POLITECNICO DI TORINO

III Facolta' di Ingegneria dell'Informazione

Corso di laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Evolution and design of a modern Datacenter

Relatore:
Fulvio Risso

Candidato:
Francesco Marabotto

Dicembre 2020

Acknowledgements

I would like to thank professor Risso for his patience and allowing me so much time to finish this document: as people foresaw, work and private life took over, leaving little time for this work. I am happy to have had the chance to keep working on this.

Thanks to my parents, that always pushed me to finish this work and supported me in the years while at university and abroad working.

Thanks to my wife Sarah for the infinite patience she has with me every day and the support.

Thanks to Cisco systems, especially the Belgium office, for allowing me the tools to learn, experiment and build the datacenter whose design is described in this document.

Thanks to the 3rd faculty of engineering of Turin Polytechnic University for teaching me all I needed to know to successfully allow me to build a career in the field I love.

Abstract

Since the beginning of the computer industry, particular attention has been given to facilities where a huge amount of storage and processing power was installed and maintained. Those rooms were (and are) containing the core of an enterprise, a university or a government since information has become the measure of the wealth of an organization and the most important asset to protect.
This thesis will discuss the evolution of the datacenter from the earlier installations in the ’60 and ’70 to the modern day. It will analyze the changes that happened during the years, will describe how a modern datacenter is built and which parameters dictate the design choices behind a facility that can easily cost up to a hundred million dollars.
Particular attention will be given to the design rules and best practices used to achieve the requested reliability levels and practical examples will also be described by using state of the art equipment to implement the designs, simulate failure situations and perform tests.

Table of Contents

Introduction

In 2017, the Economist published an article [1] stating that oil is not the most valuable resource anymore, but data overtook it.
A year later, Forbes estimated [2] that 2.5 quintillion bytes (10^18) of data are created every day.
Producing and consuming such an amount of data was unthinkable even just 10 years ago and all of this is due to the ubiquity of technology like mobile phones, the internet and the digitalization of documents.
Datacenters [3] store, process and create new data, so it is not like a finance institution that stores and reinvest funds to produce more funds rather than a bank that mostly just stores cash.
In this work I'm going to discuss the evolution of datacenters from their beginning to the current mainstream technologies that can be found in the majority of enterprises.
I'm going to do this by describing a datacenter that I've built during my professional experience that includes IP Networking, Storage Networking and Compute and Virtualization.
I will be describing some less known technologies like FC [4] and Fabric Path [5] and offer a fully working example of a complete datacenter that can be easily scaled up horizontally with real configuration.

What is a datacenter?

While giving a precise definition is not possible, a datacenter can be considered a physical location containing massive amounts of storage and processing power used to store the data of an enterprise and/or provide services to customers or to the enterprise itself.

A datacenter is composed of various components, each one with specific goals and clearly separated from each other. The basic systems we should expect in a datacenter are: servers, storage in the form of arrays of disks (just a bunch of disks, JBOD), communication equipment as switches and routers, power sources including backup systems a cooling system, a security system and a physical location that is compatible with the previous elements.

Let's analyze the components in more detail:

a) Servers: the servers are the core of the datacenter. They provide the computing power to perform operations on the data contained in the storage. The servers are the element using the biggest amount of electrical power, cooling and space. Servers have, in average, a lifespan of 3 years, after this amount of time, the machine is usually obsolete and may need to be replaced based on the requirement of the new software employed in the enterprise or in some cases, due to hardware restrictions acting as a show stopper for new applications.

   As of today, servers come in two formats: rack mount or blades.

   Rack mount are servers ranging from 1 to 5 U (a U is the standard unit of measure of the height of a computer enclosure in a standard rack. A U equals to 1.75 inches). Rack mounted servers possess their own power supply, network connectivity (NIC), DVD reader, USB connections, VGA output and usually storage (in the form of hard drives, with SAS or SCSI interface). In case of centralized storage solution, a host bus adapter (HBA) is used to connect to Fibre Channel (FC) storage, or a standard NIC for iSCSI/network attached storage (NAS). Rack mount servers possess PCI express slots for further expansion often used for nonstandard features like crypto engines, SSL accelerators and possibly audio/video cards or GPU. Modern servers may also be equipped with converged network adapters (CNA) that integrate Ethernet and Fibre Channel controllers in a single interface and enable the use of Fibre Channel Over Ethernet (FCoE, discussed later).

Figure 1: Cisco UCS C200 M2 Rack-Mount Server (source cisco.com)

Blade servers are modules that can be inserted into a chassis. Each blade contains (at minimum) CPU, memory and an interface to connect it to the backplane of the chassis. The external connectivity is obtained through the chassis as one or more specialized slots are used to host a switch or a NIC that provides external interfaces toward the Ethernet/storage network. The chassis then interfaces the external NIC to the backplane where the blades are connected. The Operating System (OS) on the blade sees the backplane interfaces as they were normal NIC/HBA. The number of those NIC depends on the backplane throughput or on the type of the backplane interface used (we will discuss later about the virtualization of the network interfaces). Even if some blade server vendors give the possibility of having local storage in a blade, the size limitation of the blade limits the capacity to a level that may not be sufficient for the application thus forcing the blade server to usually rely on storage networks. Each chassis vendor has its own blade switches and servers; different vendors' blades are usually not compatible.

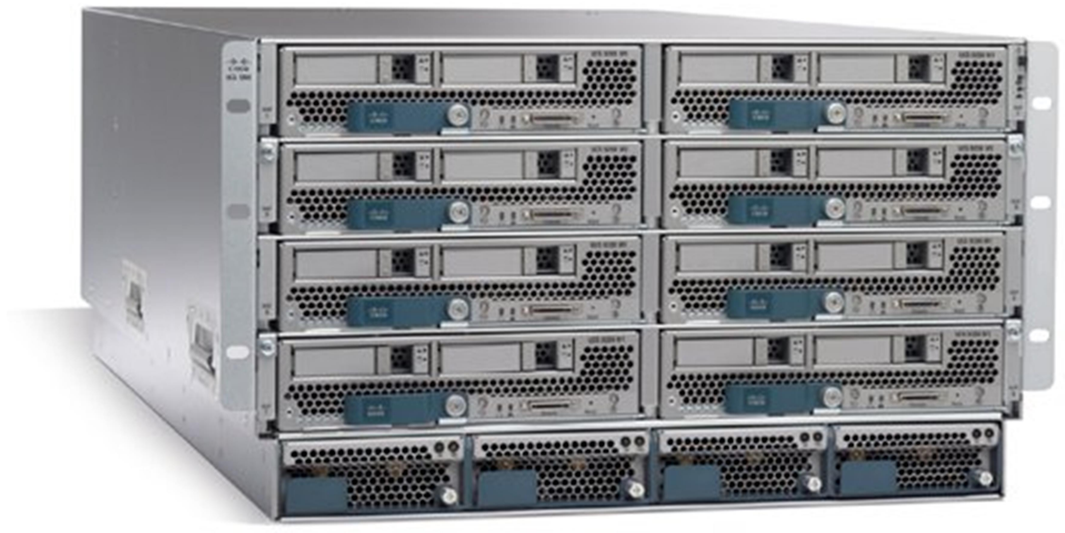


Figure 2: Cisco UCS 5100 Blade Server Chassis equipped with 8 UCS B200 M2 Blades (source cisco.com)

