to handle extra workload. Recovery from Failures should be quick if more workloads are to be added to network [22].

Figure 10 shows how implementation of Network Virtualization idealize with that of Server Virtualization. Network Virtualization takes the network functions like VLANs, VRF, Load Balancers and Firewalls and creates a virtual instances of network resources for use with VMs. Virtual Networks can be updated with new features easily without colliding with physical hardware upgrades which would takes years. Network provisioning is done in seconds. For example if a new switch port is to be added, no cabling and new physical switch ports are required. New ports are easily added to virtual instance and provisioning is done. New technologies are easily accustomed in network by just creating new logical network to serve the purpose instead of configuring a new physical network from scratch.

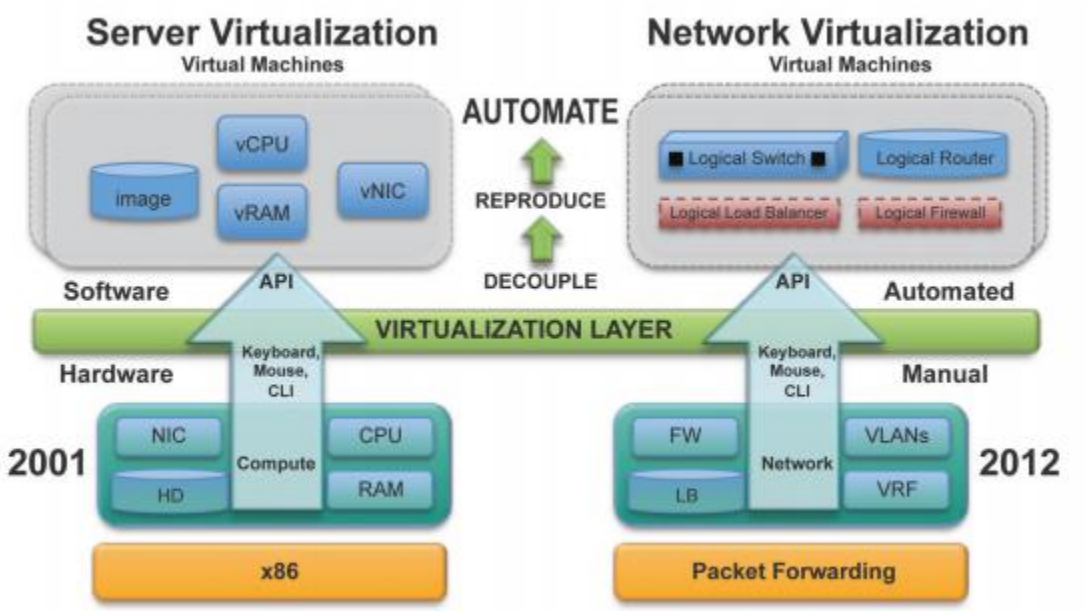


Figure 10 Network Virtualization v/s Server Virtualization.
Source: http://www.vmware.com/files/pdf/products/nsx/VMware-NSX-Network-Virtualization-Platform-WP.pdf

1.5.3 Server Virtualization

Servers have gone through a series of innovations in data centers expanding from Mainframes to Reduced Instruction Set Computing Servers to x86 platforms. Alongside the server itself, the CPU has gone through changes as well in terms of clock speed, number of processors, etc.

Servers have seen some serious improvements in the memory being used going from old vacuum tubes to super speed RAM. The physical format of server has evolved from room size to blade formats allowing more servers to be used within confined physical space. With such innovative servers in data centers, the server utilization has to be up to the mark.

With x86 virtualization in effect, VMware came up with GSX and ESX. VMware ran as an application allowing virtualization of hardware including processor, memory, storage and networking for installing guest OS. VMware ESX however was installed directly on top of physical hardware allowing more control over actual hardware than GSX solution. However both were inspired by concept of Virtual Machine. Shared file on SAN or NAS were accessed using Virtual Machine File System concurrently. Cisco developed Unified Computing System (UCS) as a solution to server virtualization using intermediate component Fabric Interconnect.

There is subtle difference between server virtualization and server consolidation. Different workloads from different servers are consolidated into single, large server in server consolidation. Email and database instances can be combined into one to reduce number of physical servers. Consolidation is more suitable for homogeneous workloads in single OS space. It reduces license cost and administration cost by eliminating excess software. Virtualization on other hand reduces number of servers offering improved administration and low provisioning time of new applications in Data Centers [23] as seen in **Figure 11**.

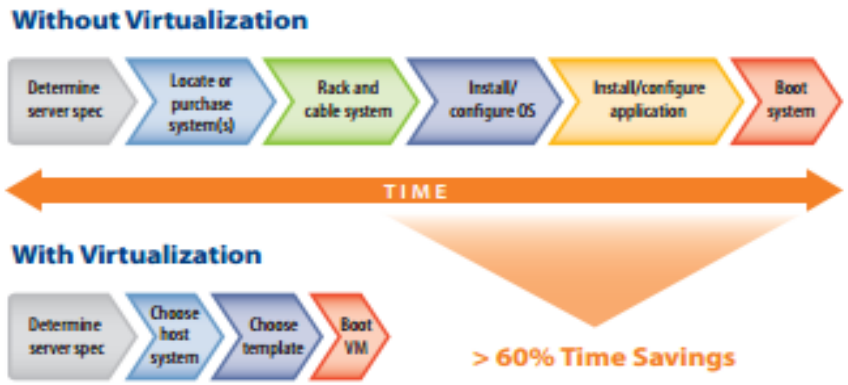


Figure 11 Server provisioning with Virtualization.

Server virtualization brings in great advantages in Data Center infrastructure as listed below and the growth of virtualized workload is exceeding physical workload every year as seen in **Figure 12**.

- Drastic change in overall utilization of server resources. Non-virtualized servers have 8% to 15% of utilization while virtualized servers is known to have about 80% of resource utilization.
- Excellent savings in Capital Expenditure and Operating Costs.
- Easy provisioning, automation and management using centralized software application.
- Multiple OS on single server.
- Easy VM provisioning allows redundancy and backup in no time.

- Having separate server for each application allowing easy test and deployment of applications without intervening with other applications.
- Failure recovery is quick and easy as Virtualized servers can be moved and installed easily.

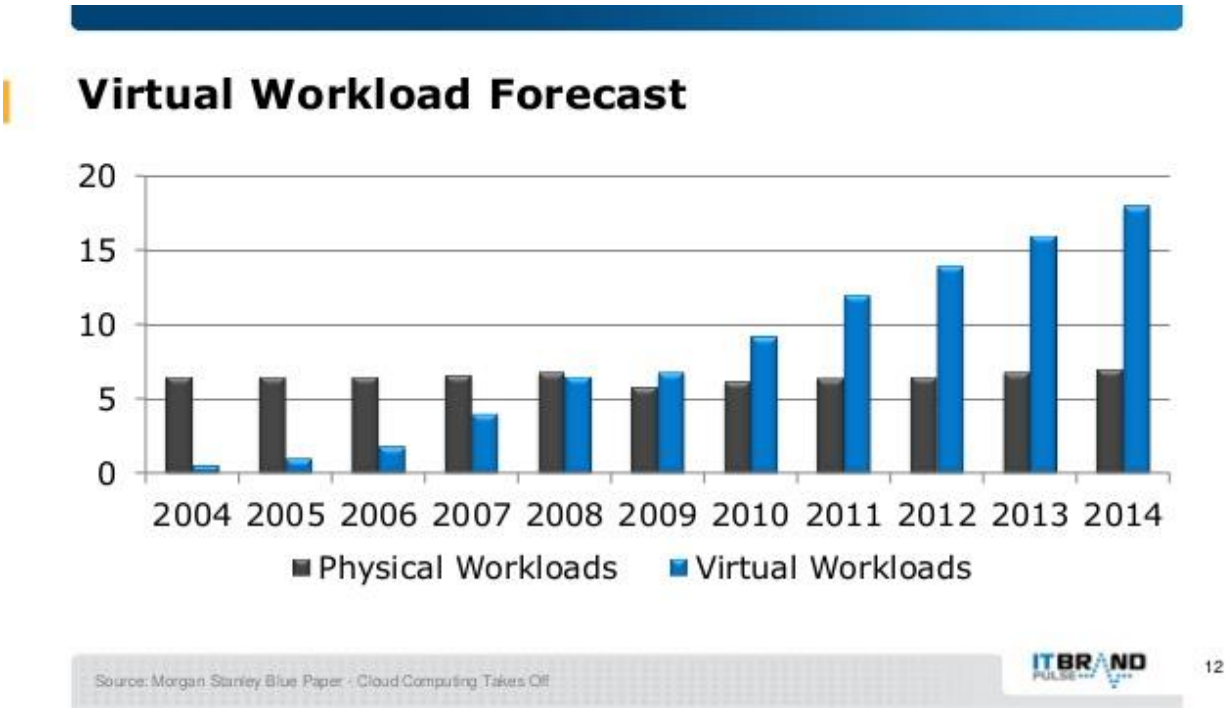


Figure 12 Physical and Virtual Workload Growth.

1.5.4 Storage Virtualization

Storage Virtualization in Data Center is primarily responsible for abstracting the internal operations of storage systems from network and applications to have a storage system that is independent of network. Some key issues that Storage Virtualization solves is failure recovery in SAN, good QoS with high performance and high utilization rates of storage array resources. Also it provides with easy administration by automating tedious administration tasks.

Storage Virtualization is created by adding a new software and/or hardware layer between storage system and server. For administrators, it feels like managing single consolidated resource. Applications are open up to multiple virtual instances to be used and hence high availability of storage resources is obtained which is crucial in this age of BigData. Using storage virtualization, unallocated storage blocks can be joint together into LUNs which can be assigned to new applications that need storage access. When a storage system is to be taken down for maintenance, such LUNs can be the source of data thus facilitating migration and replication. It also allows easy implementation of dynamic and thin provisioning to assign and shrink LUNs as per application needs.

Storage Virtualization is achieved in three different ways [24]:

1.5.4.1 Host-Based Virtualization

This type of virtualization has a software layer which is placed above the device driver handling the I/O requests and responses and retrieving of metadata. A typical Host-Based virtualization solution is shown in **Figure 13**.

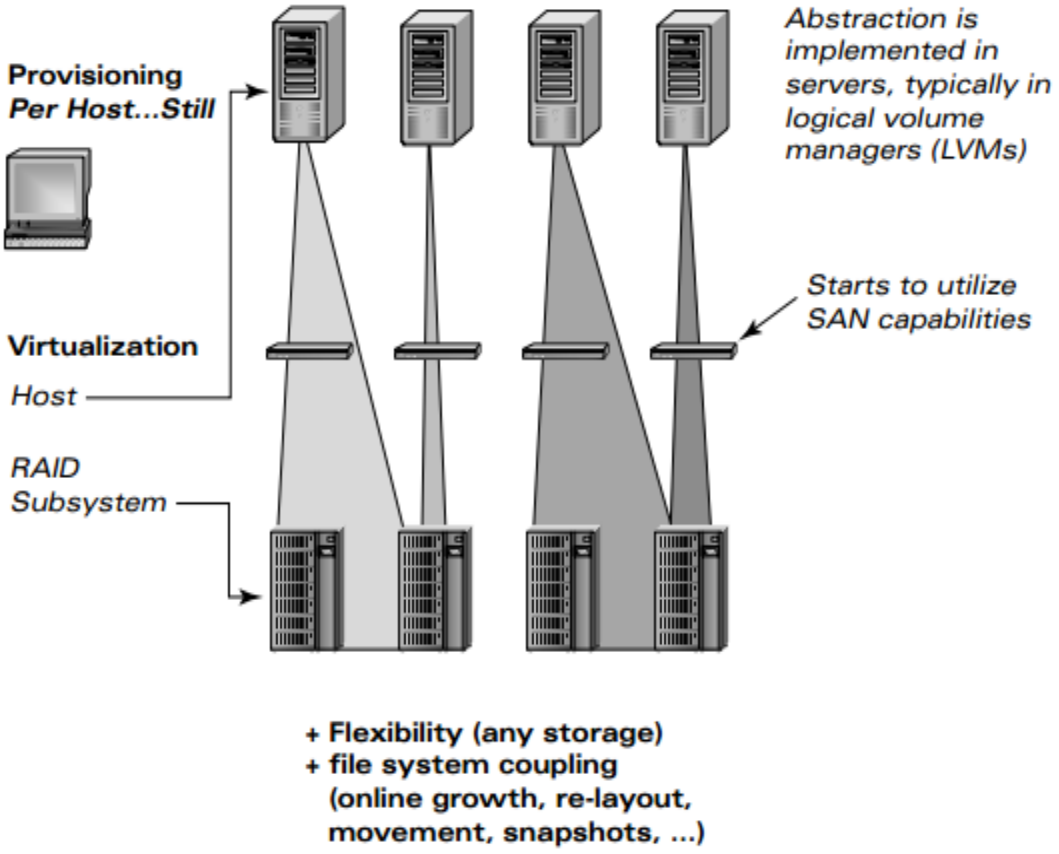


Figure 13 Host-Based Virtualization
Source: http://www.snia.org

This type of virtualization is most common because it is easily adoptable with Direct Attached Storage which is still very popular. It aggregates multiple LUNs to form a LUN that looks like a disk drive available for use by OS. Managing storage resources tightly coupled with OS is easier. However being server-centric, each host will require provisioning making it slow and tedious task.

1.5.4.2 Storage-Based Virtualization

Storage-Based virtualization also called Subsystem-Based virtualization is host independent and allow heterogeneous host connect to storage arrays irrespective of their OS and application accessing data making it highly suitable for both SAN and DAS environment. **Figure 14** shows typical Storage-Based Virtualization implementation. Hardware specific features can be easily adjusted for use with Storage-Based Virtualization for e.g. Caching. Many a times, both Host and Storage Based Virtualization are combined to achieve better performance because it is only host

which is best aware about completion of any I/O operations. Virtualization is built onto single fabric and a storage controller will handle all data requests and metadata.

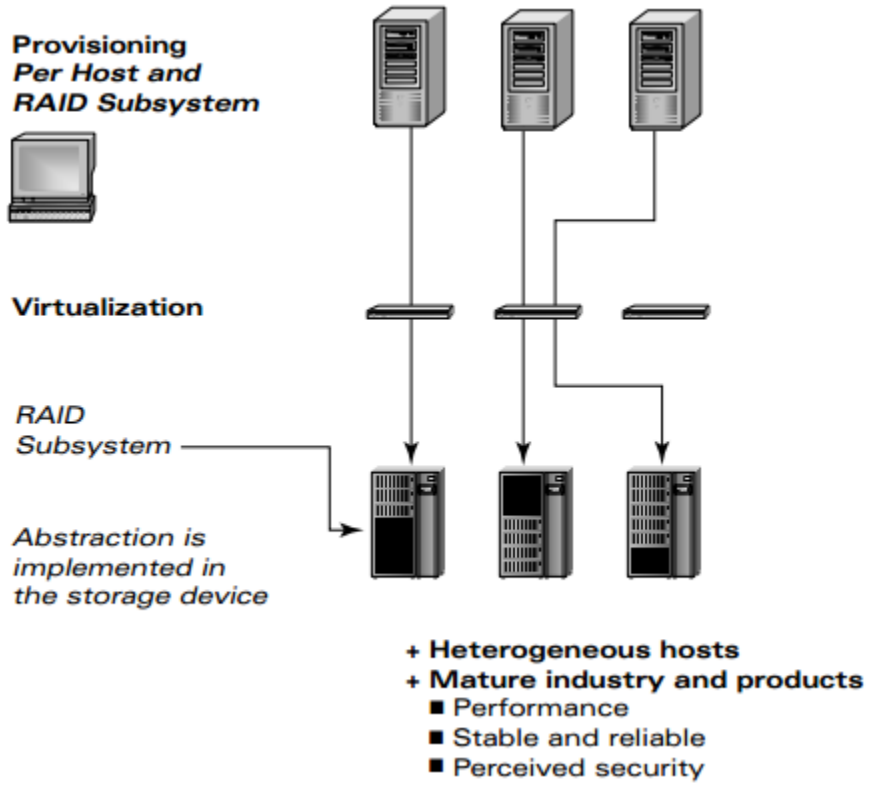


Figure 14 Storage-Based Virtualization
Source: http://www.snia.org

The only disadvantage of this technique is isolation to single storage system. Thus recovering in case of hardware failures is expensive. Even if it allows virtualization to multiple storage arrays, they are vendor-specific and vendor-locked in.

1.5.4.3 Network-Based Virtualization

Probably the latest feature and most acceptable solution of Storage virtualization in Data Center, Network-Based Virtualization views the storage array as a network device mostly used as Fiber Channel connected to SAN. It has the best chance to meet exponential growth in storage capacity and dealing with most heterogeneous data. Network Based Virtualization offers some great features like growing and shrinking LUNs for hosts to access, synchronous and asynchronous replication within SAN and over WAN links and secure access to LUN by a verified host. It also supports caching, on demand storage and QoS.

1.5.5 OTHER VIRTUALIZATION IN DATA CENTER

1.5.5.1 Application Server Virtualization

It is synonymous to reverse proxy load balancing where an application or service is accessed by different applications transparently. It provides access to end user using virtual interface and

load balance all the servers in back end. It mimics itself as web server by providing a virtual IP to end user which user thinks as connecting to web server itself. This type of virtualization is generic and can be applied to servers, storages and any system in Data Center [25].

1.5.5.2 Application Virtualization

Application Virtualization is outcome of Thin Clients. Most important examples of this type of virtualization are SaaS, browser based apps and terminal services by Microsoft.

1.5.5.3 Management Virtualization

Management Virtualization is somewhat like creating different users and access passwords in mail and web servers. Allowing per user policies and differentiating between administrative roles amongst different kind of users is prime task of management virtualization. In Data Centers, one can think of management virtualization as allowing access for routers and switches to network administrators but not providing with admin access to servers.

1.5.5.4 Hardware Virtualization

When we talk of hardware virtualization, we are sketching the role of CPUs in data centers. Multiprocessing in CPUs can be thought of virtualization as the process will only request processor time and leave it up to scheduler to decide which part of CPU and what amount of RAM is to be assigned. Hardware pre-allocation also provides some degree of virtualization by slicing out some proportion of CPU resources for specific activities or computations.

1.5.5.5 Service Virtualization

Service virtualization encompasses all the virtualization discussed by accessing a particular technique as the application gets delivered over the network. This may involve load balancing of servers, SOA requests through gateways and storage access over multiple instances. For user the virtualization at different levels is carved out and it only sees the application being accessed successfully or granted access to.

1.5.6 Less Cabling with FIBRE CHANNEL, FCoE, iSCSI and InfiniBand

The Ethernet standard kept evolving to meet high speed requirements in computer networks leading to some exciting technologies. One of such innovation was Fibre Channel. Today all SAN network traffic goes through fibre channel allowing 16Git/s of high speed transfer. Each Fibre channel consists of World Wide Node Name (WWNN) and a World Wide Port Name (WWPN) for each of the port on device. Two different type of implementations are found in Fibre Channel which are Switched Fabric and Arbitrated Loop.

Fibre Channel over Ethernet (FCoE) is a standard protocol developed for transmitting Fibre Channel frames by encapsulating them into Ethernet frames. However, data transmission failures are but obvious when using Ethernet questioning the further expansion of use of FCoE. Data Center bridging was developed (discussed in next section) to help promote use of FCoE in Data Centers of all sizes. FCoE helps reduce cabling, NIC use and overall infrastructure development cost [15].
FCoE has been widely referred as industry standard solution for I/O consolidation by using in

place well-defined Ethernet standard. Ethernet is generally used in low latency environment for transferring small chunks of data while Fibre Channel is used for I/O operations like regular backups, file servers and huge databases. A separate Host Bus Adapter and NICs are required to connect Fibre Channel to virtual hosts on the servers. FCoE allows integration of 10GE Ethernet and FC on a single network adapter allowing less physical medium and better server utilization. This also significantly decreases the number of ports used and amount of power consumption and cooling required.

The FCoE being encapsulated in Ethernet Frame requires some mapping with OSI layers. The **Figure 15** below shows FC stack, FCoE mapping and its correlation to OSI layers. It allows carry forward FC-2 layer inside Ethernet Layer hence allowing FC-3 and FC-4 to be carried into Ethernet Frames.

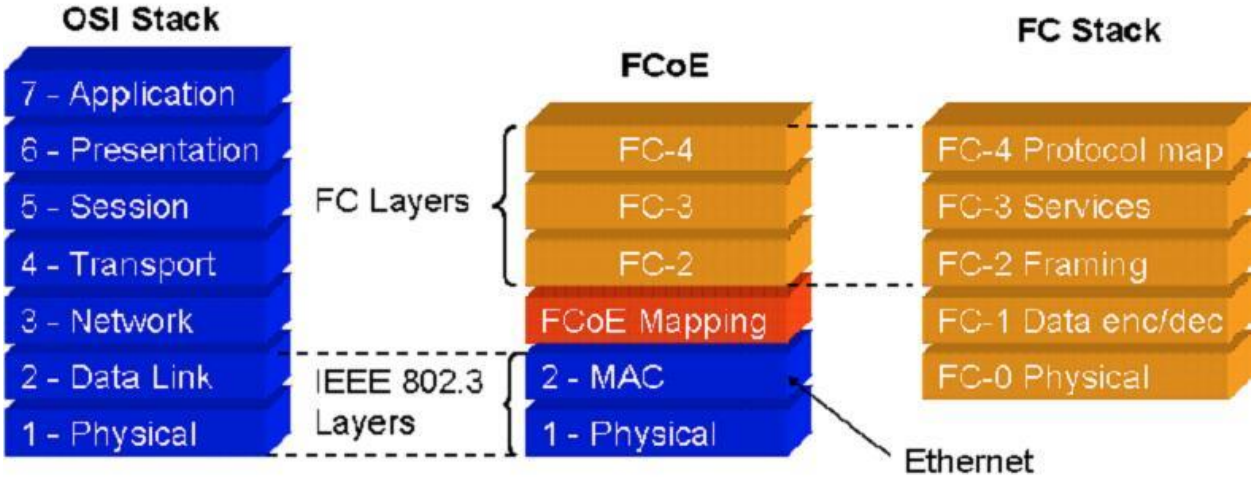


Figure 15 FCoE Mapping to Ethernet

The biggest hurdle in encapsulating the FC frames inside Ethernet frames is the lossless characteristics of FC. Ethernet uses drop packet flow control and FC frame has to manage congestion using link level credit based flow control. To accommodate this, the PAUSE feature of Ethernet frame is used and a control frame is send by receiving port to indicate that it is busy and cannot accept new packet as of now. Thus such network traffic convergence is achieved by designing FCoE switch that could unify both FC and Ethernet frames on single network fabric. Following shows how a simple FCoE switch would function. As shown, FCoE switch will comprise of both FC and Ethernet switching functions. The switch is designed with FCoE entity which is responsible for encapsulating and de-encapsulating between FC and FCoE frames. It has

Ethernet MAC address for traversing the FCoE frames over Ethernet [26].

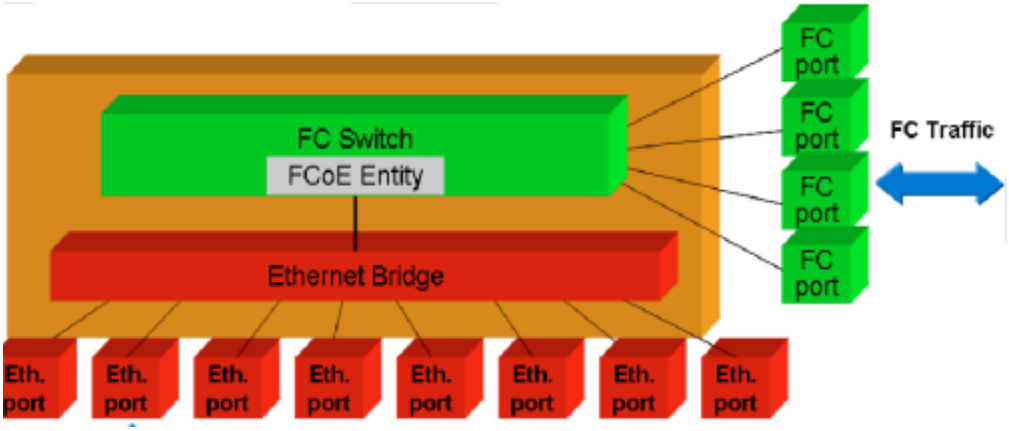


Figure 16 Sample FCoE Switch
Source: www.fibrechannel.org

iSCSI (Internet Small Computer System Interface) was developed to connect storage systems to network and use Ethernet frames for sending SAN traffic. iSCSI allows integrating SCSI and TCP/IP protocol for storage device data traffic. iSCSI provides high level of security and authentication by encrypting data over the network and the data access appears to user as normal local hard disk access. iSCSI is popular for small organizations as it can be accomplished just using the old Ethernet switches unlike FCoE where lossless switches are needed. FCoE is taken into consideration for large businesses with virtualization in place and high I/O requirements.

InfiniBand similarly like iSCSI allows high throughput and low latency by allowing high speed data flow between processors and I/O devices. It supports up to 64000 addressable devices and comes along with QoS and failover. Data is transmitted in packets of 4k size and communication is done through channel adapters which are termed as Host Channel Adapter (HCA) for processors and Target Channel Adapter (TCA) for peripheral device. It uses a serial bus and multiplexes data over multiple channels [27].

1.5.7 DATA CENTER BRIDGING.

With protocols such as FCoE and iSCSI accelerating the innovations in network convergence, there was a strong need to formulate the standards for incorporating lossless traffic inside Ethernet. This gave birth to Data Center Bridging (DCB) Task Group which in turn defined four different technologies to have DCB in place which are 802.1Qbb for priority-based flow control, 802.1Qaz for Enhanced Transmission Selection, 802.1Qau for Quantized Congestion Notification and Data Center Bridging Exchange Protocol (DCBX) [28] [29].

The idea behind DCB is to connect devices present on SAN network and Ethernet network. DCB enhances Ethernet standard to allow lossless storage traffic to be encapsulated within Ethernet frames. DCB along with the four proposed standards solves the problems of network convergence like allowing lossless transport within Ethernet frames, reducing equipment and

power cost by converging traffic on single network fabric, easy maintenance and management of single network fabric and traffic control by fine-tune bandwidth allocations to different types of traffic [30].

1.5.7.1 Priority-Based Flow Control: IEEE 802.1Qbb

When combining different types of traffic on single link, issues like blocking of certain traffic by other overwhelming traffic, high latency for one traffic when optimizing the other and sudden drop in bandwidth for a certain traffic when other traffic uses link for large bursts of data are common. DCB has modelled flow control around Ethernet Frame's PAUSE attribute. It does so by pausing traffic based on user services and priorities. Physical link is divided into 8 virtual links so that pausing traffic on one virtual link does not affect the other. Also, one virtual link can be used to carry lossless traffic while other can be used to carry packet-drop IP traffic.

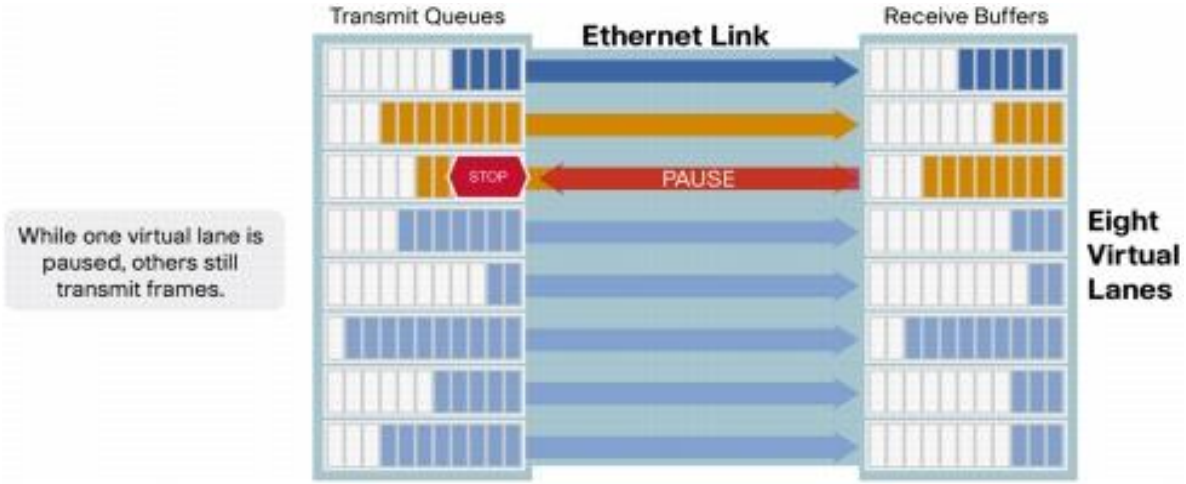


Figure 17 Priority Based Flow Control
Source: http://www.cisco.com/c/dam/en/us/solutions/collateral/storage-networking/mds-9506-multilayer-director/white_paper_c11-647658.doc/_jcr_content/renditions/white_paper_c11-647658-3.jpg

1.5.7.2 Enhanced Transmission Selection: IEEE 802.1Qaz

ETS deals with class-based bandwidth allocation for virtual links created by PFC. It provides prioritized processing by using bandwidth allocation, low latency or best effort. NIC provides virtual interface queues for each traffic class and each interface queue manages its own allocated bandwidth. ETS also allows differentiation of traffics within same class.

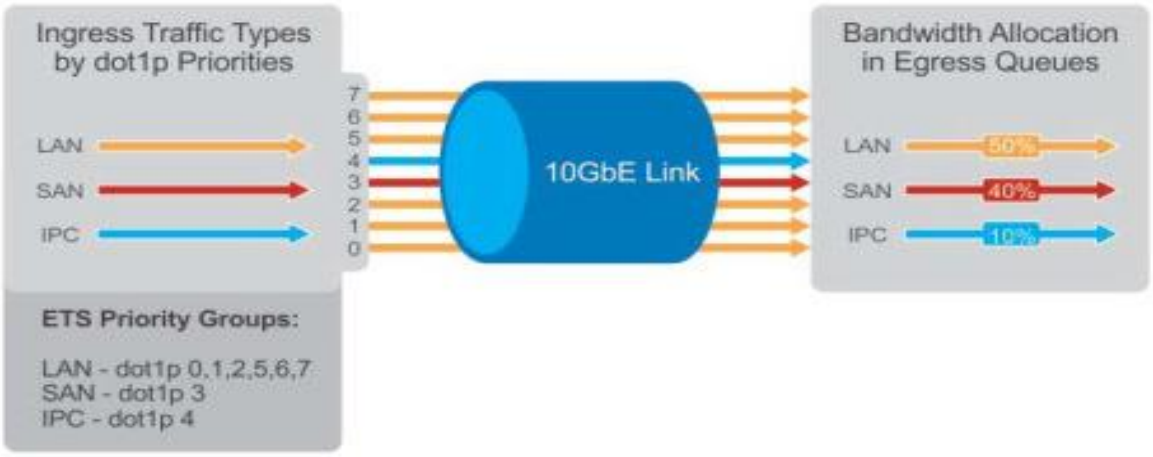


Figure 18 Enhanced Transmission Selection
Source: https://hasanmansur1.files.wordpress.com/2012/12/dcb-ets.jpg?w=459&h=227

1.5.7.3 Data Center Bridging Exchange Protocol

It is discovery and capability exchange protocol designed by DCB task group to allow exchange of configurations between DCB-compliant bridges and initiate peer discovery when needed[4]. Some core features of this protocol are DCB peer discovery, link configuration, PFC, ETS prioritization of traffic class, Congestion notification and NIC virtualization.

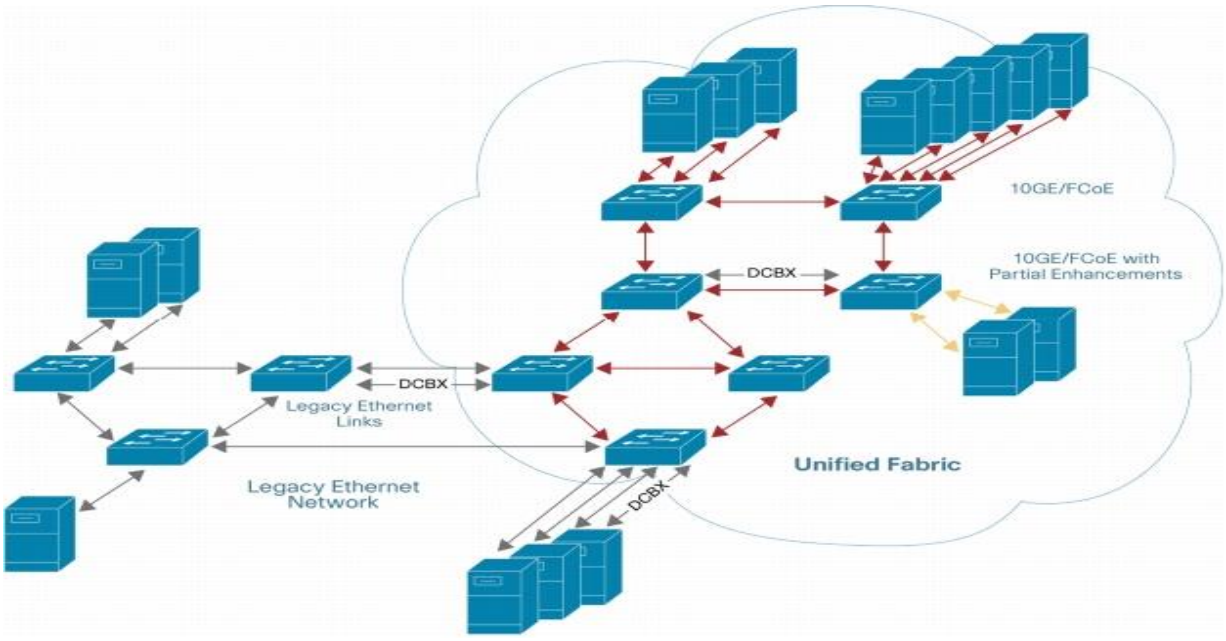


Figure 19 Data Center Bridging Exchange Protocol (DCBX)
Source: http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/ieee-802-1-data-center-bridging/white_paper_c11-462422.doc/_jcr_content/renditions/white_paper_c11-462422-4.jpg

[4] http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/ieee-802-1-data-center-bridging/at_a_glance_c45-460907.pdf

1.5.7.4 Congestion Notification: IEEE 802.1Qau

It is standard define to provide congest control in converged network so that only source of congestion is affected and rest of the network functions as usual. Congestion is measured at congestion point and rate limiters are imposed to shape traffic flows. This is achieved by having an aggregation layer switch sending control frames to access layer switch to lower its traffic flow so that network's core remains intact amid congestion [31].

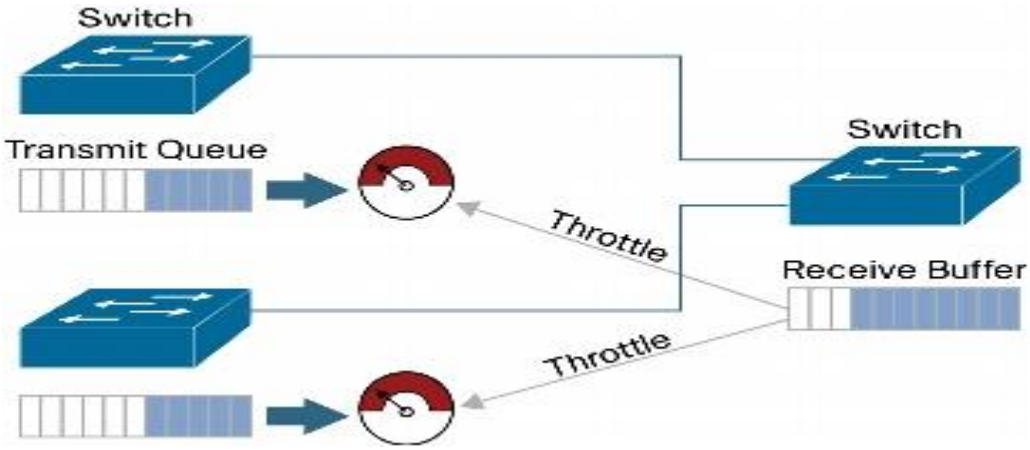


Figure 20 Congestion Notification in DCB
Source: http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/ieee-802-1-data-center-bridging/white_paper_c11-462422.doc/_jcr_content/renditions/white_paper_c11-462422-5.jpg

1.5.8 MULTITENANT VIRTUALIZED DATA CENTER

There is subtle difference between virtualization and multi-tenancy and has been always debated by industry experts. The key difference lies in what physical server is providing as multiple access to shared resource. In virtualization, a single server is able to host multiple instances of various server environment allowing user to host an environment of choice, configure it and decorate it with applications and OS of their choice. While on other hand multi-tenancy is about providing multiple instances of same application hosted on physical server to multiple users.

A simple example of multitenant application on a physical server of data center is shown in Figure 21.