b) Storage: the storage is the most critical part of an enterprise. It contains the information that, as we already stated, are the most important asset of any enterprise. It is absolutely unacceptable for any government or enterprise to lose data and even if we think of a company that makes most of its revenues through manufacturing, procedures, account and statistical data must be stored; imaging what would happen if such enterprise would lose the list of the customers it served during the years. The first way to protect storage is redundancy. Having multiple synchronized copies of the data we need to protect will help us recover from issues. Data redundancy comes in different flavors: backup, redundancy codes (CRC), redundant arrays of inexpensive disks (RAID). Without losing too much time on these topics, all of these techniques are used together in the datacenter to securely store data. The best practice when building a modern datacenter is to use a centralized storage that can be accessed by all components through a separate network (storage

access network [6], SAN [7]). This allows the designer/administrator to implement centralized policies for the storage thus reducing the costs and the maintenance needed.
Multiple racks, each containing hundreds of disks are connected together, disks are grouped in volumes (which are made redundant through RAID and CRC), volumes are divided in logical unit numbers (LUN) and LUNs are accessed via the SAN network by the servers. Backups are performed via the SAN network between arrays in the same datacenter and in other datacenter geographically separated for disaster recovery purposes (discussed later).

Figure 3: EMC VNX Storage Array (source emc.com)

The data on the array is usually accessed as blocks; an operating system "mounts" the network disks like they were local and it access them via SCSI protocol or as file using network storage system (NFS) or common internet file system (CIFS/SMB).
Blocks access allows for faster access and lookup of the files as the OS can access the portion it is interested in without reading the whole file while file access allows for better lock of a resource. Another important difference between the two is that file access relies on a transport protocol as TCP that allows for packet losses by implementing a retransmission feature while block access doesn't allow packet loss as it virtualizes the hard drives to the systems like they were local and the OS doesn't have any feature to understand if a block of data has been lost while writing to the disk. We will discuss later about this important difference, but it is crucial understanding that loss of a packet in a block based access means corruption of a file and the loss of connectivity to an hard drive means a crash of the system (what would happen if you would yank out the hard drive of your computer without having any local redundancy?): a Fibre Channel storage network doesn't allow for packet loss. To achieve this, a complex congestion control mechanism is implemented in the FC protocol to make sure that there are enough resources available to allow the packets to reach the destination or gracefully abort the write/read operation.

c) Communication equipment: servers must be connected together to work as needed and storage must also be reachable. Communication equipment solves these issues by providing devices able to switch packets between the end systems and with the appropriate level of redundancy in order to limit the damages in case of an accident (link going down or crash of a device).

   On the network point of view, the datacenter is a relatively simple environment: the network is usually a tree where the root is a pair of powerful layer 3 switches and the leaves are the servers. A single layer of nodes are implemented to separate the root from the leaves.

   The network in the datacenter is layered, with usually two three layers: core (root), distribution (nodes) and access (edge nodes connected to the servers), but in smaller datacenters, core and distribution are collapsed. The access network is usually a pure layer 2 network that connects the servers to the network. The distribution is a mixed layer 2/layer 3 network used to provide a default gateway to the server attached to the access layer. The core is a highly redundant, pure layer 3 network used to connect the distribution networks. Of those three layers, the distribution is the most complex due to its mixed layer 2 and 3 level and it is used to integrate all the additional services that the network requires such as load balancing, security, access to internet, TCP compressions and WAN optimization.

   The path from the root to the leaves is always redundant. The redundancy is obtained by using ether channels (also referred to as port channels, Ethernet bundles or link aggregation groups or LAG). Because the network is designed as a tree, the spanning tree protocol (STP [8]) doesn't really play a role ensuring a major stability, faster and predictive convergence (discussed later). STP however is always kept active as a security measure in case of a loop being created due to human error.

   Datacenters nowadays have 10 Gigabit Ethernet access layers and this is moving forward toward 40 Gigabit Ethernet access network and possibly 100 Gigabit Ethernet in the core.

   The layer of the datacenter network will be described in major details after.

Figure 4: Cisco Nexus 7000 Switches (source cisco.com)

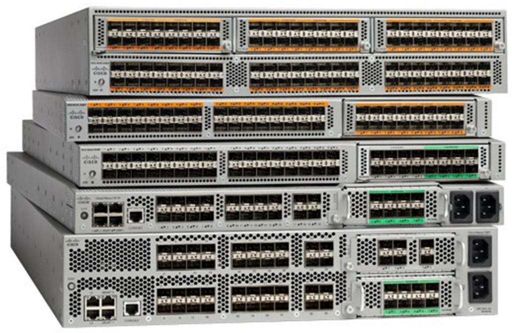


Figure 5: Cisco Nexus 5000 Switches (source cisco.com)

The storage network is as well implemented as a network of switches depending on the protocol used. Protocols like iSCSI, SAN, SMB and CIFS are usually implemented over Ethernet and transported on the same network as the normal data network, while FC networks require particular switches able to handle the FC protocol. FC networks implement redundancy differently from Ethernet; usually by having two physically identical, but completely separate networks thus giving the same (multihomed) host two different paths to each storage target (multipathing). The choice of which path to use and the actions to be taken in case of failure of one of those paths are delegated to the server through the use of a multipathing software on the same.

Figure 6: Cisco MDS 9000 Fibre Channel Switches (source cisco.com)

In the latest years various international and multi-vendor committee tried to unify the FC and the Ethernet network in the effort of building Datacenters with a single network instead of two and they created the Fibre Channel over Ethernet protocol that is a protocol stack used to encapsulate FC frames in Ethernet and provide Ethernet with a hop by hop congestion control in order to simulate the lossless capabilities of the FC networks.