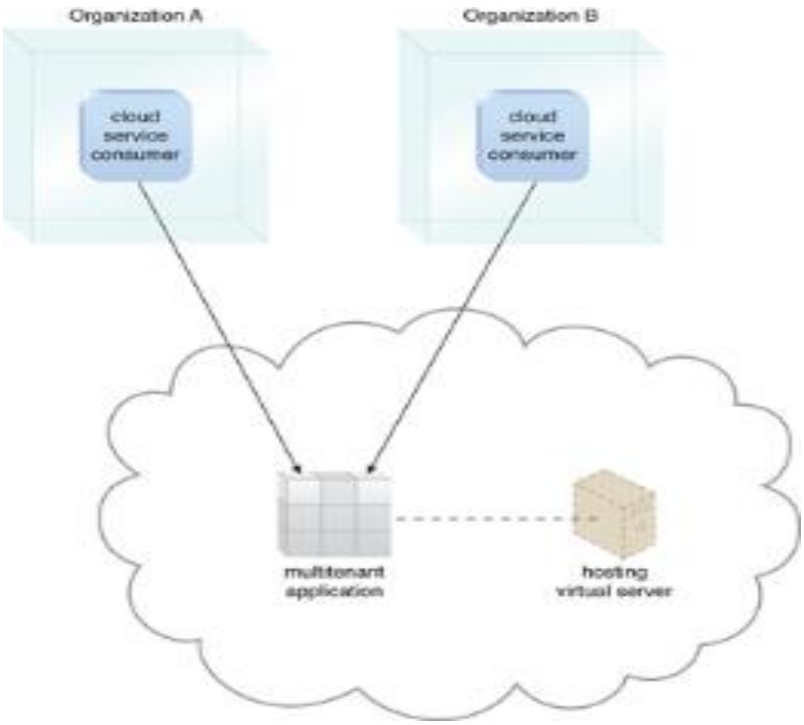


Figure 21 MultiTenant Application
Source: http://whatiscloud.com/multitenant_technology/index

Each tenant can customize application features like user interface, rules and logic of application workflows or business processes, exclude, include and rename data structures of application and manage access rights and security towards applications for certain groups and users. The security levels should be kept in mind as artifacts for a user like schema, data and middleware should not be known to other users. Multitenant Data Center allows features like usage isolation, data security, easy upgrades, on demand monitored usage, data tier isolation and scalability to name a few[5].

The combination of enterprise data centers and virtualization on every data center resource has led to widespread welcome of Multi-Tenant Data Center (MTD). Server and Storage virtualization has been successful in bringing technologies over decades now. However network virtualization is something that needs more efforts to streamline the process of creating commercial MTD infrastructure. Tenants wants easy configuration of their VMs in cloud just as they would configure their local enterprise network while provider wants to achieve all the requirements of tenants and also use network configurations and topologies of their choice. VMware has achieved this using a SDN controller and a network hypervisor executing on top of this controller using datapath enhancements [32].

In a MTD with virtualization, number of hosts connects to network with their multiple VMs supported by host hypervisor. The virtual switch in hypervisor accepts packets to and from VMs

[5] http://whatiscloud.com/multitenant_technology/index

and forwards them to a VM or a different hypervisor. This hypervisor is supposed to provide abstraction to VM for MTD to function. In a traditional data center with only server virtualization, network consists of multiple L2 domains which includes a Top-of-the-Rack switch connecting to aggregation switch. Such topologies have a serious threat of over subscription which is dealt with by adding more and more ports to aggregation switch. When network virtualization is applied to such network, generally an Ethernet trunk is carried over from ToR switch to virtualized server [33]. However, the VLAN span is restricted to size of L2 domain. L2/L3 aggregation switches provides with access lists and handles inter-VLAN traffic as shown in Figure 22.

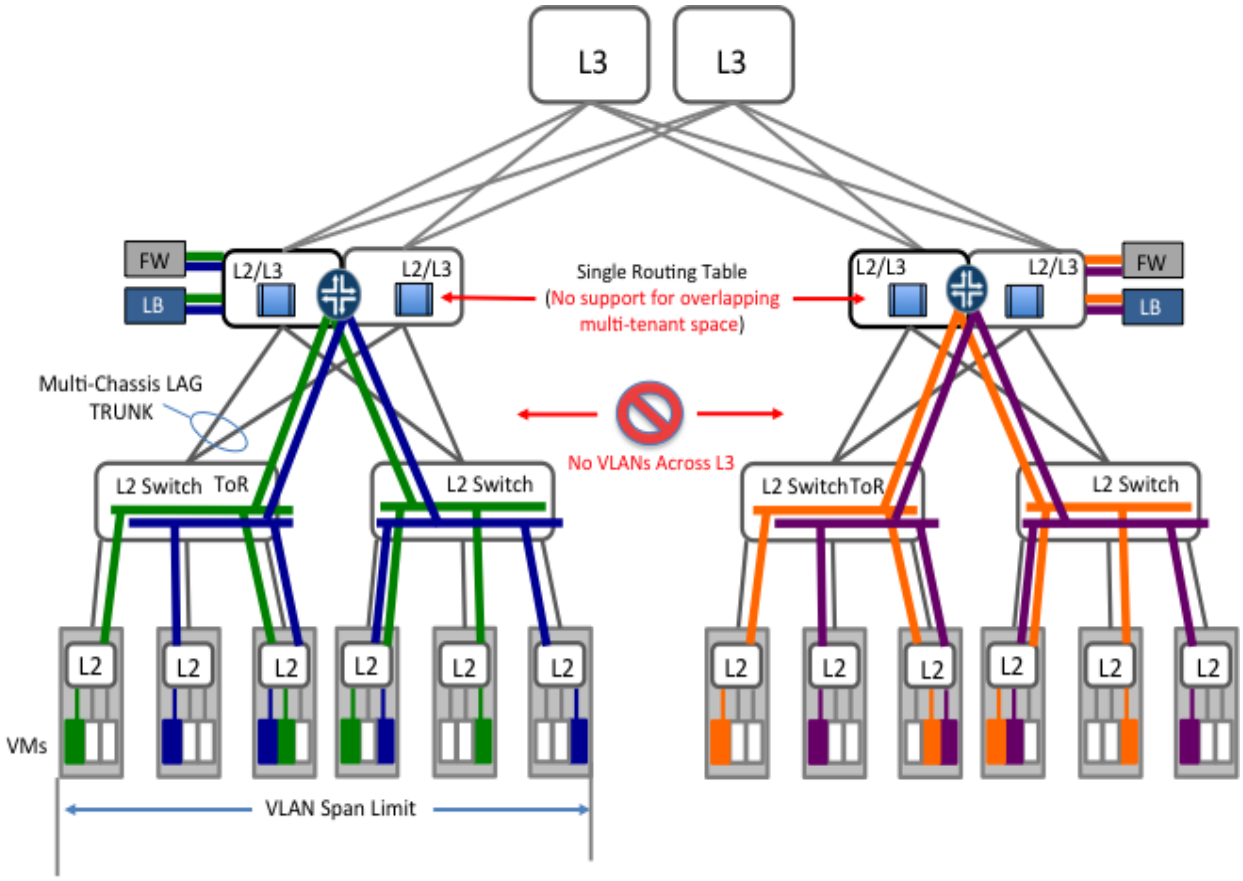


Figure 22 Legacy Data Centers with Network Virtualization
Source: http://www.opencontrail.org.

To support multi-tenant in such infrastructure, solutions like VRF-Lite or VRF with MPLS Tunneling can be used to guarantee separate routing and address space for each tenant as in Figure 23. This will allow a tenant to span over more than one L2 domain. L3 routing is applied this days to ToR switch as L3 protocols are more suitable for such topologies and can allow easy spanning over different data centers. However due to L2 domains, techniques like VXLAN is necessary to encapsulate Ethernet frames in IP Packet. But virtualized servers typically run L2 Switches and thus the traffic leaving the L2 domain has to send to gateway router or a router running inside VM as software.

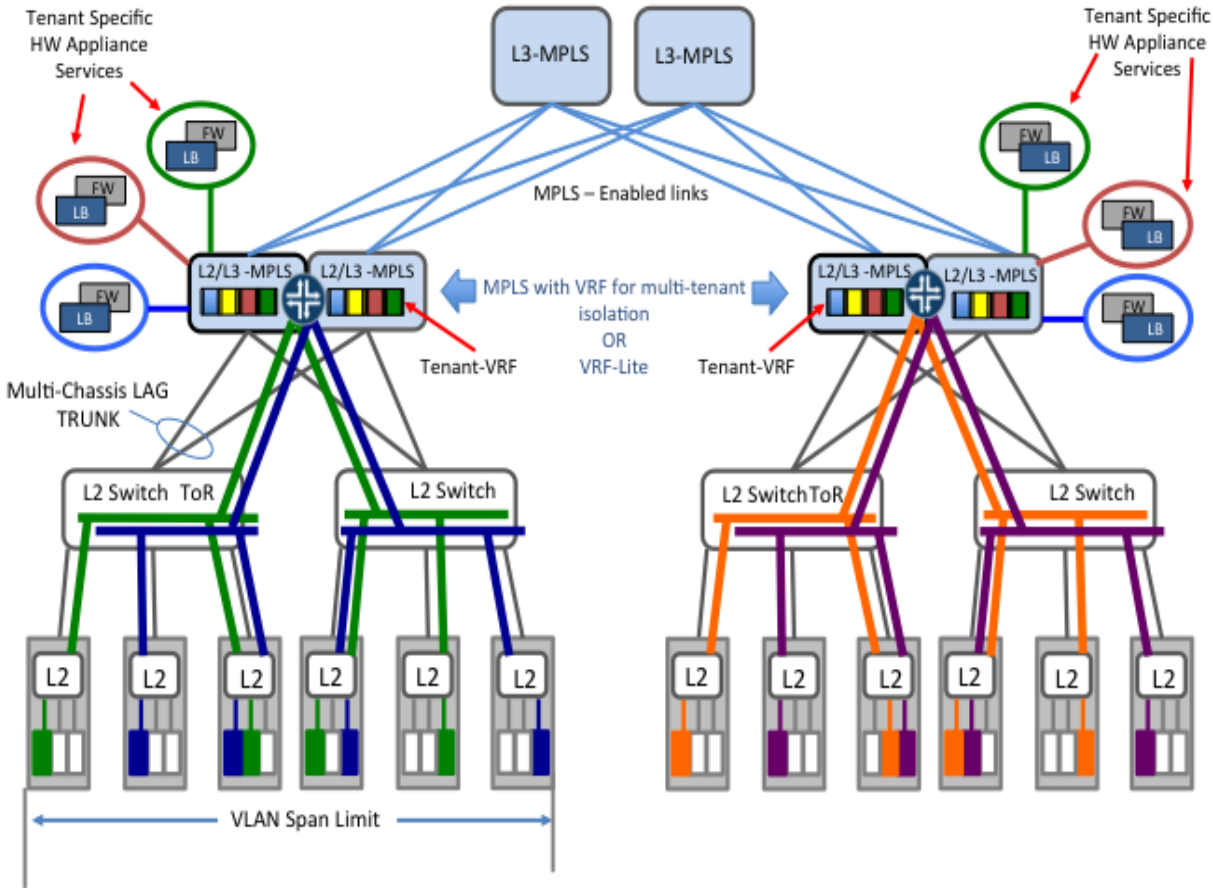


Figure 23 Multi-tenancy in legacy data centers.
Source: http://www.opencontrail.org.

It is also possible to design multi-tenant datacenter that can connect other such datacenters using L3VPN infrastructure using the above mentioned gateway router to connect with other datacenters. The idea is to remove multiple instances of software routers. Instead the kernel module of virtualized server has the capabilities of multi-VRF router (MX) such as packet filtering, network address translation and allowing access to external network or a different data center. The fabric that connects this data center is still IP and thus a L3 router gateway or L3VPN can be used to access external network resource as can be seen in **Figure 24**. Using such infrastructure it is possible to design and implement cloud network infrastructure that gives the best of both the world viz. Multi-tenancy and Virtualization and can be easily scaled out to meet business growth.

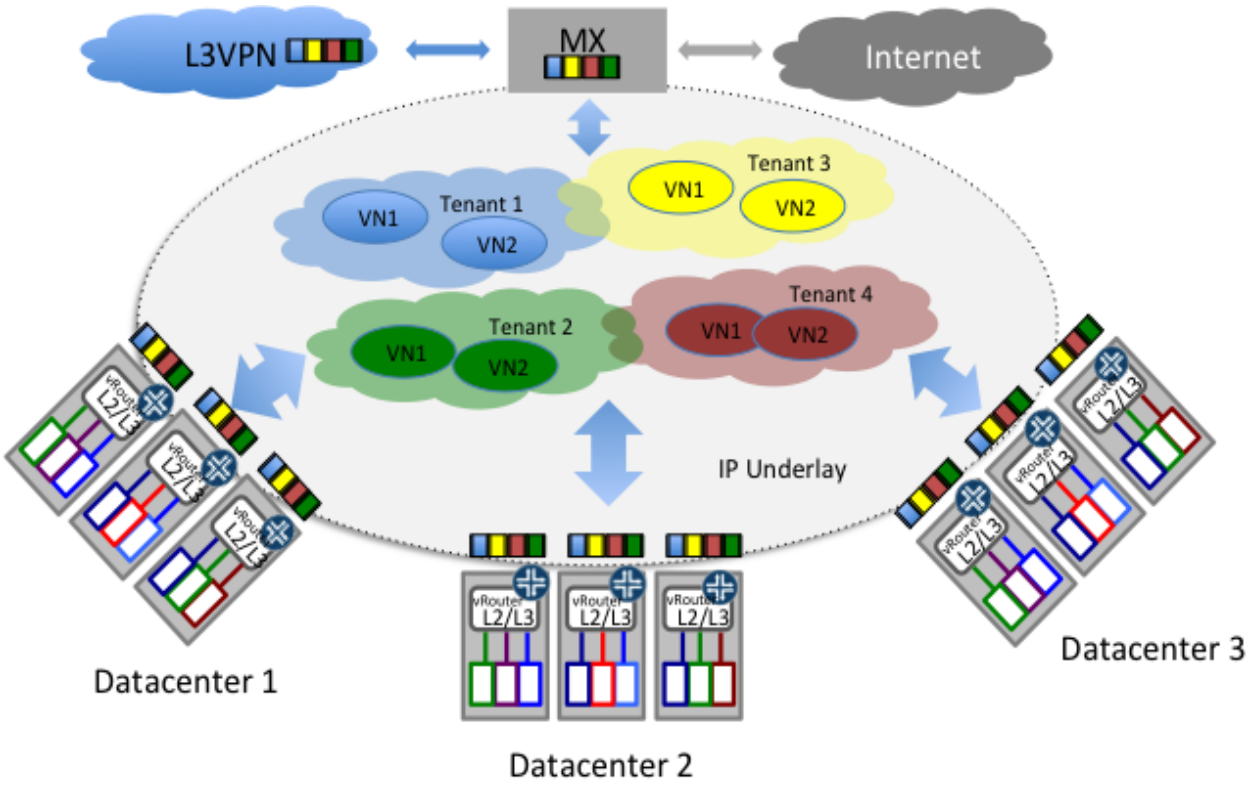


Figure 24 Multi-Tenant Virtualized Multi Data Center.
Source: http://www.opencontrail.org.

1.5.9 Challenges of Virtualized, Multi-Tenant Data Center

Definitely virtualization have brought in agility and cost efficient solutions to network infrastructure. But by adapting virtualization in data centers, we also challenge the traditional data center and security design principles. The increasing demand of high bandwidth, end users and real-time applications has touted virtualization as the most positive aspect towards graceful data center growth. But virtualization brings some new challenges towards a robust data center design.

- More and more VMs are being hosted on lesser physical servers to ease provisioning, deployment and maintenance. Technology like live migration makes it easier to balance the workload and move VMs between the servers in data center. But it is quite common to overuse server's memory and I/O resources without proper planning of capacity and workload needs of applications. These may directly impact one or all the VMs. Thus it is quite important to plan the data requirements and resource needs of each VM before deploying them into production.
- Virtualization is a win-win situation for network administrators. Provisioning new VMs was never so easy and quick for administrators. But if the administrators lacks coordination between themselves in provisioning new VMs without considering resource requirements, they could overburden the infrastructure with unnecessary VM sprawl.

- Uninvited and sudden network failures cannot be underestimated. VM mostly resides on Storage Area Network (SAN) and are loaded into server's memory during startup. Thus protecting these snapshots, having incremental backup plans, replication of snapshots will require SANs to reserve additional storage. It is like making reservation plans of storage requirements within the storage systems. Also, with real-time application getting more widespread, recovery plans needs to be as fast as possible adding more challenges to virtualized infrastructure.
- The research and development in server technologies with growing leaps and bounds. There are servers in market which would give organization the best combination of CPU, memory and I/O resource utilization in a virtualized infrastructure. But the networks lags behind the servers in providing robust solutions. With ever going traffic from each newly added VMs in the environment, switching methods and network performance starts saturating and degrading.
- Virtualization has enforced development of some very sophisticated management tools for server management. It is up to the senior administration team to decide on which Management Tools suits their virtualized infrastructure the best. As provisioning VMs and resources is getting more economical, the IT team needs to careful review the requirement of VM and have a streamlined workflow and policy in provisioning them.

Supporting multiple workloads on a server is quite a challenging task in a virtualized environment. It is quite understandable that managing such infrastructure will require extensive planning and management to abide with Service Level Agreements and Quality of Service levels [34].

2 SDN FRAMEWORK AND SOFTWARE-DEFINED DATA CENTER SOLUTION TO OVERCOME ISSUES OF MULTITENANT, VIRTUALIZED DATA CENER. SDN FRAMEWORK CHALLENGES, USE CASES AND APPLICATIONS.

2.1 SDN FRAMEWORK AND SOFTWARE-DEFINED DATA CENTER.

2.1.1 NETWORK VIRTUALIZATION and RISE TO SDN

Before diving into the SDN concepts, it is highly necessary to relate it to network virtualization that has been discussed and how together both of these technologies provides with network agility and programmable networks.

Network Virtualization as being discussed automates the whole process of network change requests by allowing VMs to easily move between the logical domains. It creates the logical network segments by dividing network at flow level. It is an overlay that saves administrators from physically connecting each new domain request by creating virtual segments on top of physical infrastructure without blocking the physical network operations itself. Networking virtualization is however more sought after by decoupling advantages of control and forwarding plane that SDN offers. Network Virtualization is also important in Multi-tenant infrastructure as

it allows creation of virtual networks that separates the traffic of multiple tenants. Thus network virtualization offers easy management of networks and automated framework by virtualizing physical network internally or externally. In internal virtualization, VMs on host share data without intervention of external network. In external virtualization, VLAN is aggregated from physical LANs or a physical LAN is broken into multiple VLANs. It is really difficult to define Network Virtualization and how it differs from SDN but according to Gartner, Network virtualization is the process of combining hardware and software network resources and functionality into a single virtual network. This offers access to routing features and data streams that can provide newer, service-aware, resilient solutions; newer security services that are native within network elements; support for subscriber-aware policy control for peer-to-peer traffic management; and application-aware, real-time session control for converged voice and video applications with guaranteed on-demand bandwidth[6].

Inspired by wave of network virtualization in data center, SDN came out as a technology to control the whole network that was just virtualized by NV by a scalable software controller. SDN achieves this by separating the control plane (part of a networking device that decides on how to control the traffic using protocols) from data plane (part of a networking device that forwards the traffic from network to network). It also controls all the data plane from a single point of reference using APIs – which is a software controller residing on server. The most famous SDN solution for designing such centralized controller is OpenFlow which uses tables consisting of rules for handling incoming traffic, finds a matching rule and respond it back to data plane to forward it to destination or next hop. OpenFlow has inspired many other controller-based tools to support SDN and create networking applications like load balancing, firewalls, NV, VM migration, etc. One of the consortium that drives SDN development and motivates standardization is ONF (Open Networking Foundation).

SDN is recent idea of creating programmable networks. However, SDN is outcome of two decades long history of creating programmable networks. Network Virtualization however remains the key use case of programmable network since its inception. Figure 25 below shows the rich history of creating programmable networks and how it was driving force in innovations of SDN [35].

[6] http://www.gartner.com/it-glossary/network-virtualization

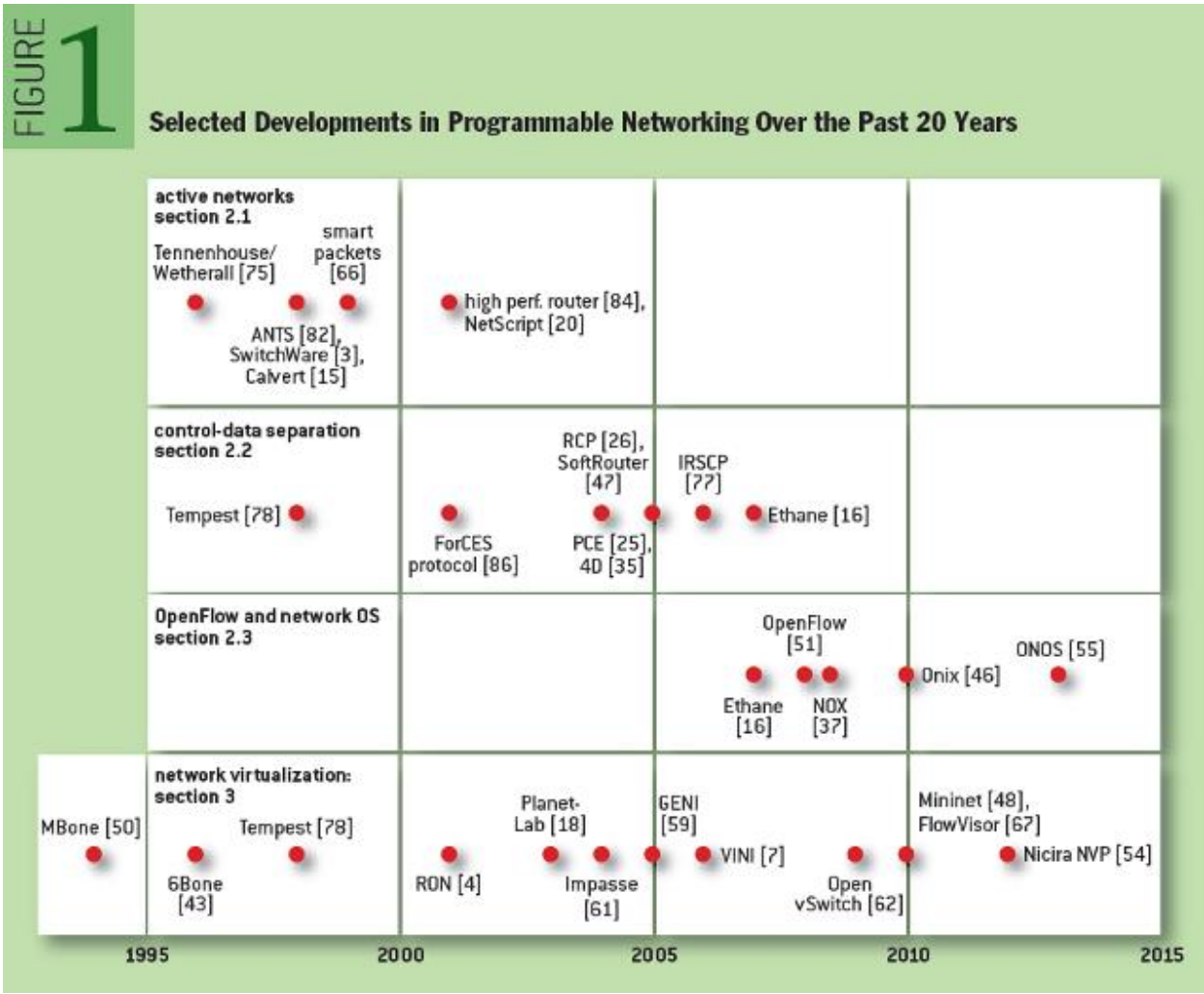


Figure 25 Programmable Networks Over two Decades.
Source: http://queue.acm.org/detail.cfm?id=2560327

The earliest inspiration of creating programmable networks came from telephony networks which then separated data and control plane to create new reliable services. In 1990s, researchers would design a protocol in small campus network and then migrate it to larger networks and then apply for standardization. However, back then this process was super slow and cumbersome. Researchers came out with an idea of providing network resources through API and then test those APIs on packets passing through real network. This workaround was then termed as Active Networking. Active networking basically had nodes that performed custom operations on packets passing through that nodes or add new information to those packets. Active networking was incorporated using two different methods: Capsule-based system [36] and programmable router/switch model [37]. Capsule-based system carried code to be executed at nodes in data packets itself while programmable router/switch model used out-of-band methods. Technologies that inspired Active networking were new programming languages, reduction of computing cost and trends in virtual machine setups. Active Networking went on to be partially funded by U.S. Defence Advanced Research Projects Agency. Active

Networking also had high user acceptance because of urgent needs of reducing network provisioning time which then took months, having network design that meets changing demands of newer applications and easy management of other network equipment like firewall and proxy servers. Active Networking brought important contributions that still inspires the development of SDN which included programmable networks, reduction of overall physical network assets and virtualization of network by using codes that operates on data packets.

In early 2000s researches started interacting with backbone networks more often to test protocols they were designing for efficient network management. The internet traffic was booming and overwhelming network operations and management. There was a need to simplify this operations by decoupling the network operations from underlying hardware to ease debugging and configuration. This brought in different approaches to separate the control plane of networking devices from the forwarding plane. Innovation with routers and switches could not cope up with innovation in server stack. Such technology trends brought in efforts from service providers and researches to decouple the control functions from routers and switches. Technologies such as ForCES, Routing Control Platform and Path Computation Element were outcomes of this effort which were standardized by IETF later on. The primary objective of these technology was to enforce network management from single control point and network programmability. Routing control software came into existence by bringing open source networking software into effect removing vendor lock-in for customers. Also, it became easier to replicate the software controller to back up server for easy recovery in case of failure.

Doubts were raised on efficiency of Programmable Networks and SDN in steering real world deployments. OpenFlow came to rescue by developing APIs that could be leveraged to perform almost any routing functions from a software controller. OpenFlow was designed in such a way that it could be immediately used on any switch or router hardware. OpenFlow is a protocol with set of rules, prioritization between these rules and corresponding actions for each rule. It maintains all these in so called Flow Table. This allowed vendors to easily accept OpenFlow standards without making changes to underlying hardware. OpenFlow was first tested at Stanford campus and then WAN capabilities were tested by using OpenFlow at multiple campuses. OpenFlow inspired distributed network state management by ONIX which saved network topology and control information called network information base along with controller software. Thus in general, OpenFlow divided network operations in three layers – control layer that controls and decides where to forward traffic based on current state of network, data layer that forwards the traffic ahead and state management layer that assures integrity and reliability of current state of network.

It is important to know that Network Virtualization and SDN are independent of each other but can do wonders when collaborated to provide network solutions. Multi-tenants in cloud wants to share infrastructure. Each tenant can be provided with virtual switch that can forward traffic between VMs or to VM on other tenant's virtual network. However using SDN, a controller can easily keep track of these virtual networks and install rules on those virtual switch without having

to manage those virtual networks separately. Once network is virtualized, it is easy to mimic a production environment by using SDN solutions like mininet. This tool can be then used to test the controller over mimicked network before putting it through production. Network can be sliced easily and can be tested with different controllers provisioning different requirements.

2.1.2 Software Defined Networking Explained

Software Defined Networking or SDN is an architecture and an open-standard research initiative meant to design networks that are dynamic, scalable, cost-efficient and adaptable in today's computing world comprises high-bandwidth, dynamic and real-time applications. Lower-level network functionalities are abstracted allowing network administrators to have granular control in managing network. SDN achieves this by separating the control plane and data plane of a network. Control plane of a network is analogous to signaling in telephone networks. The primary concerns of control plane is to decide on where to forward a packet, system configuration and management, sharing routing table information with other routers and update the routing tables based on information obtained from other routers. Data plane on other hand deals with forwarding the packets to next hop or destination based on information exchanged from control plane. Thus SDN actually splits out the intelligence of packet forwarding engine of a switch or router to control plane as shown in Figure 26. The control plane is instead a software defined controller which is programmed using an open source protocol – in most cases for now OpenFlow protocol is widely used. This controller can be centralized or decentralized. However, the controller is generally kept centralized so that the network infrastructure can be controlled, monitored and troubleshoot as a single network fabric.

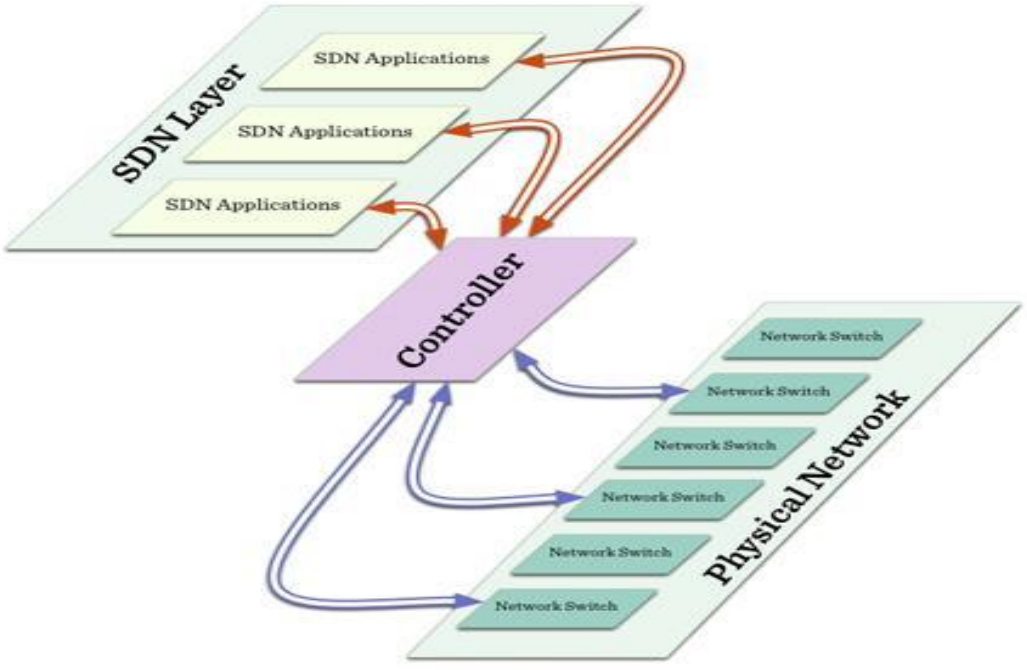


Figure 26 Software Controller in SDN.
Source: http://img.deusm.com/informationweek/2013/11/898899/3-Ferro-NWC-arch.jpg

In traditional architecture, when a switch receives a packet it will make forwarding decisions from the intelligence embedded into the application-centric integrated circuits (ASICs) of the switch. This traditional has been the de-facto standard networking for decades. However with mammoth traffic which is also heterogeneous in nature, network administrators find it difficult to cope up with it without expanding their physical infrastructure. SDN comes handy here by allowing administrators to shape traffic from single, centralized software control instead of configuring every single intended switch. This becomes quite helpful in multi-tenant data center serving virtualized environment to multiple tenants by using commodity switches which are less expensive and providing fast network provisioning. SDN also entertains single switching fabric to be used across various vendor equipment thereby removing vendor lock-in. All the hype of SDN has been quite visible in recent research by Transparency Market Research group which indicates an annual growth rate increase of 61.5% from 2012 to 2018 in adoption of SDN solutions across networking industry [38].

The high-level view of SDN architecture as defined by Open Networking Foundation is shown in below.

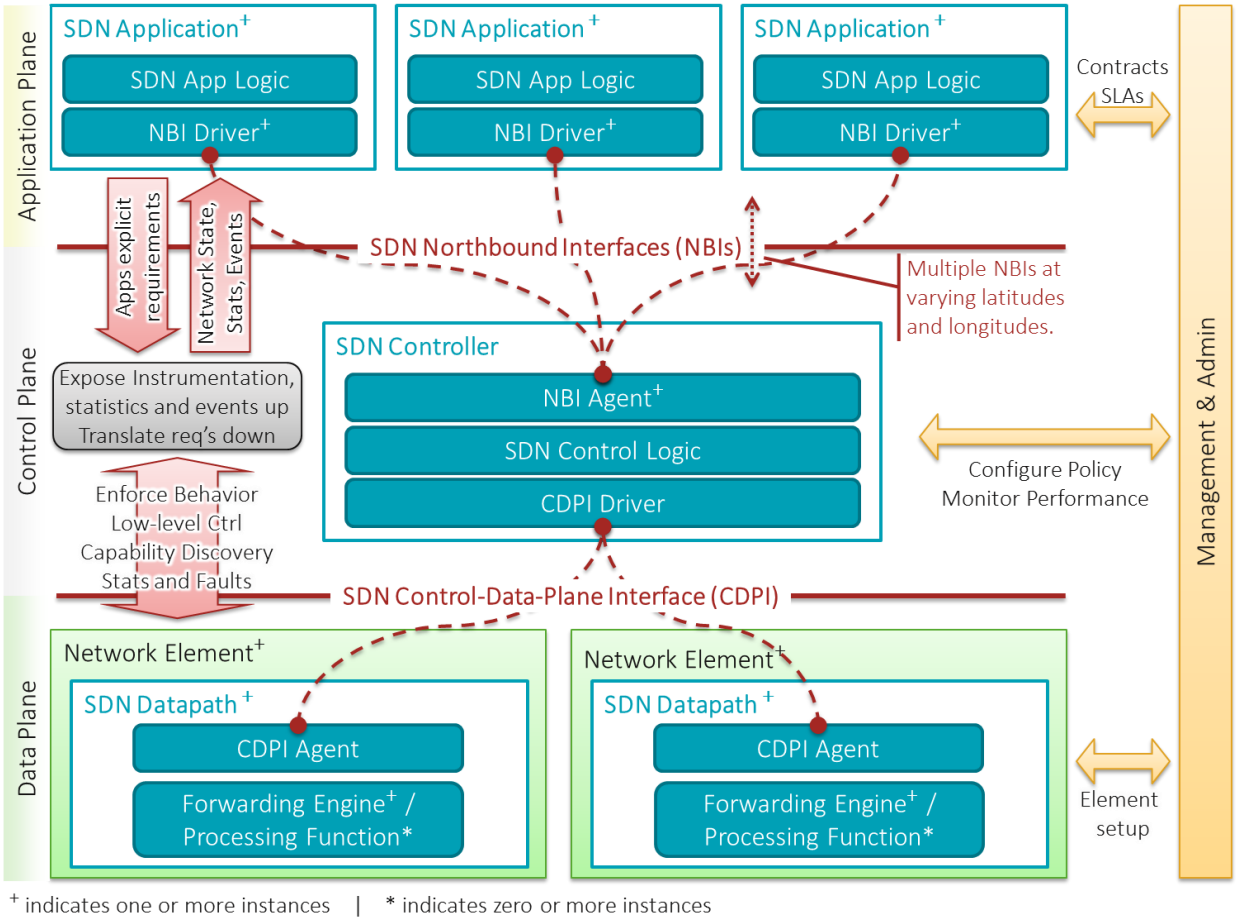


Figure 27 SDN architecture Overview (Open Networking Foundation)

SDN architecture primarily decouples control plane from data plane to build highly scalable and flexible networks with unmatched automation and network control. SDN is enabled using protocols like OpenFlow used for communication between data plane and control plane. It allows centralized management of complete network involving networking equipment from multiple vendors, free the underlying networking infrastructure from automation and provisioning by exposing APIs to perform such tasks, increase the network innovation speed without configuring every single device or waiting for appropriate release from leading vendors. Also, when the network is controlled by centralized software the chances of configuration errors is minimized and if any can be resolved in no time. It also allows policy management at user, application and session levels allowing network changes to be incorporated dynamically based on user needs. The primary components of SDN architecture are explained below [39]:

- SDN Application: SDN Apps communicates the desired network requirements to SDN controller using North-Bound Interfaces (NBIs). They can also request for current network state from controller. The primary components of SDN Application is application logic and one or more NBIs. They may need more than one NBI when they want to be a source of network abstraction to controller.
- SDN Controller: This is the heart of the SDN architecture. It deals with two way data traffic. One way is to take network requirements from SDN Application and forward it down to the SDN Datapaths in data plane. It may also provide SDN Application with network events and statistic data. The SDN Controller is thought to be a logically centralized entity but it may be distributive in nature to replicate itself across multiple servers and to have communication interface between the instances of controllers. Primary components of SDN Controller are Controller Logic, One or more NBI Agents and CDPI driver.
- SDN Datapath: SDN Datapath is logical representation of physical resources. Thus it exposes the actual data processing and forwarding features of networking device like switch and router. It comprises of elements like CDPI for communicating with controller and the actual forwarding engine. Forwarding engine is what helps the data plane make forwarding decisions or terminate a connection. It is possible for SDN Datapath to spread across one or more physical device.
- SDN Control to Data Plane Interface (CDPI): This is one component that makes SDN implementation vendor-neutral. It allows communication between control plane and data plane to programmatically control the forwarding decisions, project the network state, report network statistics and notify about events.
- SDN Northbound Interface (NBI): An interface between SDN application and SDN controller, NBI allows viewing of abstract network structure and issue network requirements to controller based on application needs.
- Interface Drivers & Agents: The both interfaces are actually a combination of driver and agent. Driver is responsible for application side or north facing communication while agent is responsible for network infrastructure or south facing communication.

- **Management & Administration:** The management plane is generally responsible for tasks that does not involve application, control or data plane like client-provider resource management, physical setup of infrastructure, network bootstrapping, etc. However SDN down the line tries to boil down everything into CDPI for a centric-management console for entire network integrated with the controller logic.

Thus SDN allows promoting applications that are network aware in contrast to traditional virtualized architecture in which network is application aware. SDN infrastructure is more approachable to user requirements of delay, throughput and availability which was not possible in traditional architecture and it also allows applications themselves to monitor network state.

2.1.3 Software-Defined Data Center (SDDC)

Software-Defined Data Center is envisioned to broaden the benefits achieved from server virtualization across the Data Center by leveraging the advantages of Software-Defined Networking. It was the single, centralized control of SDN to manage entire network that has motivated the use of virtualization over all sectors of a data center viz. storage, server, compute and security. SDDC is not a product but rather a mechanism to automate and operate entire data center from a single source of control. SDDC will allow all the infrastructure components and hardware devices to be provisioned, managed and operated by API exposed by a software controller residing on server.

SDDC infrastructure has five important characteristics that makes it adaptable in delivering highly automated and centrally controlled network solution [40]:

- **Standardized:** One homogeneous virtualization infrastructure for array of standard networking equipment to assure less complex network.
- **Holistic:** Single infrastructure modification is required to support in place and new applications without touching underlying hardware.
- **Adaptive:** The software provided infrastructure services can easily adapt to changing business application needs providing scalable solutions irrespective of physical topology of network.
- **Automated:** Policies defined in software controller can be used to provision, control and manage any device within data center providing high degree of automation.
- **Resilient:** Software-led infrastructure is less prone and easily adaptable to failures providing redundancy, fault tolerance and easy recovery in no time.