In case of FCoE, the servers are equipped with a CNA that appears to the OS as a NIC plus a HBA, but they require a single cable (compliant to Ethernet specification) to carry both FC and Ethernet traffic.

d) Power sources: datacenters require an enormous amount of power: a core switch can use 7.5 kW per power supply and be equipped with three or four power supplies. In a datacenter, the power is given by the plugs contained in each rack. Racks usually have redundant plugs, one connected to a power source, the other to a secondary one, plus a backup source. A rack must be able to failover and sustain all the devices installed into it on a single power supply, it is then very important to not install too many (or too power intensive) devices in order to be able to sustain them all in case of failure.

Figure 7: Cisco Nexus 7000 7.5 kW AC power supply (source cisco.com)

Datacenters are also equipped with diesel generators as a backup source of power in case of a complete failure. Those generators must be able to sustain the critical systems for the amount of time needed to failover the services to another datacenter.

A standard 44 U rack completely filled up with blade servers (about 56 Cisco UCS B-200 blades) with redundant power supply may need up to 20 kW per power source.

Some bigger datacenters have started using more than 1 GW of power.

e) Cooling system: if we multiply the 20 x 2 kW rack by 200 we can imagine the amazing amount of power that a whole datacenter requires. We perfectly know that even the most efficient of the electronic device dissipate some energy in the form of heat. On top of this, datacenters are usually very dense and this doesn’t help cooling.

Huge ventilation systems are installed in the datacenters in order to prevent the electronic component from failing due to the high temperatures reached during peak or prolonged usage. The cooling systems are usually implemented as a series of fans attached to the ceiling shooting cold air downward between the streets (a street is a series of adjacent racks). The cold air enters the devices and then is expelled on the other side of the street where another fan will be installed with the duty of eliminating the heated air that naturally tends to go up.

As a consequence, proper installation of the devices in the rack is required in order to have all devices of parallel, facing streets to have the exhaust in opposite directions so they inhale the cold air to avoid the heated air exiting a device to enter another device and overheat it.

When designing a datacenter it is important to know what the racks will have to contain a proper cooling system because a too small cooling system will limit the amount of installable devices in a rack.

Sizing of the cooling system is done based on the size of the density of the racks and the predicted exhaust and it is measured in W or BTU (British Thermal Units in non-metric countries, 1 W = 3.41 BTU/h).

f) Security systems: hosting the most precious assets of an enterprise, the datacenter has to be protected from any possible danger. While network security is not that important in the design of a datacenter (mostly because it is limited to creating DMZ in a firewall and protecting the access from the WAN), particular attention is given to the physical security of the datacenter. Access to the infrastructure is always strictly controlled via authentication through badges or biometric devices (fingerprint or eye scanner). The most secure datacenters require third party intervention to be accessed as they don't have lights or the quantity of oxygen in the air is not enough for a human to survive (in order to prevent fires).

g) Physical location: finding a suitable physical location for a datacenter is challenging as companies are usually targeting countries that offer low taxes or offer particular facilities. Geographical characteristics are also kept in mind as the climate of an arctic country may lower the costs of cooling the structure while building in a seismic zone will for sure influence the creation of a contingency plan.

Once the location has been chosen, the builder needs to create a structure that is big enough to sustain the weight of the racks (the chassis alone for a modular core switch weighs about 90 kg). The racks are usually installed over a raised floor made of panels. Below those panels, power cables, cooling pipes and some backbone network connections are passed.

The raised floor is certified to sustain a specific amount of weight, as per the cooling system, this has to be taken in consideration to avoid having racks too heavy that could break the floor and limit the vertical utilization of the datacenter increasing the surface size and thus the cost of ownership.

History of the datacenter

In this chapter we will go through the major steps that helped define the datacenter as it is now. We will analyze the steps that the datacenter went through starting with the initial computer era to the modern virtualization systems. We will also have a look of what the immediate future will be for the datacenters and which will be the new technologies that we should expect to find in the next 2-5 years.

Mainframes

The term datacenter was created about 50 years ago during the “mainframe era”. Mainframes were the first example of computers that the enterprises used for their business purposes, being storing the data of the company, hosting applications used to run the company or helping the employees create products.

The mainframes originate directly from the first examples of computers built during the fifties, colossal machines usually taking a whole room of space. With the advent of microprocessors and miniaturization, the devices became relatively smaller, cheaper and this opened their way to be commercially viable. The first examples of enterprise mainframe were seen in the sixties in the US. The main “player” of the mainframe business was (and still is) IBM.

The mainframes offered back then an extremely high level of performances and business continuity: hardware upgrades/substitutions were doable in service and without any impact to the applications. Same applies to OS and software upgrades. Load was shared among multiple mainframes and applications could “migrate” seamlessly from device to device.

Figure 8: IBM Mainframe (source: ibm.com)

It is amazing to see how the mainframe invented over 50 years ago almost all the concepts that we can find in the modern datacenter: IBM zSeries offered among other features, a complete hardware partitioning of the system (LPAR) and virtual machines using their proprietary UNIX-like operating system. Also the Virtual Desktop Infrastructure (VDI), a modern approach to handling laptops and desktop computers in the IT organizations was partially invented back then where the mainframes was hosting applications for multiple users and it was accessed via remote terminals similar to today's thin clients.

Even if any modern server is more powerful than an old mainframe, the market of those products is not yet dead, multiple enterprises built businesses on applications running on mainframe that are so critical that the risk of migrating them to a newer system is so high that they prefer to invest in maintaining the old mainframe infrastructure. Wal-Mart for example has multiple tens of megabit per second of traffic of credit card numbers going to a mainframe, interrupting those flows for a migration is unacceptable for them.

Microcomputers

The microcomputer era started in the mid-seventies as an alternative to mainframes. Microcomputers are the smaller, slower, but less expensive devices that could be adapted to run business applications.

The low cost of microcomputers allowed the companies to deploy lots of machines in different branches and then buy network equipment to connect them to a datacenter containing the mainframes that were still running the most intensive application or storing the shared data required by the software running on the microcomputers.

The microcomputers were not deployed in a centralized, controlled environment as the mainframes and this caused the creation of a somewhat chaotic environment with multiple rules and procedures needed to create order and organization. On top of this issue, multiple mini computers manufacturers created different and incompatible operating systems. An Operating System was unable to run applications written for other Operating Systems or compiled on a different architecture.

As IT was growing in importance around the world, data losses and availability were starting to become a major concern, the fragmentation of technologies, the different OS and the lack of a central organization to mandate standards caused lots of issues and a solution was needed.

To solve this problem, the server client paradigm was created. Servers are mainframes or more powerful micro computers used to run the business critical application and where the micro computers were connecting to upload or retrieve data. Servers are centrally managed in a datacenter and this allows lower downtime by implementing rules and change management. Servers are accessed by using a protocol, a shared language between the application running on the server and the client. These protocols are transported over a network using a packet switching or circuit switching technology. This allows for the maximum interoperability as the applications “speak” a language that is independent of the Operating System or the architecture of the machine they are running on and no knowledge of the transport media is needed.

The internet datacenter

The server rooms were becoming bigger and bigger. More applications had to be created and more and more clients were using them.

Of the various protocols used to transport the application data, TCP/IP was by far the most successful, killing the competition composed of apple talk, IPX, SNA and others.

The Internet was creating new markets and new needs. Application had to be running 24/7 because customers were able to purchase goods from their home at all time, including the weekend. Websites like Amazon are serving customers in any time zone; an online shop could just not shut down during the night.

This brought various challenges to the datacenter administrators: how to scale the datacenter with the new customer requesting everyday new services? How to operate the datacenter 24/7? How to deploy new systems without network or application interruption? How to make the datacenter more reliable? How to load balance traffic and requests among multiple servers when the single server is not able to cope with the load?

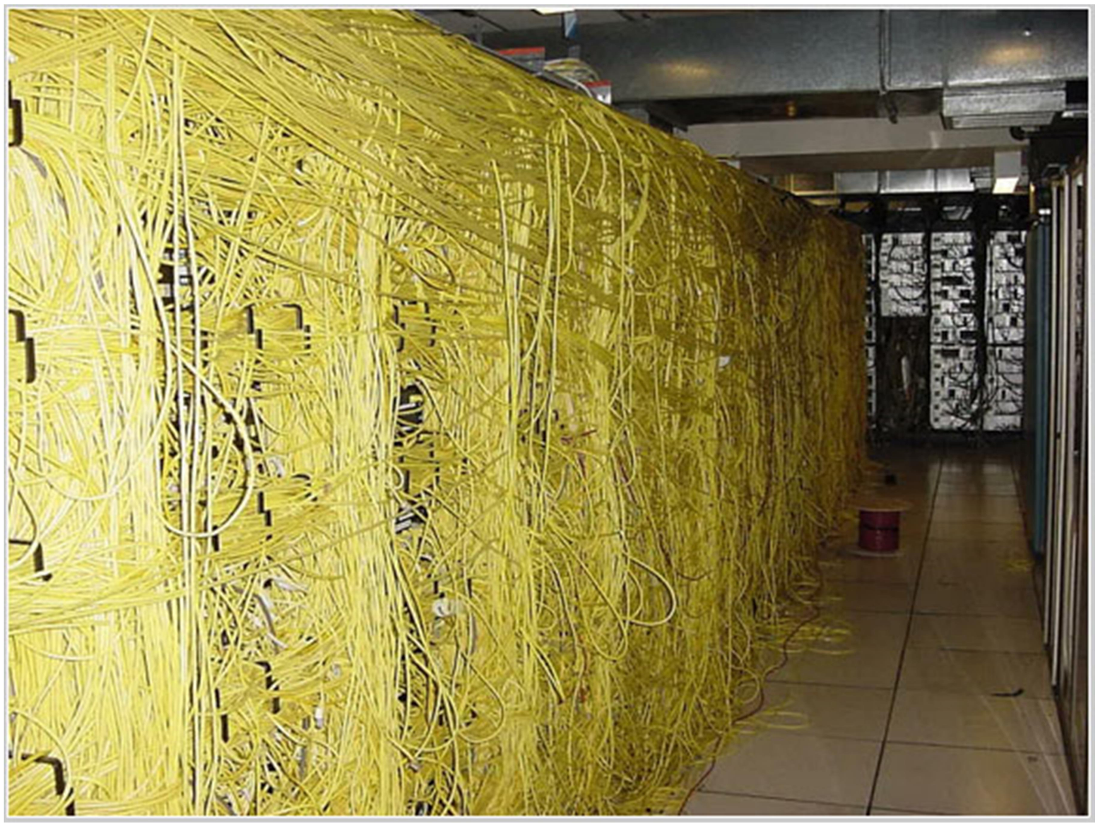
In this period, we are starting to see what resembles a modern, current datacenter. The datacenter becomes a common word among IT and huge specialized spaces are being used to build facilities to store the machines.

We need to remember that at the time the first internet datacenters were created, there was not x86 virtualization available: the only machines able to virtualize were the mainframes, but as mainframes are very expensive and critical machines, built for stability, they were not used to deploy internet based commercial application. One of the motivations of this is that a mainframe is a closed architecture, heavily tested and nothing is put in production before months of regression testing and simulation. This approach is not compatible with an environment that rapidly changes and needs multiple updates of the software, patching and new applications with daily frequency.

The lack of virtualization was forcing the server administrator to install the Operating Systems directly on the hardware (also called bare metal). One server may have not been able to run multiple applications because, for example, different applications were using the same TCP/IP ports or they were requiring too many resources or they were just tested to work on a particular version of an operating system.

This lack of virtualization caused the first datacenter to be way more expensive than the current one because for servers with particular applications, not compatible with others, we may see low hardware utilization, so a waste of resources. This larger installed base of servers of course causes a raise in the costs of power and cooling on top of the larger physical size of the datacenter (as the space can be leased and not owned by the company).

Having so many devices also requires a huge amount of ports on the network equipment used to connect them and lots of cabling. This is an example of what a datacenter can become:



Structured cabling and blade servers helped a lot in creating a cleaner datacenter and increased the uptime of the services, but the big push forward is virtualization and consolidation.

The modern datacenter

The modern datacenter is the child of the dot com bubble. When the bubble exploded in the early 2000s, the companies saw that they couldn't invest the same amount of money in the IT budget as before just because their growth expectations were greatly lowered.

This general collapse of the internet economy killed the crazy spending and created a new need for optimized solutions to cut costs in order to maintain the high margins.

New requisites

The modern datacenter must be available and cost as little as possible.

Datacenters are benchmarked to measure their availability. Depending on the application requirement, datacenters are designed for different levels (tier) of availability.

Datacenters are categorized as:

- Tier I: composed of a single path for power and cooling distribution, without redundant components, providing 99.671% availability.
- Tier II: composed of a single path for power and cooling distribution, with redundant components, providing 99.741% availability.
- Tier III: composed of multiple active power sources and cooling distribution paths, but only one path active, has redundant components, and is concurrently maintainable, providing 99.982% availability.
- Tier IV: composed of multiple active power sources and cooling distribution paths, has redundant components, and is fault tolerant, providing 99.995% availability.

A Tier IV datacenters has an allowed downtime of 5 minutes per year. This means that the sum of all downtimes of any services provided by it can be a maximum of 5 minutes per year and that includes periods where changes are being made like maintenance windows.