Thus SDDC is one-stop solution to deliver on-demand services to support almost any kind of application and heterogeneous data meeting increasing demands from data center infrastructure. Organization looking forward to implement SDDC can use one of the two approaches. One is the turnkey approach where various infrastructure problems are converged to one single integrated solution. This is suitable for organization who wants to start with small integration and eventually scale to a larger solution. Other approach is DIY where each use case is handled with a dedicated piece of software control plus underlying hardware. This approach is

technically more challenging than turnkey approach and is less preferable to organization starting a move to SDDC [41].

Software-Defined Data Center (SDDC), also termed as Software-Led Infrastructure is a new wave or futuristic operational and application changes in data center architecture that takes all the benefits and challenges of virtualization to all aspects of a data center viz. network, compute and storage. It deals with virtualization of every hardware device being used in a data center and operating and managing it with software-based control. Key features that SDDC promises to bring to networking world are summarized below.

- Virtualization is good but an organization had to still stick with their SLAs once defined. Addition or Deletion of resources had detrimental impact on SLA and QoS set by organizations. With SDDC implemented, turning up and down resources gets extremely efficient, highly automated and less painful with the total control of infrastructure being exerted from a centralized software residing on servers.
- Total exertion by software over deployment, configuration, provision and management of resources. A single, centralized software hub will manage networks of data centers. To add more granular power, the same software hub will have control over physical and hardware components and will be able to automate even power and cooling infrastructure.
- SDDC can be thought as a meet-up between elasticity of cloud computing and automation power of virtualization.
- It will be backward compatibility design with support for both static legacy applications and dynamic, real-time cloud applications.
- With centralized control, applications can easily share resources between data centers and just relying on their own data center.
- Research shows that SDDC will take the resource utilization of virtual data centers from 50% to 70%. SDDC will achieve this by holistic orchestration and providing agility towards changing business needs. SDDC takes the siloed infrastructure to peered-architecture where addition of new service can be thought of as single event rather than an event for each layer of the infrastructure.[7]
- SDDC could lead a future to IT-as-a-Service (ITaaS) by having centralized management control and provisioning within minutes.

SDDC is not about being hopeful but it promises to lead data centers into service-driven architecture. Cloud computing has revolutionized service-driven industry and bringing agility and resource-optimal solutions to data centers. However, what makes SDDC stand out as a leader in agile data centers is following distinguished elements.

- The most important characteristic of SDDC is adoption of SDN into data centers. By adopting to changing application and business needs and bringing SLAs to application

[7] http://www.avaya.com/usa/documents/the_software-defined_data_center_is_key_to_it-as-a-service.pdf

level, moving network management from silos to software control, prioritizing traffic and resource needs to enable high performance and a centralized control to monitor networks of data center through SDN is of prime importance to SDDC.

- SDDC would have all resources virtualized from compute, storage and network to physical and hardware components. A mix of these resources can be allocated and de-allocated and services can be orchestrated through coded software control to meet ever changing and real-time needs of various applications.
- Having well-designed and developed APIs and protocols will allow holistic and synchronized management of applications, devices and data. This can be achieved by having a unified view of network metadata that could be controlled with the APIs and protocols coded in centralized software control.
- SDDC promises to have even storage components software-driven. With the use of flash storage, high persistent storage components will be deployed across data centers ensuring integrity, quick disaster recovery, immediate availability and optimized access.

SDDC as a data center design embarks on a journey taking the best out of all data center architectures and optimize the time-to-market for service, agile business needs, cost reduction and better productivity with like-never-before speed and performance.

Figure 32 shows logical view of how a futuristic SDDC infrastructure could be implemented keeping the mainstream adoption problem in mind.

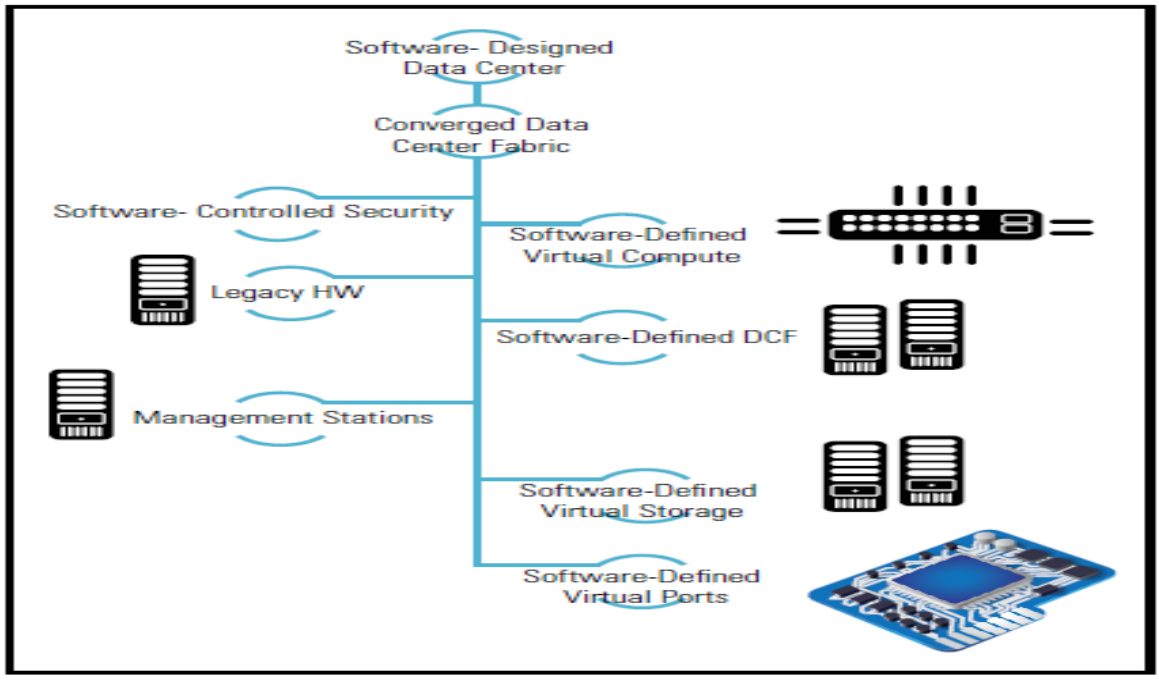


Figure 28 Logical View of SDDC Model.
Source: White Paper: The Journey toward the Software-Defined Data Center | Cognizant 20-20 Insights | September 2013.

The above logical view indicates the granular control SDDC needs to incorporate in every sector of Data Center from primary components like servers, storage and network to power management and cooling infrastructure. The vision SDDC carries is to unify network fabric for cooling, storage and IP data. Software-defined Power and cooling also plays important role in SDDC design as it has to keep in pace with ever growing provisioning of VMs across the infrastructure. For management of such infrastructure being controlled by centralized software controller, monitoring and reporting tools should be vendor-neutral, integrated, comprehensive and cohesive [42].

The primary stack blocks of a holistic SDDC solution could comprise of those shown below in Figure 29.

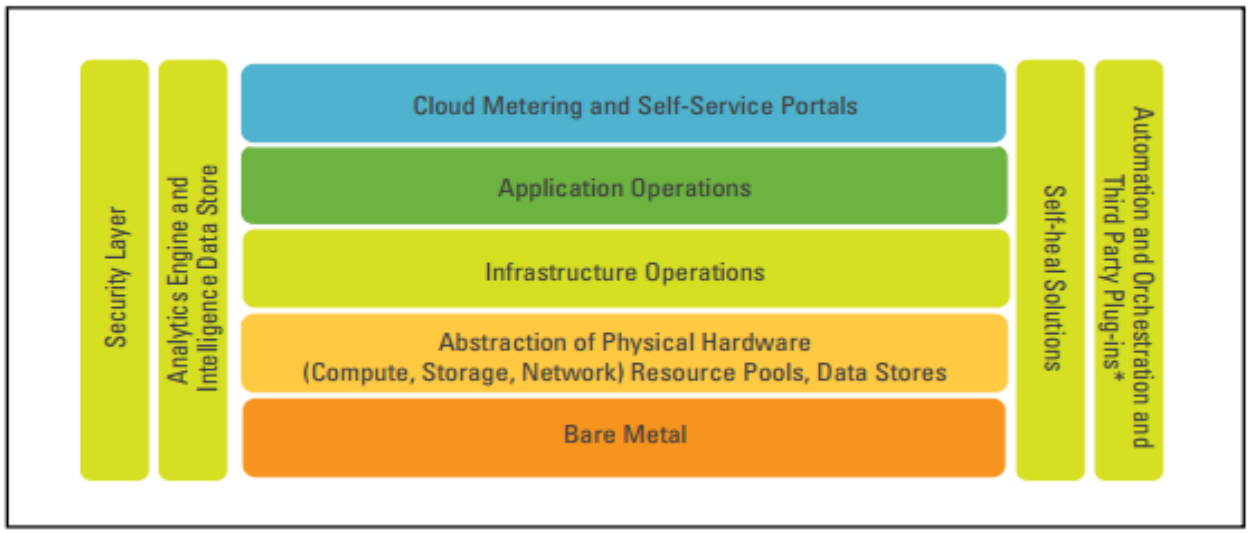


Figure 29 SDDC Stack Block Diagram.
Source: White Paper: The Journey Toward the Software-Defined Data Center | Cognizant 20-20 Insights | September 2013.

The overview of each stack block is discussed below [42]:

- **Bare Metal:** Physical hardware inclusive of storage, compute and server.
- **Abstraction Layer:** This the most important layer where all Data Center equipment are virtualized and presents individual logical containers of each device.
- **Infrastructure/Application operations:** Usually management and monitoring tools exposed through APIs for tuning the infrastructure.
- **Cloud Metering/Service Portals:** SDDC is designed by keeping in mind the response time from infrastructure in case of end user actions. This block ensures timely updates and event notification in such scenarios.
- **Security Layer:** Architecture level security control to ensure Confidentiality, Integrity and availability of data being consumed or accessed.
- **Self-Heal Solutions:** SDDC automation should be able to identify, diagnose and resolve the issues.

- **Orchestration and third-party plug-ins:** Data flow and resource access from third party tools needs to be controlled and monitored. This layer will expose APIs for that.
- **Analytics engine and Data Store:** A database that stores reporting and analytics data such as audit and logs and a sophisticated engine to provide such information to systems requesting logs and reports.

2.1.4 Network Functions Virtualization (NFV)

While SDN tries to virtualize almost everything in Data Centers, the need of having a new hardware appliance for each network service was still to be addressed. In 2012, telecom companies joined hands and introduced a NFV call to action document. Soon after that, the increasing market demand of NFV triggered setup of new committee called European Telecommunications Standards NFV. NFV uses virtualization techniques to consolidate networking services like content delivery, firewalls, load balancers, NATs, mobile base station controllers, etc. from its corresponding hardware appliance to commodity servers.

The primary benefits of NFV are [43]:

- Reduce CapEx by moving purpose-built hardware appliance functions to commodity servers.
- Reduce OpEX by reducing space, power and cooling requirements.
- Agile Network Operations by shrinking or growing network services based on changing demands in network.
- A single network appliance can be used to promote multiple network services.
- Designing ecosystem by using software-only network services.
- Easily deliver new services by evolving software written for that services.

The following figure shows the ETSI vision in delivering network functions through cloud.

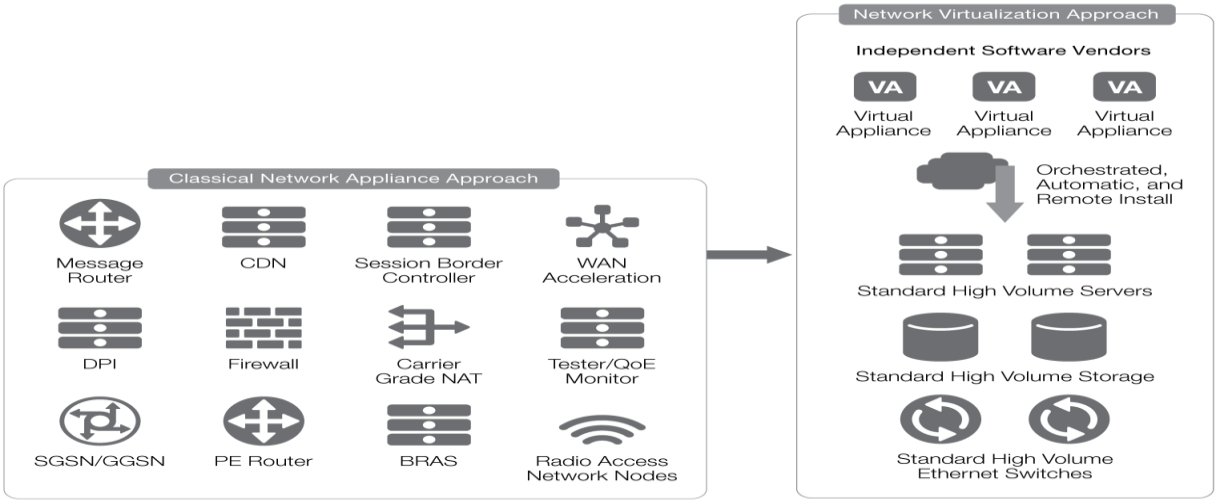


Figure 30 ETSI Vision in delivery NFV.
Source: https://f5.com/Portals/1/Images/whitepaper-images/NFV%20Everything%20Old%20New%20Again%20Service%20Providers/CS01-2320_WP_NFV-2.png

NFV looks promising but still needs to address few challenges before getting a widespread acceptance.

- Developing such wide range of virtual network appliances and orchestrating them needs a special care to ensure security from misconfiguration and attack.
- NFV makes sense only when each network appliance can be automated from a single control.
- Virtual appliance should be resilient to hardware failures on which it is deployed.
- Assuring no vendor lock-in and operability of NFV across multiple vendor hardware.

Discussion of NFV is however important in context of SDDC because it highly relates to the fundamentals of virtualization and SDN. SDN deployments may take longer so providers want to take leverage of providing quick NFV solutions meanwhile. SDN and NFV both rely on separation of services from underlying hardware and designing software solutions which are vendor-neutral and can be executed on commodity hardware. Both promotes high degree of automation and orchestration to deliver virtual services [44].

2.1.5 Software Defined Security or Protection (SDSec/SDP) – a NFV Example.

One of the most important challenge SDN adoption is facing right now is securing the access and integrity of SDN controller itself. It is highly important in SDDC to have the controller orchestrating the whole infrastructure to be tightly controlled in terms of access and also to have a backup of exact copy of controller ready in case of failure. Software Defined Security is partitioned into three layers viz. **Enforcement Layer, Control Layer and Management Layer.** Enforcement Layer is responsible for inspecting the moving traffic for possible threats. It creates network segments and possible physical and virtual enforcement points. Segmentation of network is important as different compartments of a physical network infrastructure imposed different level of security threat. Enforcement points can be gateways, VMs or software applications.

The control layer maintains security policies and deploy them at all the enforcement points. This layer is the heart of the SDSec architecture as it develops the policies based on knowledge about organizational data access rights, data assets and classification and information about possible threat to organization. SDSec can be considered a NFV solution because of variety of security solutions implemented in control layer which may include firewall, antivirus, anti-Bot, spam and email security and Intrusion prevention system.

Management layer orchestrates infrastructure and collaborates security norms with business processes. This layer provides overview of network operations and helps in assigning administrators to handle security operations within different compartments of an organization. Security incident reporting and monitoring is part of this layer.

A conceptual SDP solution would look as shown in **Figure 31**.

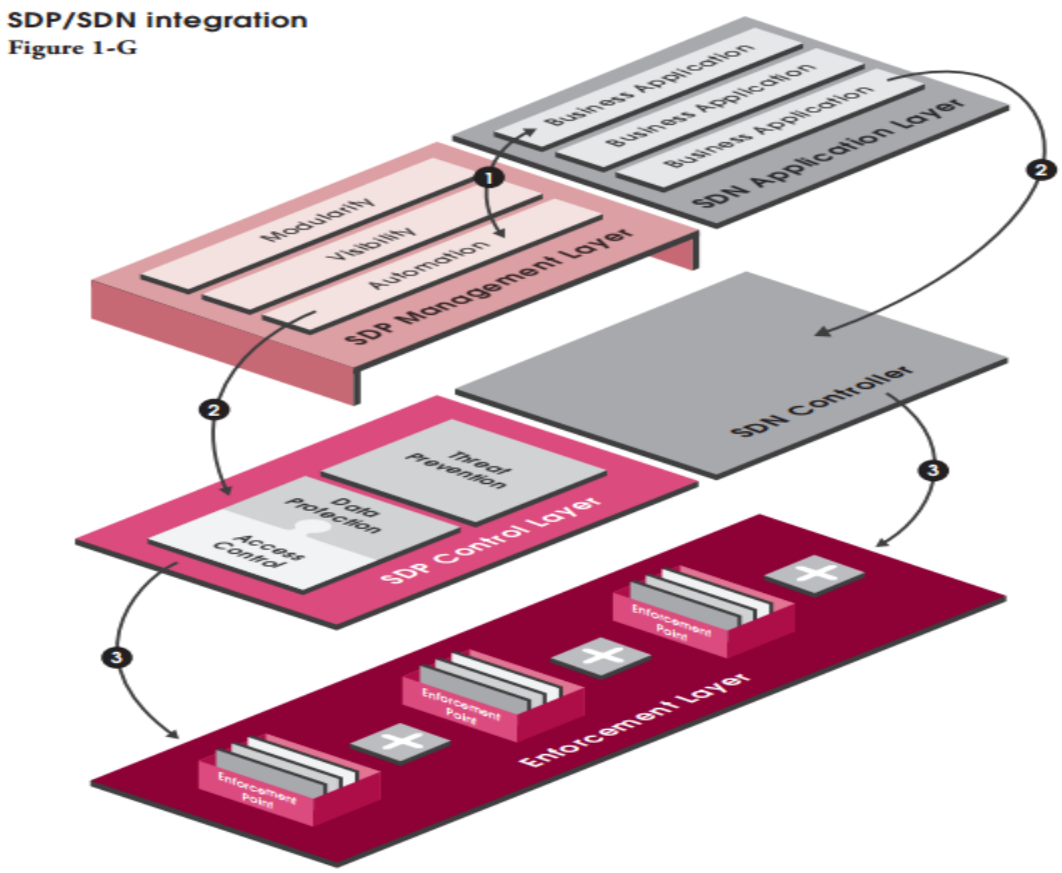


Figure 31 SDP/SDN Integration.
Source: White Paper: Software-Defined Protection, Check Point Software Technologies Ltd.

2.1.6 SDDC as a solution for issues in Multi-Tenant, Virtualized Data Center

SDDC is still evolving into a major offering to computing world. However being proactive and taking baby steps with holistic design of data center can help organizations incorporate SDDC on large scale and with full support. Organizations has to strategically apply virtualization to different data center resources one step at a time. Proper training methods is a must for everyone from junior administrators to senior architects to understand the fundamentals of SDDC is a must. SDDC redefines everything in today's traditional and virtualized data center and thus proper implementation of policies and support for OEM products from other vendors is a must. Organizations also needs to design a strict incremental plan designed by industry mentors viz. server, storage and network architects. A formal adherence to security policies and standards and regular audit for SDDC integration is highly encouraged.