This downtime measurement includes as well all the time taken by protocol to converge in case of an event: rapid spanning tree for example takes about 3 seconds to converge, this time is almost unacceptable in an environment where the maximum allowed cumulative downtime per year is 5 minutes. Companies usually don't think that more than 1 second of downtime is acceptable, this rules out spanning tree as a means to allow redundancy, relegating it to a backup measure in case of bigger problems (loops), but redundancy is implemented in other ways.

Datacenters are usually paired for disaster recovery and business continuity. This means that if a datacenter is completely destroyed or isolated due to a natural disaster (also called act of god), a terroristic attack or a human mistake, the services can be brought up/transferred in a reasonably short time to another datacenter located in a different geographical location.

Virtualization

Virtualization [9] (or virtualisation in UK English) is a technology that existed since AIX and LPAR implementation on Mainframes technology, but has only recently become mainstream mostly because VMware implemented the first working virtualization solution for the x86 architecture.

The need for virtualization comes from a need to reduce IT costs by better utilizing the hardware. An OS installed on bare metal is usually idle even when applications are running; CPU utilization is about 5 to 10% and the same is true for I/O to disk. Servers nowadays are every day more powerful and fast, it is not difficult to find machines in a 4-5 U chassis that can hold 1 Terabyte (TB) of memory and have 4 to 8 sockets containing CPUs with 4 to 8 cores. There is no operating system or no mainstream application (let's exclude high performance or scientific environments) that can use all that computing power.

On top, in the first datacenters, the need of having 2 different OS or 2 different applications almost certainly required to install another server with heavy costs on hardware, cabling, maintaining, buying or licensing new ports on the network equipment, cooling and power.

Virtualization is a technology that helps solve those problems by allowing multiple software environments to coexist in the same physical machine.

Virtualization comes in different approaches: full virtualization, para-virtualization and OS-level virtualization.

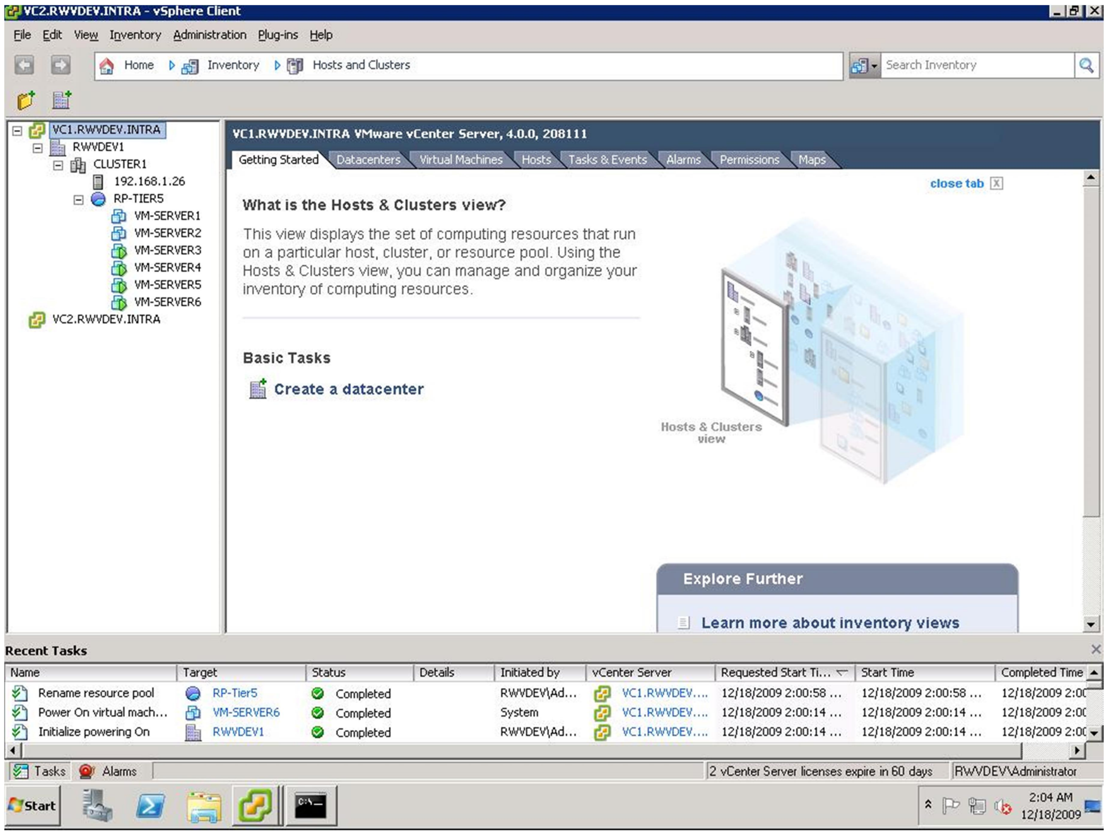
Full virtualization is the simplest one and the most deployed. It is based on the concept of hypervisor: a software or a special purpose operating system (host) that runs on the bare metal emulates hardware properties to a guest OS run as a process in a separate space of memory. The hypervisor exposes virtual hardware to the guest OSes where IO and system calls are then mapped to the real hardware where the hypervisor runs. Examples of hypervisor are VMware ESX [10], VMware ESXi, Microsoft Hyper-V and Xen.

OS-level virtualization is a technique where the kernel of an operating system allows for the creation of multiple user spaces all coexisting at the same time. An example of this is chroot on UNIX and Linux machines.

Para-virtualization is a technique that allows a guest operating system to bypass the hypervisor and communicate with the hardware directly. In para-virtualization, a guest OS knows that it is a virtual machine and thus is able to use particular system calls to use the hardware without passing through the software layer of the hypervisor, while in full virtualization, the guest OS has no idea he's not running on bare metal.

As already mentioned, today's most used virtualization technique is full virtualization. The hypervisor is a lightweight Linux based OS (VMware ESX and Xen) or Windows base OS (Hyper-V).

The hypervisor is installed on bare metal and once basic network connectivity is established, some management services are exposed to allow the process of configuration. Usually the configuration is stored in XML files or on a (external) database. The hypervisor can be configured usually via a command line interface (CLI) over telnet or SSH or via a GUI that communicates with it through HTTP or HTTPS. The CLI option is usually kept as a last resource for troubleshooting purposes or to allow for scripted configuration and deployments or vast amounts of VMs.


VC2.RWVDEV.INTRA - vSphere Client
File Edit View Inventory Administration Plug-ins Help
Home Inventory Hosts and Clusters
Search Inventory
VC1.RWVDEV.INTRA
RWVDEV1
CLUSTER1
192.168.1.26
RP-TIER5
VM-SERVER1
VM-SERVER2
VM-SERVER3
VM-SERVER4
VM-SERVER5
VM-SERVER6
VC2.RWVDEV.INTRA
VC1.RWVDEV.INTRA VMware vCenter Server, 4.0.0, 208111
Getting Started Datacenters Virtual Machines Hosts Tasks & Events Alarms Permissions Maps
close tab
Basic Tasks
Create a datacenter
Hosts & Clusters view
Explore Further
Learn more about inventory views
Tasks Alarms
Start
2:04 AM 12/18/2009

What is the Hosts & Clusters view?

This view displays the set of computing resources that run on a particular host, cluster, or resource pool. Using the Hosts & Clusters view, you can manage and organize your inventory of computing resources.

Recent Tasks

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Ti...	Start Time	Completed Time
Rename resource pool	RP-Tier5	Completed		RWVDEV\Ad...	VC1.RWVDEV....	12/18/2009 2:00:58 ...	12/18/2009 2:00:58 ...	12/18/2009 2:0
Power On virtual mach...	VM-SERVER6	Completed		System	VC1.RWVDEV....	12/18/2009 2:00:14 ...	12/18/2009 2:00:14 ...	12/18/2009 2:0
Initialize powering On	RWVDEV1	Completed		RWVDEV\Ad...	VC1.RWVDEV....	12/18/2009 2:00:14 ...	12/18/2009 2:00:14 ...	12/18/2009 2:0

2 vCenter Server licenses expire in 60 days RWVDEV\Administrator

Figure 9: example of a management GUI for a hypervisor (vCenter, source: vmware.com)

Once the hypervisor has been installed and a configuration method chosen, the administrator starts deploying virtual machines. Each virtual machine has a series of virtualized hardware components that the guest OS sees as hardware and as for real hardware, device drivers are provided for the proper utilization of those components. It is very important to remember that as the components do not exist, it is easy to add or remove additional devices from the VM: hypervisors support adding and removal of CPUs, RAM, NIC, hard drives, serial ports, USB ports and other components. Some components can be added at runtime (like NICs) while others require a reboot of the VM (CPU and memory). The virtualized hardware also adds some interesting functions like the possibility of increasing the hard drive capacity at run time (if the guest OS supports it).

As the guest operating system is in total control of the host, the host has access to the complete memory space and hard drive content. During the years this allowed the development of features that allowed a VM to be migrated from a hypervisor to another without downtime. This is done by having two or more hypervisors to share the same data store where the VM files are located and be able to encapsulate the content of the memory and the state of the CPU within a network protocol and transmit it to another hypervisor. Once the memory has been moved, the first hypervisor stops running the virtual machine and the other takes over. MAC address tables of the switches are updated through the use of gratuitous ARP responses and only a limited number of packets toward the VM are lost in the transition. This feature allows for load balancing of the workload among hypervisors as it allows moving of VMs that require more processing power to less loaded hosts and simplify maintenance as a whole hypervisor can be emptied of VMs before patching or reloading it.

Motion of VMs is usually implemented as a layer 2 protocol and this has influence on the design of a datacenter as it requires having trunks to carry over the control traffic for the migration in what is usually a Layer 3 environment.

Motion of VMs has also lots of time related constraints as the latency for the migration should be very low thus limiting the applicability of the feature to DC located within a relatively small geographical distance. The motivation for this resides in the fact that once a migration operation has started, the memory of the VM is “frozen” until it has been copied over to the other hypervisors. During this operation the VM that is being migrated still runs and changes in the memory are stored by the supervisor as “diffs” in separate structures. Once all memory has been migrated, the hypervisor starts sending those “diffs” over to the new hypervisor. Memory migration is critical and needs to succeed and replicate perfectly on the new hypervisor, so the protocol carrying it needs to be acknowledged. If memory “diffs” are produced at a speed that is superior to the one that takes for acknowledgement to be received, the migration will never complete. This is why delay in the network is important and a highly used VM takes more time than an idle VM to be migrated.

Mobility of VMs created the concept of “cloud”. Without going into marketing speech, the cloud can be considered an environment where services are guaranteed the capacity of migrating to ensure continuous availability and load balancing of resource utilization.

Simply speaking, the cloud is a cluster of hypervisors configured to allow virtual machine migration from a host to another without loss of information or downtime. It is difficult to know where a certain service is running, but it is not that important as long as it is running properly.

As VMs need one or more network connections, the networking configuration needs to follow the migrations of the VM in the cloud. Hypervisors implement software switches to allow vlans mapping to VM NICs and VM can only successfully migrate between hypervisors that have the same network configuration so that when a VM migration has finished, the vlan mapping to its NIC is respected and the guest has access to the network. To simplify this process, hypervisor vendors have invented the concept of distributed virtual switches.

A distributed virtual switch is a software switch that interfaces with hypervisors to support automatic configuration of the networking when a VM migrates. When a VM is configured, the connectivity is configured by assigning the VM to a port group, a list of vlan and properties for that port (bandwidth and QoS settings). When the machine is migrated to another host configured to be part of the same distributed virtual switch, the networking configuration of the VM moves with it.

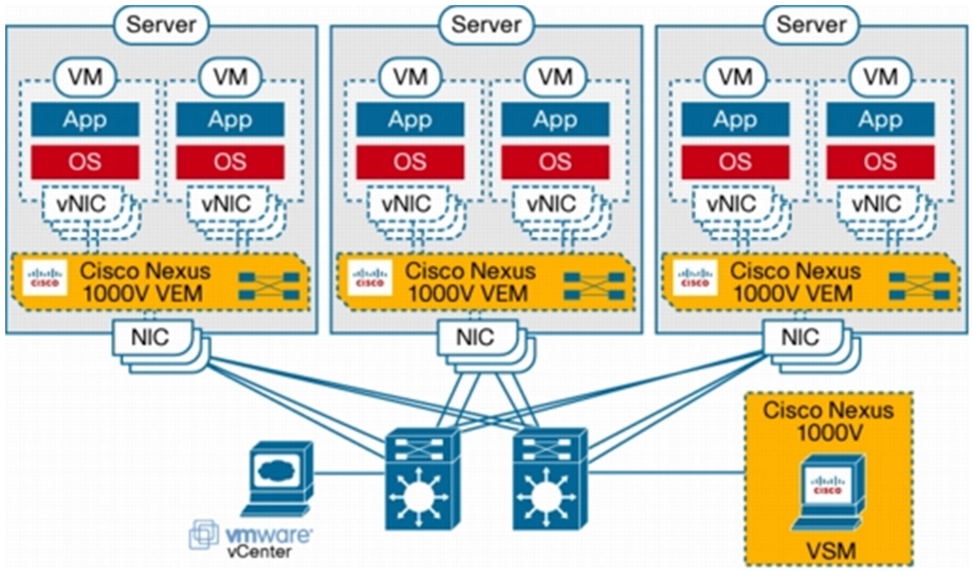


Figure 10: high level concept of a distributed virtual switch (Cisco Nexus 1000v, source: cisco.com)

The green datacenter

The green datacenter is a change in the building and design philosophy datacenters as response to the environment challenges [11].