A well-designed plan when executed can help leverage following benefits from SDDC:

- Improved resource utilization over all components of a data center.
- Single, centralized view of entire network of data center. This will require less human resource that was needed in siloed, application-centric data centers.
- Agility and easy adoption to ever changing business needs.

- Taking design philosophy to IT-as-a-Service model and reducing time-to-market by considerable factor.
- Less expenses on maintenance and addition of physical devices into infrastructure by having optimal resource utilization. Research shows the CapEx and OpEx to have been dropped by 50%.[8]
- Get the best of the commodity hardware from competitive vendors as all the networking functions will be controlled by software running on servers.
- With sophisticated management and monitoring cloud-centric tools of SDDC, human resource can spend all their valuable time in driving computing innovations instead of fixing repetitive data center issues.

SDDC is more reliable, resilient and efficient than all the previous data center architectures – thanks to the provisioning and management of resources through programmed and coded software layer.

Legacy Data Center	Virtual Data Center	SDDC
Clearly defined IT silos	Forced integration of silos	No IT silos
Poor infrastructure utilization (30%)	Improved infrastructure utilization (50%)	Optimized infrastructure utilization (70%+)
No virtualization	Virtualization enables consolidation	Virtualization enables automation
People are 40% of data center TCO	People are 30% of data center TCO	People are 20% of data center TCO
No orchestration	Server/network orchestration	Full infrastructure orchestration
Provisioning time is months	Provisioning time is days	Provisioning time is hours/minutes

Source: ZK Research, 2013

Figure 32 aggregates the enterprise-wide benefits that could be achieved with a well-organized migration to SDDC.

[8] http://www.vmware.com/files/pdf/accelerate/VMW_13Q1_BB_SDDC_020813_FINAL_LTR.pdf

Feature	Description	Benefits	Capex Savings	OpEx Savings
Centralized Management and Configuration	Configuration of server resources, networking resources, and storage resources is done in the SDDC platform software	Simplified administration Automated configuration changes in support of dynamic operations, and cloud automation	None	Existing admin staff can support more servers Fewer expensive network and storage admin's needed
Server Consolidation	80% to 90% of the physical servers are eliminated	Fewer servers to purchase at the next refresh cycle Fewer new servers to power and cool	Only have to purchase 10% to 20% as many servers as in the past	Only need to manage 10% to 20% as many physical servers as in the past
Network Consolidation	Fewer network ports are required in servers and in network switches	Fewer top of the rack switches are needed	Fewer switches need to be purchased	Fewer switches need to be managed
Per Switch Cost Reductions	Some switching moves into the SDDC software, allowing for less expensive hardware switches to be used.	Cost savings from replacing expensive switches with commodity switches	Replacing expensive switches with inexpensive switches	Fewer network administrators needed
Support for Hybrid Cloud Computing	Software Defined Networks can be stretched across data centers	Workloads can be moved to their most cost effective location of execution.	Servers, networking and storage no longer need to be purchased for cloud resident workloads	The cloud vendor pays to administer the cloud based servers, networking and storage
Storage Optimization	Less expensive direct attached storage replaces expensive network and fiber attached storage	Cost savings from replacing expensive storage with less expensive storage	Less expensive storage needs to be purchased	Fewer expensive storage administrators are needed
Software Based Services	Services like load balancing and firewalls are implemented in software instead of hardware	Software based service are less expensive and easier to manage	Fewer dedicated hardware appliances need to be purchased	Software based services are easier and less expensive to manage
Improved Business Agility	The entire data center can be quickly configured to meet business needs	The data center will become more responsive to the needs of the business	Less specialized hardware will be needed to accomplish business objectives	Fewer IT Operations staff will be able to provide more agile IT services
Management Tool Consolidation	Fewer different management tools will be needed	Savings from the purchase of management tools and the staffing required for them	Less money spent to purchase many different management tools	Less money spent to administer and use many different management tools

Figure 32 Benefits of SDDC.
Source: White Paper: Business Benefits of Software Defined Data Center, Bernd Harzog, The Virtualization Practice, August 2013.

2.1.7 SDN Use Cases

SDN can be viable for data center and network wide operations that requires scalability, automation, agility and on-demand changes. An intensive research and survey was recently

conducted by Webtorials Analyst Division to figure out how SDN and NFV is helping organizations and their viewpoint on different aspects and trends on SDN. In one survey, recipients were asked about challenges and opportunities that SDN brings to the networking world and the following table summarizes the response.

Table 5: Opportunities & Challenges that SDN Can Address

Challenge or Opportunity	Percentage
Better utilize network resources	55%
Perform traffic engineering with an end-to-end view of the network	54%
Ease the administrative burden of configuration and provisioning	53%
Support the dynamic movement, replication and allocation of virtual resources	52%
More easily scale network functionality	45%
Enable applications to dynamically request services from the network	45%
Have network functionality evolve more rapidly based on a software development lifecycle	41%
Reduce OPEX	40%
Implement more effective security functionality	35%
More easily implement QoS	33%
Reduce CAPEX	29%
Reduce complexity	24%
Other	5%

Figure 33 Webtorials survey: Opportunities & Challenges that SDN Can address.
Source: http://www.webtorials.com/main/resource/papers/webtorials/2015-Guide-to-SDN-and-NFV/2015_Guide_Chapter_1.pdf

In another survey by same company, recipients were asked as to where they would like to implement SDN and following response was aggregated.

Table 7: Focus of SDN Deployment

Focus of SDN Deployment	Percentage
Data Center	64%
WAN	26%
Branch and/or Campus	25%
We are unlikely to implement SDN within the next two years	12%
Don't know/NA	10%
We are likely to implement a service from a WAN service provider that is based on SDN	8%
Other	6%

Figure 34 Webtorials Survey: Focus of SDN Deployment.
Source: http://www.webtorials.com/main/resource/papers/webtorials/2015-Guide-to-SDN-and-NFV/2015_Guide_Chapter_1.pdf

Let us try accumulate the probable use cases of SDN based on trends organization has responded with in above research and survey. It is clear that organization adores SDN as a solution to address wide range of challenges and that Data Center is likely to get wide spread SDN acceptance. So let us divide the SDN use cases in three different categories of deployment viz. Data Center, WAN and Campus Networks.

2.1.7.1 SDN Use Cases: Data Center

VM Migration within data center between physical servers can be costly and complex operation. Administrators have to make sure that VM remains in the same VLAN post migration. SDDC requires workload management and having a graceful VM migration is crucial. With the help of NV, when a VM is moved to a new subnet, the edge switch of overlay network will update the mapping table automatically reflecting VM's new physical location. SDN suggests using technologies like VXLAN for tunneling the VMs between location [45].

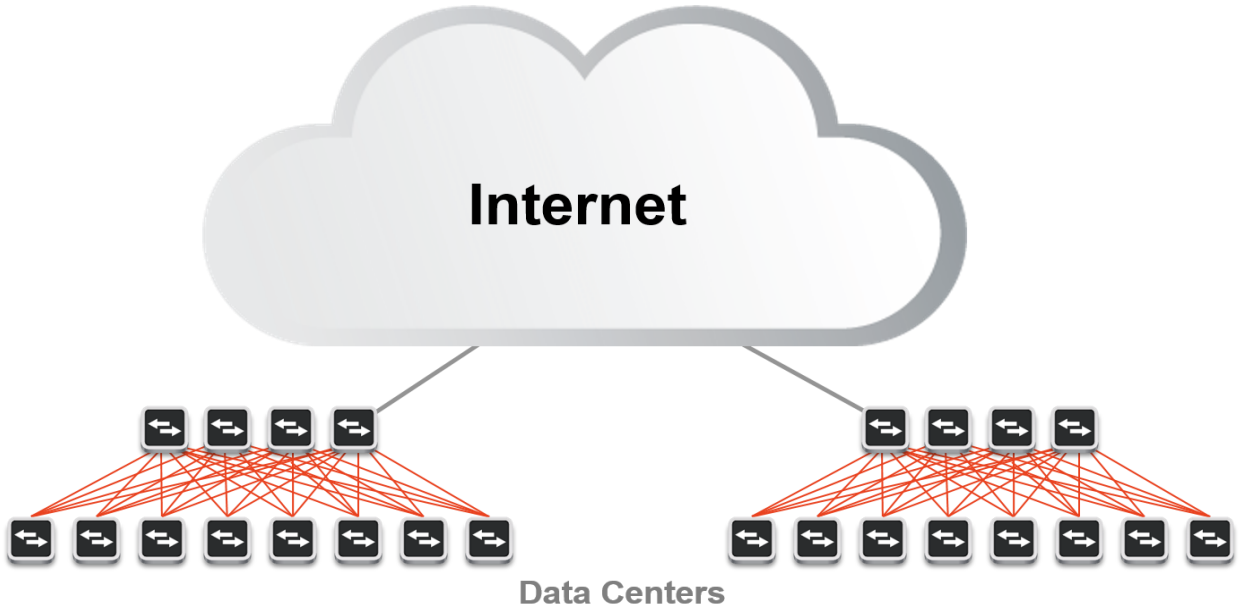


Figure 35 VM migration between physical locations.
Source: https://www.sdxcentral.com/wp-content/uploads/2015/02/The-data-center-is-one-of-the-most-vocal-SDN-battlegrounds.png

Another excellent use case of SDN is Network Function Virtualization to provide **Network Service Insertion and Chaining** for various L4-L7 services. For e.g. suppose a data center needs a new firewall service. With traditional data center, we would have to rent or purchase a new network service appliance exposing firewall services and cable it to existing network. This technique is waste of physical space, human resource and time required for adding a new service. With SDN and NFV, we could have a commodity server implementing the network service and then direct the traffic flow through the server or chain of servers (if necessary).

SDN is also extremely useful in providing **Security Services.** OpenFlow switches allows modification of packet headers thereby allowing the switch to redirect suspicious traffic to security devices and match them to pre-defined list of malicious signatures.

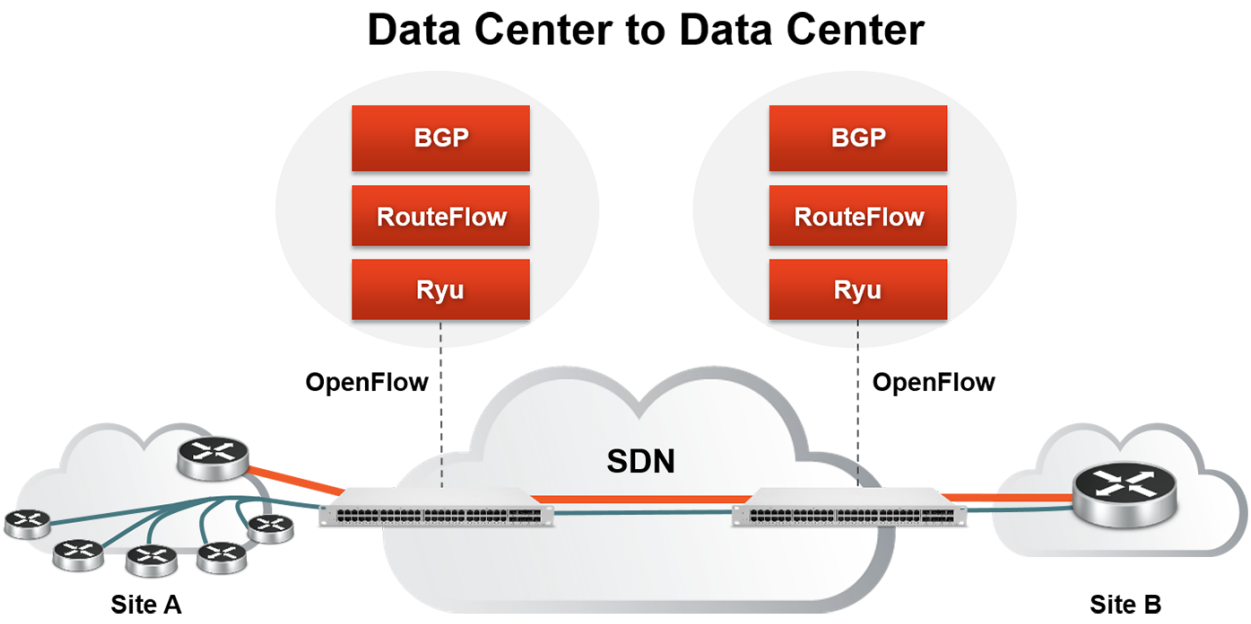
It is also possible to design a **SDN-enabled cloud infrastructure.** Such infrastructure would not only allow the goodness of compute and storage virtualization but also add the features of network virtualization to it. Adding UI-based network control allows easy access for customers and allows provider to handle more customers and VMs efficiently. It will allow data center wide programmability which gives full workload portability and consistent network services. VM-level security over entire data center allows secure channeling of customer needs.

2.1.7.2 SDN Use Cases: WAN

OpenFlow is a powerful tool for having a centralized control over the forwarding plane of entire network using its Forwarding Information Base (FIB) and allows calculating the most optimum route for each flow. This allows each flow to be serviced based on the traffic needs it requires. Changing traffic patterns can be accustomed with bandwidth on demand. It also helps in designing network that knows the possible failure routes proactively allowing excellent resource utilization.

Google's G-Scale is a production example of OpenFlow Layer 3 Network. G-Scale control plane comprises of 128 port (each of 10 GB data rate) switching fabric implemented fully only with OpenFlow [46]. This implementation took Google's Network Utilization to around 95%.

Google implemented traffic flow between data centers supported by different providers using OpenFlow metering by splitting traffic based on SLA performance or cost and then push or pop MPLS headers to drive traffic either way.



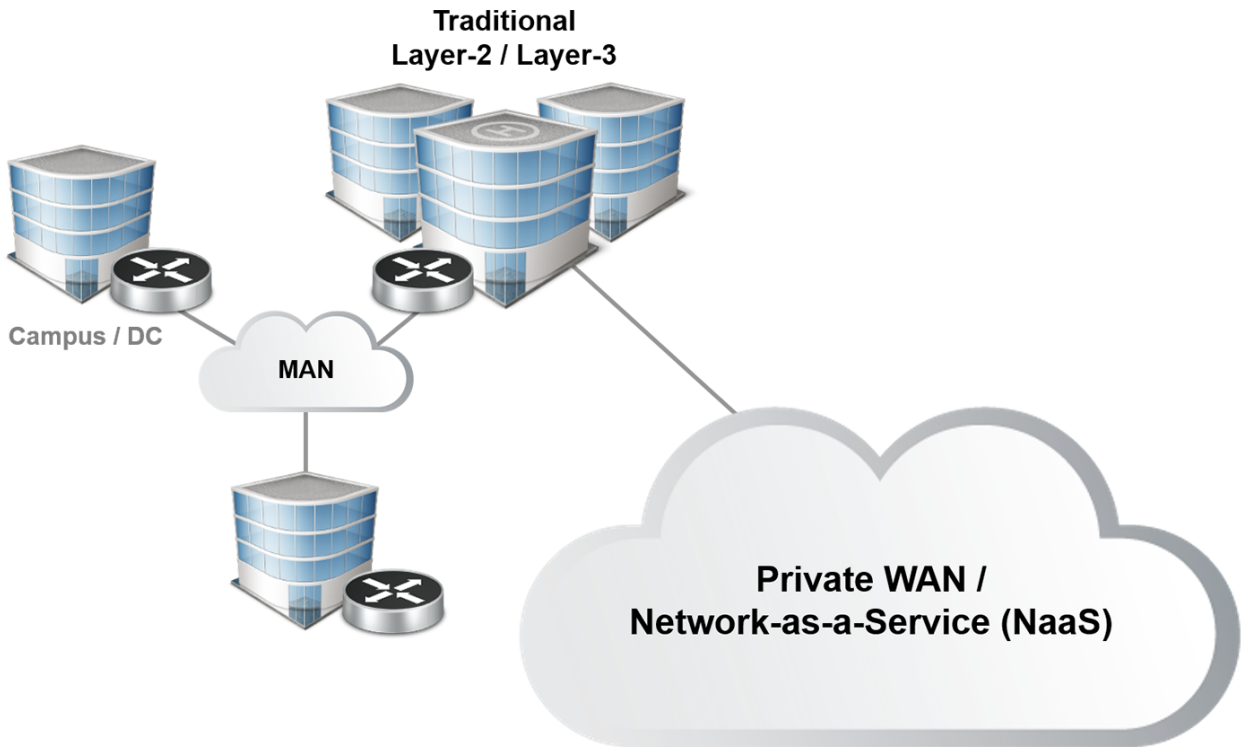
DRIVERS	TECHNOLOGY
• Private to public cloud transition • Business logic intelligence • Application based SLA assurance • Intelligent geographic-based load balancing	• Tunneling / Overlays • OpenFlow • Open source • Bare metal – hardware and software separation

Figure 36 SDN implementation between Data Centers.
Source: https://www.sdxcentral.com/wp-content/uploads/2015/02/Picture3b2.png

2.1.7.3 SDN Use Cases: Campus Networks

Unified Communication tools are quite used on campus networks. But it is not possible to prioritize UC traffic over other communications within the network. OpenFlow allows to **dynamically set QoS** parameters and apply **Traffic Engineering** by sending information to SDN controller to prioritize traffic between pair of IP addresses. SDN controller sets the priority and informs all the OpenFlow-enabled switches about the changes so that they can find the most optimum path for flow to traverse through.

SDN also allows **unification of both Wired and Wireless networks** so that there is only single point of management of network of two different types. This makes network provisioning dynamic by changing the network policies automatically for a user device when it moves between different Access Points. It also possible with SDN to bypass firewall for trusted traffic to restrict unnecessary usage of network services like Firewalls.



DRIVERS	TECHNOLOGY
• Business logic intelligence in the network • Private to public cloud transition • Centralized decision making • Application based SLA assurance • BYOD drives more dynamic security needs	• OpenFlow • Layer-2 / Layer-3 demarcation • Bare metal movement, hardware and software separation

Figure 37 SDN implementation in Campus.
Source: https://www.sdxcentral.com/wp-content/uploads/2015/02/SDN-applications-in-the-enterprise.png

SDN also allows setting up campus network with **Role Based Access**. OpenFlow takes the advantage of authentication process involved between a user and Network Access Control. When authentication is done, a SDN application dedicated for role based access is notified of user's MAC address, port where it enters the network and the role of user. Based on this information, OpenFlow can maintain a list of users and their associated devices, other users they talk to and the bandwidth they can leverage based on role. This list is transferred as a message to SDN Controller which then notifies all the devices to allow network access based on roles.

One more use case that is generic of the network type is **Network Taps** [47]. A usual physical tap in a traditional network would require one switch per tap and a dedicated production port for each tap. This tap then functions for specific networking needs like filtering, mirroring, sniffing, aggregation, etc.