The reason to exist for a green datacenter is mainly the following: the enterprises are under pressure to reduce the carbon footprint of the datacenters. As a datacenter consumes a huge amount of energy to power the servers, network equipment and cooling, the amount of CO2 used to produce that power globally is about 1% of the total world emissions (out of a total 4% produced by IT). We should expect that number to increase fast and steadily given the growth ratio of datacenters.

Companies and governments are recently approaching the green datacenter idea with (mainly economic) interests.

Enterprises can negotiate tax discounts with the government based on the costs used to cover the construction and operation of green datacenters and the green datacenter itself has economic advantages in the long run after an initial investment that is usually bigger than the one needed for a normal datacenter.

Green datacenters can leverage the natural environment where they are built:

- Datacenters built in north of Europe or US can leverage the low external temperatures (at least in winter) to help with the cooling of the equipment or having conductive material composing the roof to help dissipating heat
- Datacenters built in hot country can have solar panels installed on the roof to produce part of the energy used to power up the equipment
- Datacenters built on windy landscapes can leverage wind power similarly as the ones built in sunny countries
- Datacenters built in non-underground facilities can incorporate windows to let the natural light into the building and turn on lights (if needed) only at night

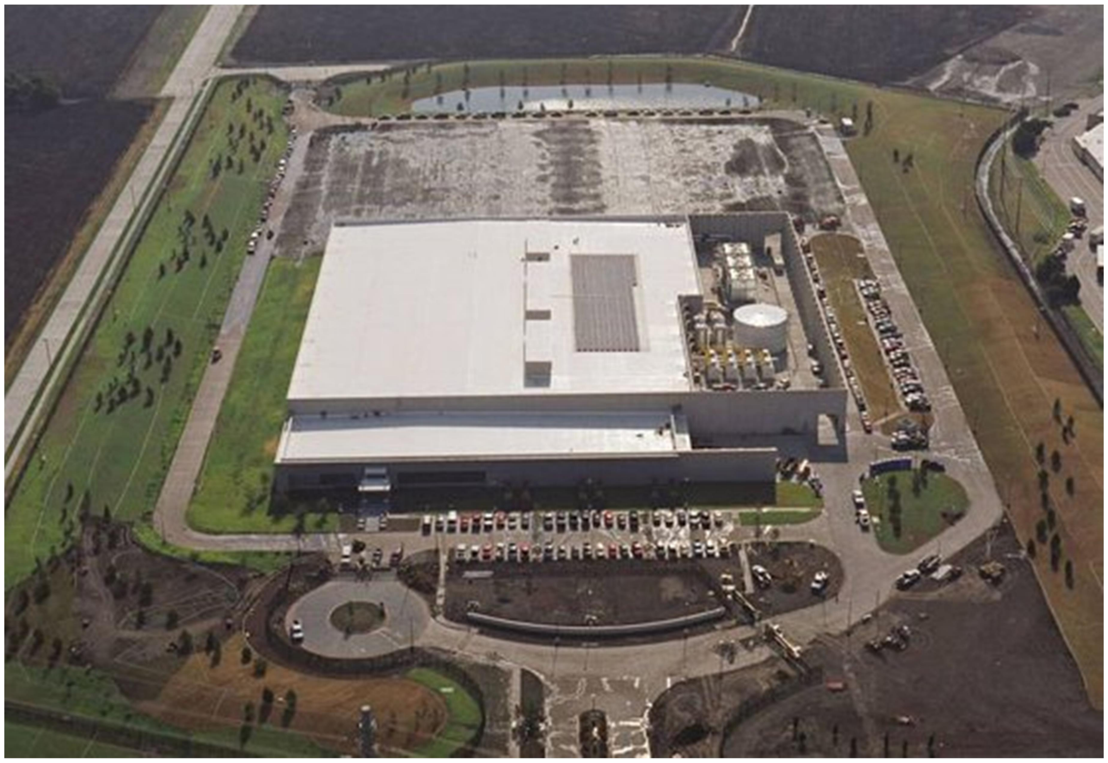


Figure 11: green datacenter (source: cisco.com)

The above picture displays a new green datacenter that Cisco built for his internal IT use. Solar panels can be seen on the roof surface and the datacenter uses non cooled water to regulate the temperature.

Future of datacenter

Given the interest and the amount of money around building and administering datacenters, it is normal that vendors will try to push new features and technologies to increase their shares in a growing market.

The evolution of the datacenter is a mix of new needs that the customers have, a need to obsolete technologies that are outdated and a push from vendors.

In this chapter we are going to see some of the latest evolutions that will be “hot” in the next years to come. Without going too deep, we will talk about the ideas behind them, how they are implemented and how they will change the future. In particular we will take example of Cisco and VMware technologies as Cisco and its partner were the biggest pusher for the standardization of these new protocols and evolution of the datacenter.

Desktop virtualization

Before talking of more network related evolutions, let’s talk of the virtual desktop infrastructure (VDI). VDI [12] is a general idea that will help companies reducing IT costs by stopping physical desktop management to move to a virtualized desktop environment were each employee will have a virtual machine assigned to him/her and it will be able to access it via a thin client or any device having a network interface and capable of speaking Remote Desktop Protocol (RDP) or PC over IP (PCoIP). Thin clients are small machines with a minimal hardware and operating system. The hardware is a cheap small PC with a VGA/DVI output to connect to a monitor, some USB ports to connect keyboard, mouse and other peripherals and at least one network interface. The OS running on those devices is usually a lightweight Linux distribution that implements the PCoIP and RDP protocol plus the essential TCP/IP protocol suite. Once the user connects the thin client to the network, the client gets an IP via DHCP and then connects to a broker (described later) that handles authentication. Once the user is authenticated, the thin client connects to the remote desktop and shows it onto the monitor. The input of the keyboard, mouse and eventually other USB devices attached to the thin client, is redirected to the remote desktop.

VDI is a general idea and is not proprietary of a vendor. Different vendors implement VDI in different ways [13], VMware calls it VMware View and Citrix calls it XenDesktop just to quote the two major players in this field.

We are going to describe briefly what the VMware VDI implementation is.

VMware view is a collection of software that is installed on servers in order to provide connectivity and other features for users connecting via a client to a remote desktop running as a virtual machine on a hypervisor present on a server in a datacenter.