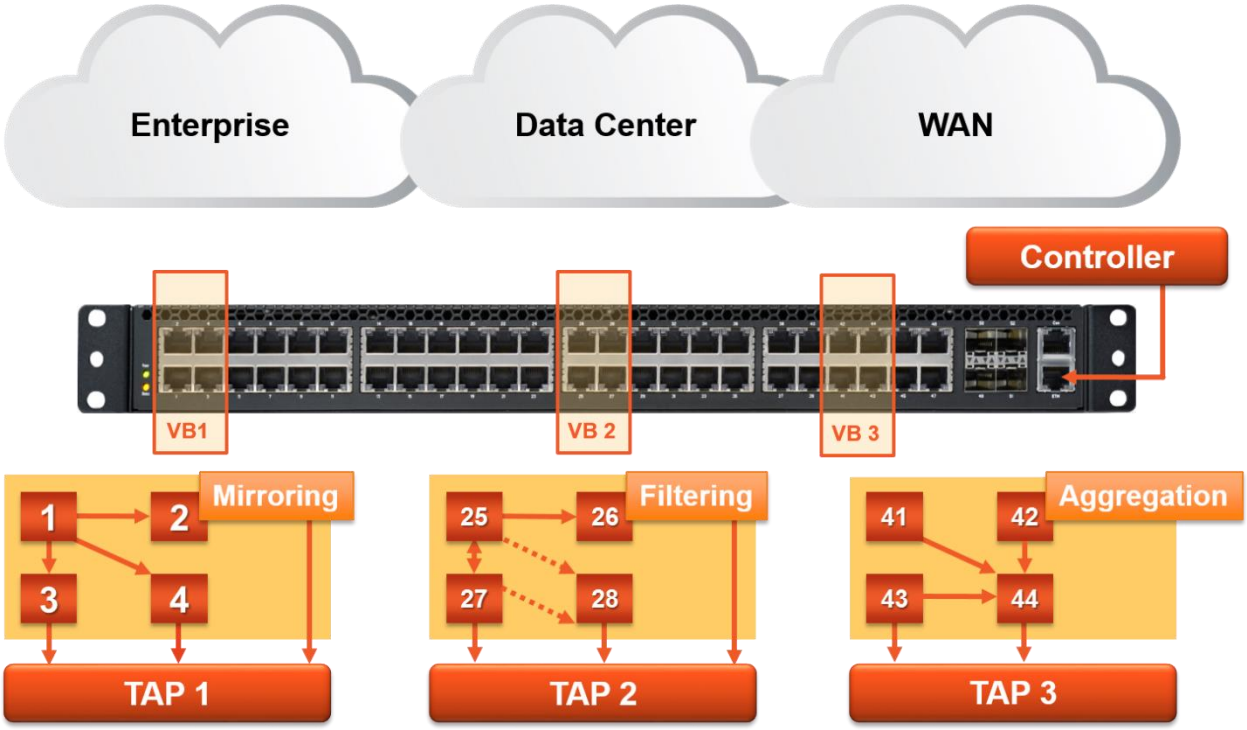


Figure 38 SDN Taps
Source: https://www.sdxcentral.com/wp-content/uploads/2015/02/IT-has-leveraged-network-taps-for-decades.-SDN-provides-a-more-programmable-and-therefore-more-cost-effective-approach..png

By using SDN, all we need is only one switch with a port dedicated for each tap eliminating the need of excessive hardware. You could use mirroring to separate voice, data and video traffic and send it to multiple ports from single port or you could filter traffic to send it to two different ports for analysis or you could check the SLA and link performance by aggregating different traffic over a single link.

2.1.8 Few SDDC Solutions by Vendors

2.1.8.1 Solution by EMC Corporation and VMware

In mid of 2013, EMC[2] and VMware joined hands to form the EMC federation along with RSA, VCE and Pivotal Inc. It aims to provide solutions in software-defined enterprise, cloud, BigData and social platforms. The Software-Defined Data Center solution is a joint venture of EMC Corporation and VMware.[9] The 'Federation Software Defined Data Center' solution takes the server virtualization wonders to data centers to enable enterprise to deliver applications and services with agility, speed, security and efficiency.

EMC Federation calls it a fully automated and virtualized data center design with exertion control completely leveraged from their software platform.

Storage management is automated and centralized with adaptability towards various storage types like file, object, HDFS and top storage vendor's products.

[9] http://www.emc.com/collateral/solution-overview/h13541-sddc-so.pdf

With network virtualization, they have tried to address what VMware calls as 'microsegmentation' – which is developing a network hypervisor equivalent to keep virtual networks isolated from each other and the underlying physical network, preventing threats from moving into the data center. These hypervisor equivalent also redefines the layer 2 – layer 7 network services from routing and switching to firewall and load balancing.

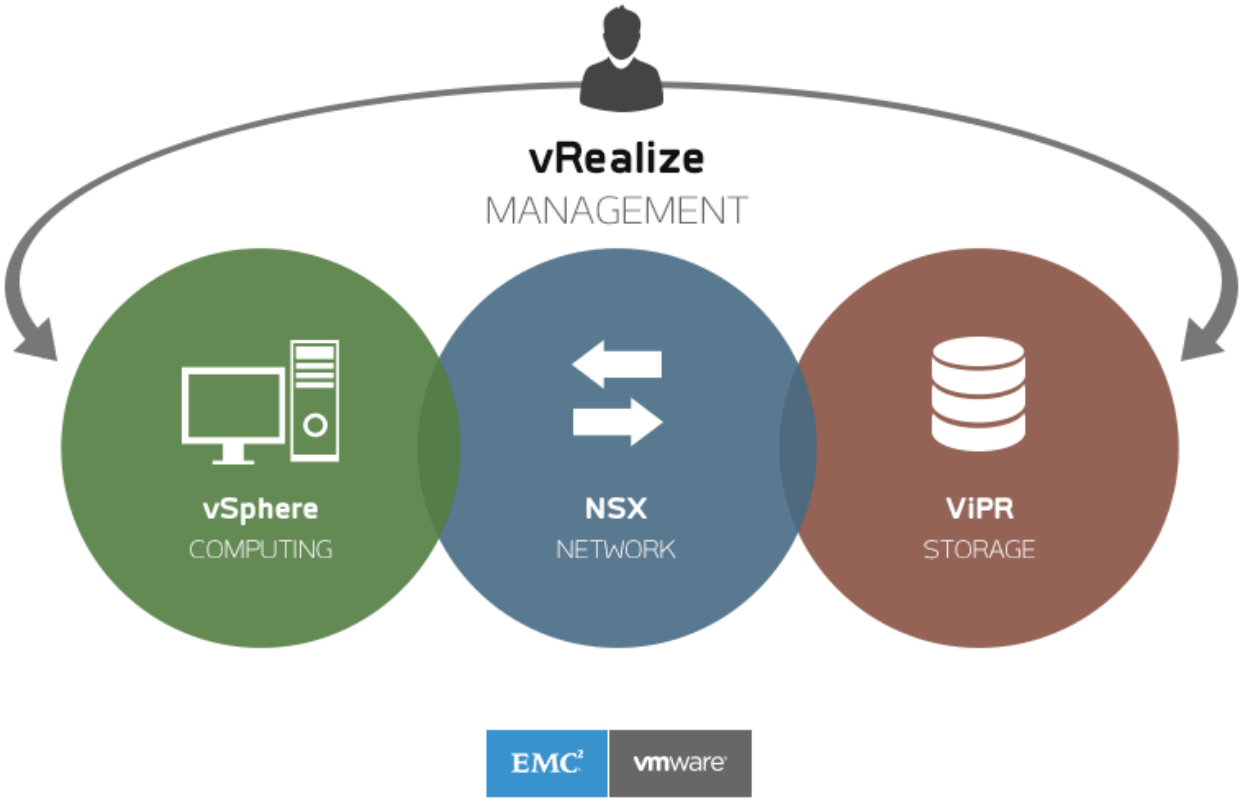


Image Source: http://emcfederation.com/solutions/software-defined-data-center.htm

The solution proposed by EMC Federation for Software Defined Data Center has following components for various data center pillars.

- VMware vSphere® is a primary compute component for virtualized SDDC infrastructure.
- VMware NSX™ takes the lead as network virtualization and security platform in EMC Federation's SDDC implementation.
- EMC® ViPR® is storage platform from EMC Corporation to meet real-time, dynamic cloud workloads.
- They bring in a very sophisticated management suite and tools for managing the SDDC. Both VMware vRealize™ Suite and EMC Storage Resource Manager combines to deliver analytics, resource optimization, configuration analysis and visualizations of storage relationships on dashboards.

Some of the primary features as stated on EMC Federation portal are[10]:

- Automated VM, Storage and Dynamic Network Provisioning.
- Automated Monitoring.
- Automated Disaster Recovery.
- Hybrid Cloud workload migration and connectivity to public cloud.
- Backup and Recovery Services and Micro Segmentation.

This use case clearly depicts how SDDC is not 'vendor-specific' and how different vendors with different focus areas can come together to provide a robust solution to large networking problems.

2.1.8.2 Solution by Lenovo and VMware.

In late of 2014, Lenovo collaborated and partnered with VMware to deliver holistic IT infrastructure inclusive of benefits of SDDC[11]. The first embedded VMware hypervisor was result of their long term collaboration. This partnership will bring Lenovo's extended line up of servers like System X and networking hardware with VMware's infrastructure to deliver a promising SDDC infrastructure with high speed provisioning and significant cost reduction in establishing sophisticated data centers.

Some key features of this partnership in the world of SDDC includes:

- Lenovo being the dedicated networking partner for VMware NSX™, will bring in new integrated traffic management and virtualization gateways.
- Collaboration of VMware Virtual SAN™ and Lenovo servers to bring resilient, elastic and scalable Software-Defined Storage solutions for all kinds of budgets, storage and performance requirements.
- Lenovo eXFlash technology couples with VMware Horizon® VDI platform to provide enhanced end-user virtual desktop services with extremely low latency.
- Rapid provisioning of services and resource management with Lenovo servers and VMware vRealize Suite™.
- VMware vSphere with Operations Management on System X for optimizing virtualization.
- VMware vCloud Suite on X6 Systems to build private clouds for mission critical applications and scale it SDDC.

This is one more use case of great collaborative efforts to bring conceptual SDDC design into real networking world.

2.1.9 Challenges with SDN and SDDC and Next Steps to Migration.

SDN and SDDC will definitely shape the networking of tomorrow with the versatile features it brings to the Data Center Design and Development. However, it brings some risks and

[10] http://emcfederation.com/solutions/software-defined-data-center.htm
[11] http://news.lenovo.com/article_display.cfm?article_id=1850

challenges with it which needs to be addressed before completing migrating the DC infrastructure to SDDC solution. Some key challenges that SDDC brings would be:

- When one single software control manages the entire network, the reliability and proper configuration of software control is biggest challenge to be addressed.
- SDDC tries to abstract workload from every possible physical device. Thus it becomes really crucial and difficult to monitor if the abstracted workload is actually entitled with the resources it needs. Thus SDDC requires constant monitor of virtualized workloads – more frequent than it would be needed with just physical devices in infrastructure.
- With everything virtualized, contention of a physical resource will happen more frequently. Thus the contention resolution logic that was once implemented in physical resource has to be incorporated into software control now.
- A minor configuration error can turn into a snowball effect affecting excessive workloads blocking complete data center operations. Thus management, reporting and monitor tools developed for SDDC has to be highly sophisticated.
- A single software platform is expected to assign, monitor and configure all the data center resources including network, storage and compute. This software platform with such wide variety of applications has to be sophisticated and properly designed and tested.
- Resource utilization will be better than ever with SDDC but to make sure that each workload gets its expected resources is not an easy task. This will be a very important part of SDDC software platform design and development.
- Frequent auditing and change logs of resources should be triggered by software platform. This is needed because two workloads can always demand same urgency and size of same resource. The software platform should be able to prioritize this based on application supremacy levels. Also, the storage, network and compute devices should be versatile to handle such high demands.
- Workloads can be easily get affected and form a snowball effect from minor errors in software platform designed for SDDC. This requires the software platform to be unit and integration tested considering all possible scenarios.
- SDDC software stack must be able to transit between SDDC infrastructure and standard legacy infrastructure on demand.
- There is a definite need to create open standards and frameworks to leverage SDDC to its full potential.
- The software stack itself has to be highly secured considering the fact that it is the only entry to control the data center.

Apart from this possible challenges in migrating to SDDC, organization needs to take baby steps to assure guaranteed, fault-free migration. Some key points to be remembered before and while performing migration would be [48]:

- Developing a roadmap to promote effective strategy in migrating from traditional/virtualized data center to SDDC.
- Instead of being greedy to incorporate SDDC into existing network, organization should add value and benefits to infrastructure in incremental fashion.
- Consolidating on hardware from vendors that promotes SDDC infrastructure is a must so that cheap and effective commodity hardware can be used to support variety of network services.
- Automate as many processes as possible to use IT staff in more valuable network operations and also provide business and IT agility.
- Arrange formal training for staff to understand the complex structure of SDDC.
- Regular assessments of incrementally evolving SDDC architecture will help organization smooth transition and protect from any snowball effect.
- Standardized security and data policy organization-wide.

Following summarizes how an organization can perform a smooth transition from their current data center architecture to SDDC infrastructure.

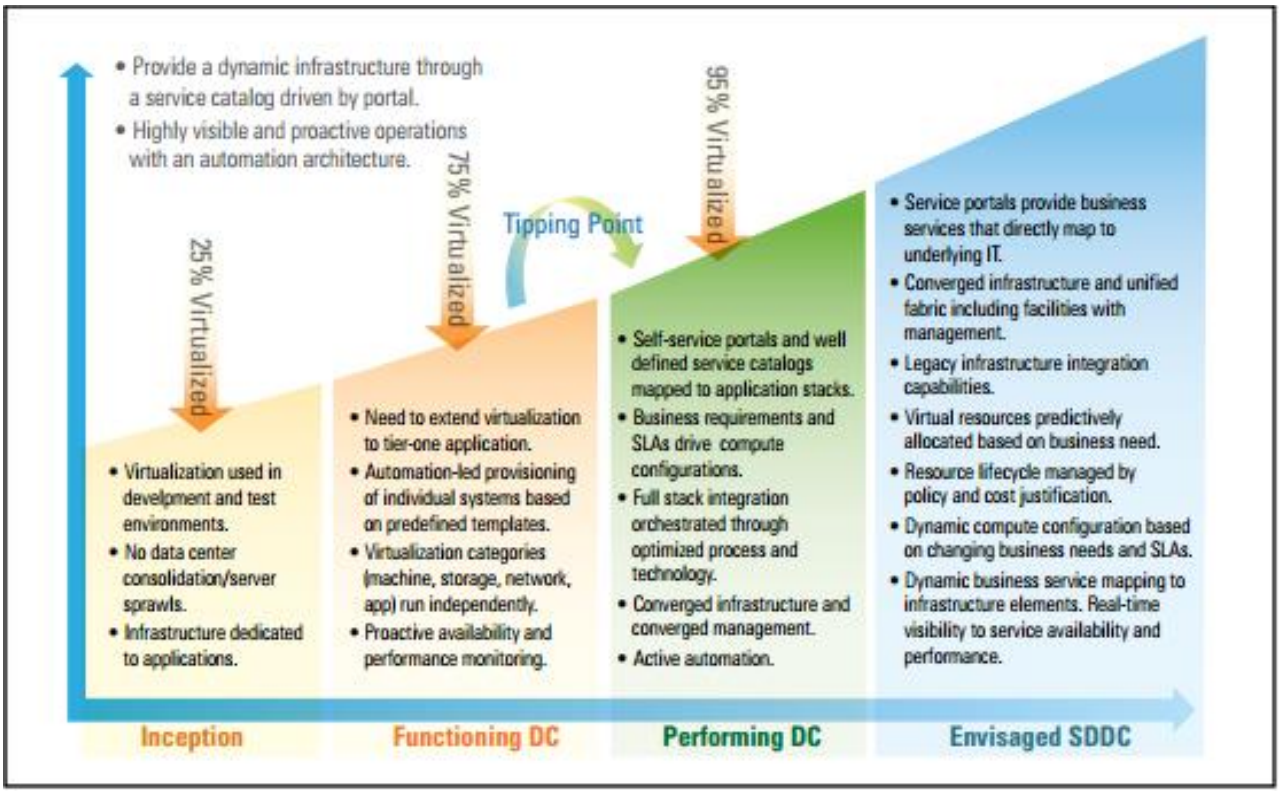


Figure 39 SDDC Adoption Roadmap
Source: Source: White Paper: The Journey towards the Software-Defined Data Center | Cognizant 20-20 Insights | September 2013.

3 RESEARCH TRENDS IN SDN COMMUNITY AND DATA CENTER ARCHITECTURES.

SDN, NFV and SDDC is gaining widespread acceptance in networking industry from small proof-of-concept research projects to actual large data center use cases. However SDN still faces several open challenges from architectural, implementation and design perspective. Some of these challenges includes the degree of programmability of controllers, distribution level of centralized control, assuring consistency of state of controllers and standardization of common APIs.

Academic researchers and industry specialists have join hands to promote challenges SDN and Data Center community is facing. One such research involves creating an Open Source Network Operating System with scale-out, fault tolerance and high availability features providing a global view of the centralized network even though the OS is distributed across servers [49]. The latest prototype improvise on performance by using low latency data store and optimized data model. It allows caching of topology and event notifications.

SDN allows flow-based forwarding abstraction for data plane and a logically centralized control plane. However, the data plane flow forwarding operates only using L2-L4 header field inspection. Thus there is still a huge possibility of handling flow forwarding for top layer information within the data plane. Research work has been undertaken to propose SDN architecture that allows an all layer flow forwarding decision within the data plane limiting the need to access the controller for making forwarding decision. This architecture has low overhead, improvised performance and scale out design to take the most out of the SDN layer abstraction [50]. The architecture has data plane that is application-aware by intercepting the messages between open vSwitch and controller and intercepting of packets after flow table lookup. It also adds application-level actions against flow table entries to have most of the top layer packet handling within the data plane.

One of the biggest problem in incorporating SDN in enterprise data center is to have SDN-capable network infrastructure or at least investing in such infrastructure. Organizations cannot discard in place legacy infrastructure and thus it slows down SDN adaptation by a huge percentage. Researchers have come up with solution like ClosedFlow which allows having popular OpenFlow control over proprietary network devices [51]. Legacy switches won't allow to have layer-2 interaction with controller. The solution uses minimal version of OSPF based on the route-map configurations and access list matching patterns. The topology discovery is maintained by having each switch connecting to remote controller and log changes in node adjacency. Packet match and actions is triggered using ACLs, Route Map and Interface Configuration. In OpenFlow, when a packet does not match any criteria, it is just forwarded to controller for further forwarding decisions. In this solution, a special explicit deny entry is added at the end of ACL which is sent when a packet has no matching flow.

SDN and NFV comes with bulk of positive traits which when utilized together can do wonders. One such research application is to solve the middlebox service implementation with PaaS delivery. With today's PaaS delivery, it becomes really difficult to provide sophisticated network functions like intrusion detection and transcoding with PaaS service. The research application called *MiddlePipes* aims to provide network functions virtualization of such network functions or middlebox services by piping it between the application and the PaaS service. It allows Network Functions to move close to the actual data streams, allows low-level and API level access and allows chaining of multiple network functions together [52].

One major challenge that still remains with SDN implementation in Data Center is to lower the forwarding decision making time in data plane. This could have adverse effect when dealing with real time data streams. Forwarding rules as of now deals with tens of milliseconds of time. However, video streaming and real-time synchronous data will require forwarding time of microseconds. A SDN-enabled Proof of Concept shows how an uncompressed HD video that can be streamed with SDN-enabled switches switching between the video feeds at a particular point of time. It does this by "source-timed" flow changes based on some packet header changes [53].

A full-fledge implementation of SDN with single provider data center architecture is getting common. However, SDN has not yet been implementation in WAN by more than few large providers say – Google. In a WAN architecture, SDN can be fully leveraged if there is nested network virtualization where a low-level virtual provider provides resource access to high-level virtual provider. To make this work, providers need to filter out the resources they are ready to delegate to other providers and to slice down the network along these resources by taking ownership of network resources in WAN. One such solution has been proposed by researchers by thinking of SDN in WAN as resource-management tool allowing delegation of resources in multi-domain and multi-provider WAN. The framework proposed takes the concept of label matching in SDN as it is and applies matching rules to it. It divides the network substrate into multiple domains which are owned by different providers. Each domain resources are managed by Resource Managers who then delegates the resources to the domain controller or to virtual provider's RM. The RM of any virtual provider can further subdivide the delegation to allow nested virtualization [54].

Network Security in SDN infrastructure has been rigorously discussed by architects and industry experts. Implementing security appliances and network functions like firewall, intrusion detection and prevention systems in OpenFlow-based network is really challenging. Considering firewall implementation, the primary job of firewalling is to be as an intermediate protection between private and public domains. However, it is not practical to think of all private domain nodes to be trusted entities. Firewalls in SDN-enabled environment needs to address the dynamic flow changes and that real time threat detection and resolution is incorporated for flow changes that instills threats into the network. A research initiative called FlowGuard has been proposed by researchers to design a robust SDN firewall that provides accurate, flexible and in real time policy violation detection and resolution [55].

Modern switch are constructed using TCAMs for implementing forwarding lookup tables. However they are limited by the number of forwarding entries they can store as it relies on unstructured MAC addresses. One of the research initiative is to use these MAC addresses as structured labels for flow entry lookup as with ATM and MPLS technologies. With this method, SDN Controller can intercept the ARP packets and modify it instead of adding new labels to the packets. This allows host to store the received MAC address in its ARP cache and put destination MAC address into all outgoing packets towards a particular IP address. This eliminates storing of entries into FIB tables thus reducing the usage of limited FIB entries. This method will have receiving host ARP table consisting of source host's labelled MAC address instead of actual MAC address implementing packet forwarding without hampering or violating Ethernet Standards [56].

SDN Controller was initially designed to be centralized in behavior monitoring the entire network state from single point of reference. However this controller cannot cope up with large networks and varying and increasing traffic loads. Research work has been undertaken to physically distribute these controller as instances on servers yet having one logically centralized controller to manage the workload. However, these architectures have extremely high flow setup latency and inefficient resource allocation resulting in high operating costs for controller. One such variant of controller architecture called *Pratyaastha* (elasticity) has been proposed that overcomes these challenges with some substantial results. Two important tasks in SDN architecture is division of network into application state and optimal assignment of application state and switches to controller instances. Designing SDN control plane in this way yields an infrastructure where the communication between distributed controller instances is highly reduced and the resource consumption by each controller instance is minimized [57].

SDN architecture primarily separates the control and data plane. The switches are now responsible only for maintaining the forwarding state of the network. The flow tables of each switches constitutes of all the data plane activities. However, control plane does not keep copies of these flow tables with it and primarily deals with network topology and policy information. However, research work has proved that storing forwarding states in controller can be helpful in quick recovery from failures, reducing overhead in communication channel between switch and controller and providing consistency of network forwarding state in both controller and switch. However, storing such set of network forwarding state in controllers can directly affect the scalability in network and occupy large amount of costly memory. To overcome this, one could reduce the flow table sizes by looking for redundant entries in different flow tables and removing the redundant entries. Also, appropriate lossy compression techniques can be applied to reduce large network state to compressed tables [58].

SDN has traversed all the aspects of networking. However, little has been done to apply the goodness of SDN to cellular (mobile) WAN networks. These networks is divided into large and widespread regions each comprising of packet gateways that handles almost all the network policies. There is minimal interaction between these gateways blocking network-wide

optimization. Long routes in large subnetwork is not suitable for optimal routing which makes cellular WAN less scalable and reliable. Research work has been undertaken to design Cellular WAN architecture leveraging the control-data plane separation of SDN. Data plane would consist of programmable switches making sure there are enough number of egress points to eliminate long suboptimal routes. NFV is used for traffic engineering using software-driven middlebox services. Control plane would comprise of distributed yet logically centralized controllers organized into hierarchical structure with leaf controllers managing the logical regions of data plane. The next level would be exposed by leaf controllers to root controller along with middlebox services, switches and base station it manages [59].

SDN can be leveraged in cloud computing data center architecture. Most of the resources in this pay per use model is assigned effectively excluding CPU and memory usage. These resources are accounted for time being used than for the actual usage. VM once assigned stays idle even when not used and cannot be assigned to a different user in frequent need. SDN controller can be used to suspend idle VMs based on expired flow entries and if a flow entry is missing controller can resume the VM. Thus using SDN, we can reduce the latency when a user has to be provisioned an idle VM [60].

SDN benefits can also be used in implementing futuristic WiFi networks. Today WiFi is most common technology accessed through vivid devices. With ever growing user base, WiFi networks needs to more scalable than ever before. The control plane communication overhead and latency can be reduced by creating multi-tier control plane. Latest SDN research trends shows how 2-tier control plane can be used to overcome above issues in WiFi Network. We can have a controller close to the data plane viz. Access Point that could handle localized events and a Global Controller that would take care of authentication, mobility management, load balancing, various middlebox services and other global events. In addition, the Radio Agent at each AP can control WiFi transmission settings and gather data and statistics related to wireless network. We can have power, retry count and transmission rate per-user, per-flow or per-slice using multi-tier control plane [61].

NFV has steered the challenges of service chaining into the Data Centers. In service chaining, it is extremely difficult to steer small-packet traffic flow through various MiddleBoxes and is less efficient in power usage and configuration. Situation worsens when the same coarse-traffic has to be steered through optical domain providing terabit per second of speed yet lacks buffering schemes and agility. However using NFV, we can virtualize network functions like Serving GPRS support node, Gateway GPRS support node and session border controllers for steering traffic in optical domain effectively. Also, such traffic steering can be designed for data center supporting hybrid packet-based and optical flows [62].

Due to ever growing number of data center applications relying on one-to-many communication, multicast-featured datacenter architecture was evolved which uses optical splitters for fast, efficient and reliable group delivery. However it is only possible to steer data unidirectional. SDN can be leveraged to deliver multicast data over optical splitters channeling other traffic

through different paths. Using SDN, network topology and hence the location of optical splitters can be easily learnt. Also, multicast trees can be quickly developed using already populated group membership. Finally, at application layer, the QoS for applications can be easily negotiated using centralized SDN Controller [63].

SDN benefits has been leveraged across network and data center architecture designs. However, two main challenges that hampers SDN adoption is fault tolerance and reliability. In SDN environment, server hosting the controller can fail, the controller code can be buggy, network devices in data plane could crash or the SDN application could crash due to various reasons. Except for SDN application crashes, rest of the challenges have been vigorously worked upon. SDN Apps and controllers are usually interlinked in SDN design. However, research work has been undertaken to prove that isolation of SDN components is necessary and that SDN Apps and Controller needs to be loosely-coupled and provide a set of abstractions. State maintained by SDN App can get corrupted on crash and can affect other SDN Apps relying on it. This can be handled by isolating SDN Apps from the controller it runs on and from other SDN Apps handled by that controller. We also need to make sure that all the operations done during failure recovery are part of single transaction to avoid inconsistent state in SDN App [64].

4 SUMMARY

Today Software Defined Networking (SDN) has been incorporated in almost all areas of computer networks and Data Centers. It has progressively modified the way internet data has been steered through various data center architecture constructs. Software Defined Data Center takes the goodness of SDN to provide with scalable, flexible, fault tolerant and agile Data Center architecture with maximum resource utilization thus reducing the capital expenditures and operational cost by huge factor. This project report depicts the evolution of Data Center Architecture to everything Software-Defined and also provides with detail explanation on how each Data Center Construct solved the challenges and issues of its predecessor. It also summarizes various research trends with SDN and Data Center Models undertaken by academic researchers and industry experts. The project gives an in-depth explanation of few Use Cases of SDN in Data Center, WAN and campus networks.

SDN has now gained widespread acceptance and new vendor solutions incorporating SDN is surfacing more frequently. As with any technology, SDN brings along its own challenges but however it is the most prominent technology in transforming the networking industry and the operational behaviour of Data Centers of today's world. From the number of SDN research projects and vendor implementation surfacing, it is quite clear that the future of SDN is bright and will definitely steer the Data Center Architectures in solving the ever growing networking challenges for over decades.

5 REFERENCES

[1] J. Kozlowicz, "Green House Data," 21 May 2013. [Online]. Available: http://www.greenhousedata.com/blog/data-center-tiers-explained-the-great-raised-floor-debate.

[2] P. Jones, "Data Center Dynamics," 9 April 2014. [Online]. Available: http://www.datacenterdynamics.com/it-networks/tia-to-remove-tier-from-its-benchmarking-system/86004.article.

[3] W. Gruener, "Tom's IT Pro," 8 April 2013. [Online]. Available: http://www.tomsitpro.com/articles/cloud_computing-modular_datacenter-mainframe-eniac,5-26-7.html.

[4] A. Nutt, "Article Base," 10 September 2008. [Online]. Available: http://www.articlesbase.com/technology-articles/history-of-the-data-centre-556616.html.

[5] J. Woods, "Silicon Angle," 5 March 2014. [Online]. Available: http://siliconangle.com/blog/2014/03/05/the-evolution-of-the-data-center-timeline-from-the-mainframe-to-the-cloud-tc0114/.

[6] S. Miniman, "Wikibon," 6 February 2014. [Online]. Available: http://wikibon.org/wiki/v/The_Data_Center:_Past,_Present_and_Future.

[7] D. Floyer, "Wikibon," 14 February 2013. [Online]. Available: http://wikibon.org/wiki/v/Defining_Software-led_Infrastructure.

[8] M. Bullock, "Data Center Definition and Solutions | CIO," 14 August 2009. [Online]. Available: http://www.cio.com/article/2425545/data-center/data-center-definition-and-solutions.html.

[9] R. McFarlane, "Search Data Center," 2 August 2011. [Online]. Available: http://searchdatacenter.techtarget.com/tip/Considering-a-raised-floor-in-the-data-center.

[10] K. Kant, "Data Center Evolution A tutorial on state of the art, issues and challenges," *Computer Networks,* vol. 53, no. 17, p. 2940, 2009.

[11] S. Kieffer, W. Spencer, A. Schmidt and S. Lyszyk, "Network Systems Architects, Inc.," 2003 February. [Online]. Available: http://www.nsai.net/White_Paper-Planning_A_Data_Center.pdf.

[12] Siemon, Data Center Storage Evolution, 2014.

[13] G. K. Margaret Rouse, "TechTarget Network," 13 August 2014. [Online]. Available: http://searchstorage.techtarget.com/definition/network-attached-storage.

[14] M. Rouse, L. L. Brennan and M. Olanie, "Tech Target Network," 17 August 2014. [Online]. Available: http://searchstorage.techtarget.com/definition/storage-area-network-SAN.

[15] R&M, R&M Data Center Handbook, 2011.

[16] Semsin.com, "Tech Target Network," 16 July 2004. [Online]. Available: http://searchnetworking.techtarget.com/tutorial/The-Cisco-three-layered-hierarchical-model.

[17] Cisco.com, "Data Center Multi-Tier Model Design".

[18] J. Snyder, "BizTech," 10 2013. [Online]. Available: http://www.biztechmagazine.com/article/2013/10/understanding-different-layers-routing-and-switching.

[19] P. Galvin, "Pluribus Networks Blog," 17 July 2012. [Online]. Available: http://www.pluribusnetworks.com/blog/detail/traditional-network-infrastructure-model-and-problems-associated-with-it/.

[20] J. Wallen, "Tech Republic," 10 April 2013. [Online]. Available: http://www.techrepublic.com/blog/10-things/10-benefits-of-virtualization-in-the-data-center/.

[21] J. Mears, "Network World," 22 February 2007. [Online]. Available: http://www.networkworld.com/article/2295569/data-center/the-8-key-challenges-of-virtualizing-your-data-center.html.

[22] S. J. Bigelow, "Tech Target Network," 19 December 2008. [Online]. Available: http://searchitchannel.techtarget.com/feature/Network-virtualization-explained.

[23] T. Lams, "Tech Target Network," 14 April 2005. [Online]. Available: http://searchdatacenter.techtarget.com/tip/Consolidation-and-virtualization-The-same-but-different.

[24] StoneFly Inc., "StoneFly Resources," [Online]. Available: http://www.stonefly.com/resources/Storage-Virtualization.asp.

[25] F5 Networks, Virtualization Defined - Eight Different Ways, 2007.

[26] G. Lemasa and S. Gai, "Fibre Channel over Ethernet in the Data Center: An Introduction," 7 April 2008. [Online]. Available: http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/ieee-802-1-data-center-bridging/white_paper_FCIAFCoE.pdf.

[27] M. Rouse, "Search Storage," 3 December 2008. [Online]. Available: http://searchstorage.techtarget.com/definition/InfiniBand.

[28] HP, "Converged Networks and Fibre Channel over Ethernet," 2012. [Online]. Available: http://h20565.www2.hp.com/hpsc/doc/public/display?docId=emr_na-c03440617-1&docLocale=.

[29] HP, "HP," October 2011. [Online]. Available: http://h20565.www2.hp.com/hpsc/doc/public/display?docId=emr_na-c01681871.

[30] Steve Levine, "Juniper Networks," March 2014. [Online]. Available: http://www.juniper.net/techpubs/en_US/learn-about/data-center-bridging.pdf.

[31] Cisco, "Cisco Data Center Bridging," November 2010. [Online]. Available: http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/ieee-802-1-data-center-bridging/at_a_glance_c45-460907.pdf.

[32] VMware, "Network Virtualization in Multi-Tenant DataCenters (TR-2013-001E)".

[33] P. Lahiri, "OpenContrail Blog," 19 November 2013. [Online]. Available: http://www.opencontrail.org/comparing-network-virtualization-techniques-available-in-the-cloud/.

[34] Brocade Communication Systems, Building a Reliable Foundation for Expanded Data Center Virtualization, San Jose, 2008.

[35] N. Feamster, J. Rexford and E. Zegura, "The Road to SDN: An intellectual history of programmable networks," *ACM,* vol. 11, no. 12, 2013.

[36] D. Wetherall, "Active Network vision and reality: lessons from a capsule-based system," *ACM Symposium,* vol. 34, no. 5, pp. 64-79, 1999.

[37] S. Bhattacharjee, E. Zegura and K. L. Calvert, "An architecture for active networks.," *IEEE,* 1997.

[38] Transparency Market Research, "SDN Market - Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2012-2018," 2013.

[39] Open Networking Foundation, "ONF: SDN architecture overview v1.0," 12 December 2013. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf.

[40] M. Sandorfi and M. Clayville, "The Software-Defined Data Center and the new Hitachi Unified Compute Platform".

[41] B. Kepes, "Gigaom Research," 12 February 2015. [Online]. Available: http://research.gigaom.com/report/the-software-defined-data-center-in-the-enterprise/.

[42] S. Raghuraman, The Journey Toward the Software-Defined Data Center, Cognizant Inc., 2013.

[43] F. Yue, Network Functions Virtualization - Everything Old is New Again., F5 Networks, 2013.

[44] M. Taylor, A Guide to NFV and SDN, Metaswitch Networks, 2014.

[45] J. Metzler, "Webtorials Analyst Division," 27 January 2015. [Online]. Available: http://www.webtorials.com/main/resource/papers/webtorials/2015-Guide-to-SDN-and-NFV/2015_Guide_Complete.pdf.

[46] Google Inc., "Inter-Datacenter WAN with centralized TE using SDN and OpenFlow," 2012.

[47] S. Garrison, "Emerging Use Cases for SDN," 20 February 2015. [Online]. Available: https://www.sdxcentral.com/articles/contributed/emerging-use-cases-for-sdn-steve-garrison/2015/02/.

[48] B. Harzog, Business Benefits of the Software Defined Data Center, The Virtualization Practice, 2013.

[49] B. Pankaj, G. Matteo, H. Jonathan, H. Yuta, K. Masayoshi, K. Toshio, L. Bob, O. Brian, R. Pavlin, S. William and P. Guru, "ONOS: Towards an Open, Distributed SDN OS," in *ACM SIGCOMM Workshop*, Chicago, 2014.

[50] M. Hesham, H. Fang, M. Sarit, Z. Zhi-Li and V. L. T, "Application-aware Data Plane Processing in SDN," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[51] H. Ryan and K. Eric, "ClosedFlow: OpenFlow-like Control over Proprietary Devices," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[52] J. Hani, W. Dan and S. Upendra, "Don't Call Them Middleboxes, Call Them Middlepipes," in *ACM Sigcomm Workshop (HotSDN 2014)*, Chicago, 2014.

[53] G. E. Thomas and B. Warren, "Using SDN To Facilitate Precisely Timed Actions On Real-Time Data Streams," in *ACM Sigcomm Workshop (HotSDN 2014)*, Chicago, 2014.

[54] B. Ilya, H. Shu and G. Rajesh, "A Resource Delegation Framework for Software Defined Networks," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[55] H. Hongxin, H. Wonkyu, A. Gail-Joon and Z. Ziming, "FlowGuard: Building Robust Firewalls for Software-Defined Networks," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[56] S. Arne and K. Holger, "Using MAC Addresses as Efficient Routing Labels in Data Centers," in *ACM SIGCOMM workshop (HotSDN 2014)*, Chicago, 2014.

[57] K. Anand, P. C. Shoban and G.-J. Aaron, "Pratyaastha: An Efficient Elastic Distributed SDN Control Plane," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[58] Z. Ying, N. Sriram, H. Xin, B. Neda and M. Ravi, "A compressive method for Maintaining Forwarding States in SDN Controller," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[59] M. Mehrdad, E. L. Li and M. Z. Morley, "SoftMoW: A Dynamic and Scalable Software Defined Architecture for Cellular WANs," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[60] K. Thomas, K. Pradeep, H. Matti and F. Christof, "Sloth: SDN-enabled Activity-based Virtual Machine Deployment," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[61] S.-Z. Julius, S. Nadi and S. Stefan, "Towards a Scalable and Near-Sighted Control Plane Architecture for WiFi SDNs," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[62] X. Ming, S. Meral, Z. Ying, G. Howard and T. Attila, "SOLuTIoN: SDN-based OpticaL TraffIc steering

for NFV," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[63] Y. Xia and E. N. T.S., "A Cross-Layer SDN Control Plane for Optical Multicast-Featured Datacenters," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.

[64] C. Balakrishnan and B. Theophilus, "Tolerating SDN Application Failures with LegoSDN," in *ACM SIGCOMM Workshop (HotSDN 2014)*, Chicago, 2014.