Those servers are used to authenticate (a working Active Directory (AD) infrastructure is necessary), assign VMs to users and if needed, provide additional features like encryption of communication and reduce the used disk space by providing a base image for the VMs and saving only the diffs. On each VM an agent is installed to allow the machine to communicate with the broker. This is used by the broker to poll for availability of the VM, to force a reload of the VM and get statistics.

Here is a picture of a standard VMware View environment:

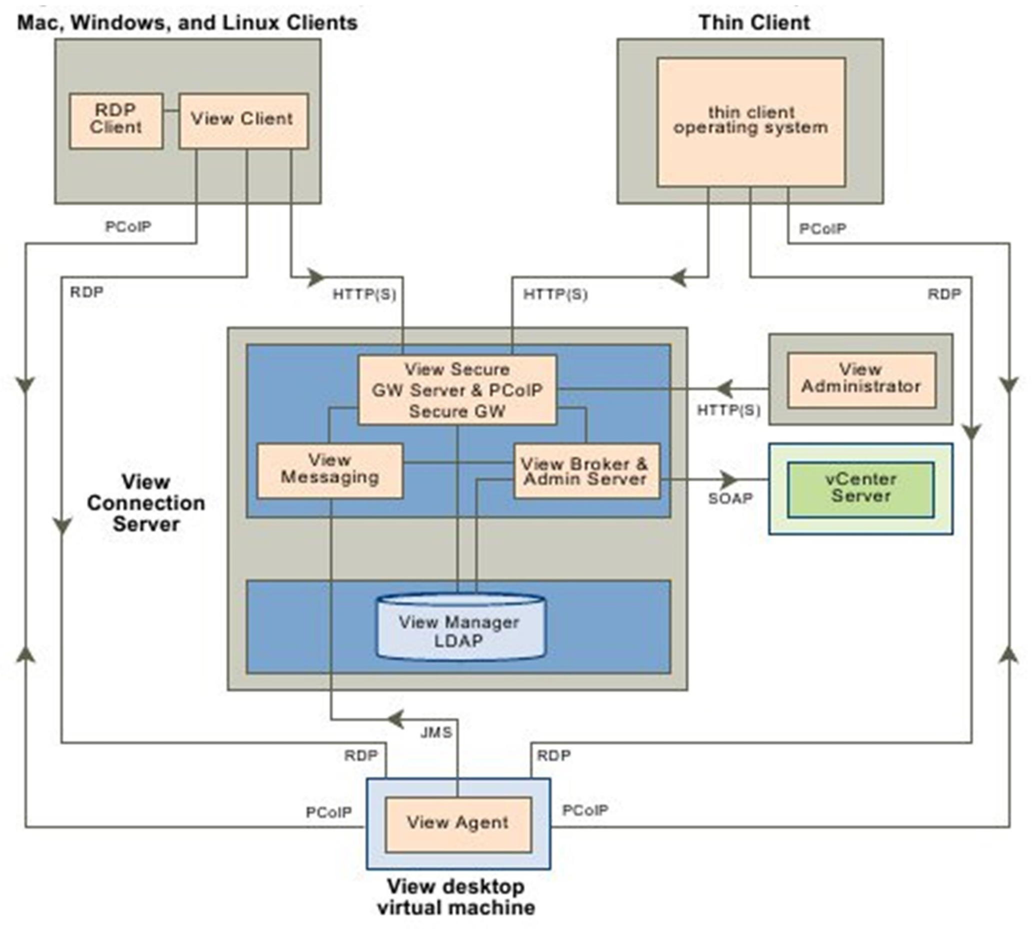


Figure 12: standard VMware View infrastructure (source vmware.com)

In this picture, we can see how the various components interact together and which protocols are used. The arrow indicates the flow of the connections: an arrow pointing from the Thin Client toward the View Agent indicates that the Thin Client initiates a connection toward the Agent. This is important because of how firewalls may be configured as by default they block any connection from the outside network to the inside. Issues often rise with outbound PCoIP connections as PCoIP is a UDP based protocol. As we know, UDP doesn't have any built in congestion control mechanism and it may cause a denial of service (DoS) if the bandwidth utilization would rise a lot. Some firewalls don't allow outbound UDP connections, so to add a new layer of security and allow for congestion control, a Secure Gateway Server is introduced with the scope of being the only contact point from the clients to the VDI. Clients connect to the secure gateway via HTTPS and all traffic from this moment onward (being authentication traffic, RDP or PCoIP) is encapsulated over HTTPS.

The entire infrastructure is configured and operated via the View Admin Server that interacts with the vCenter server via SOAP. vCenter is administering the ESXi servers that host the virtual desktops.

The administrator needs to create a base image that will be a template to deploy the virtual desktops (usually a deployment contains some thousands of virtual machines). This template needs to contain the agent and other common software like antivirus and standard software. Once the template has been created, the administrator populates a pool of desktops by cloning this template. Each machine created this way can be assigned to a user manually or dynamically. When a user connects, the broker instructs vCenter to start the VM assigned to the user and then, as soon as the OS has booted up and the agent can communicate with the broker, the user is allowed to connect to the remote desktop.

If a secure gateway is in the path, all communications from the user will be done with this server via HTTPS and the secure gateway will act as a proxy toward the internal infrastructure (virtual desktops, AD server, broker…).

One of the biggest advantages of VDI is that the security perimeter is completely contained in the datacenter: administrators can block the USB usage, control file copy, force disk encryption and policies to reduce risks. This is one of the motivations that move enterprises in deploying VDI.

TRILL and L2ECMP

In normal layer 2 networks, redundancy is achieved by having multiple links connected between the network elements.

Each switch in the network is paired with another one that acts as a standby switch that will come into action if the primary switch experiences a failure.

The traditional networks are usually divided in 3 layers:

- Access
- Aggregation (also called distribution)
- Core

The access layer is the layer made of switches where the hosts are connected. These switches then connect to the aggregation layer via usually 2 uplinks (one to each aggregation switch). Connections to the hosts are usually not redundant. The access layer implements also some host related features like DHCP snooping, MAC based access control list, ARP inspection.

The aggregation layer connects to different access switches and implements services like traffic filtering, load balancing and gathering of statistics. These services are usually achieved by using service modules plugged in the switches or standalone appliances. The aggregation layers switches are connected to the core via layer 2 uplinks (campus networks) or layer 3 uplinks (datacenters). As in the access layer, each uplink connects to a different core switch in the redundant pair.

The core switch is the simplest layer of the three and it is used to pass huge amounts of data between the different aggregation layers. The core layer needs to be built with scalability and fast convergence time in mind.

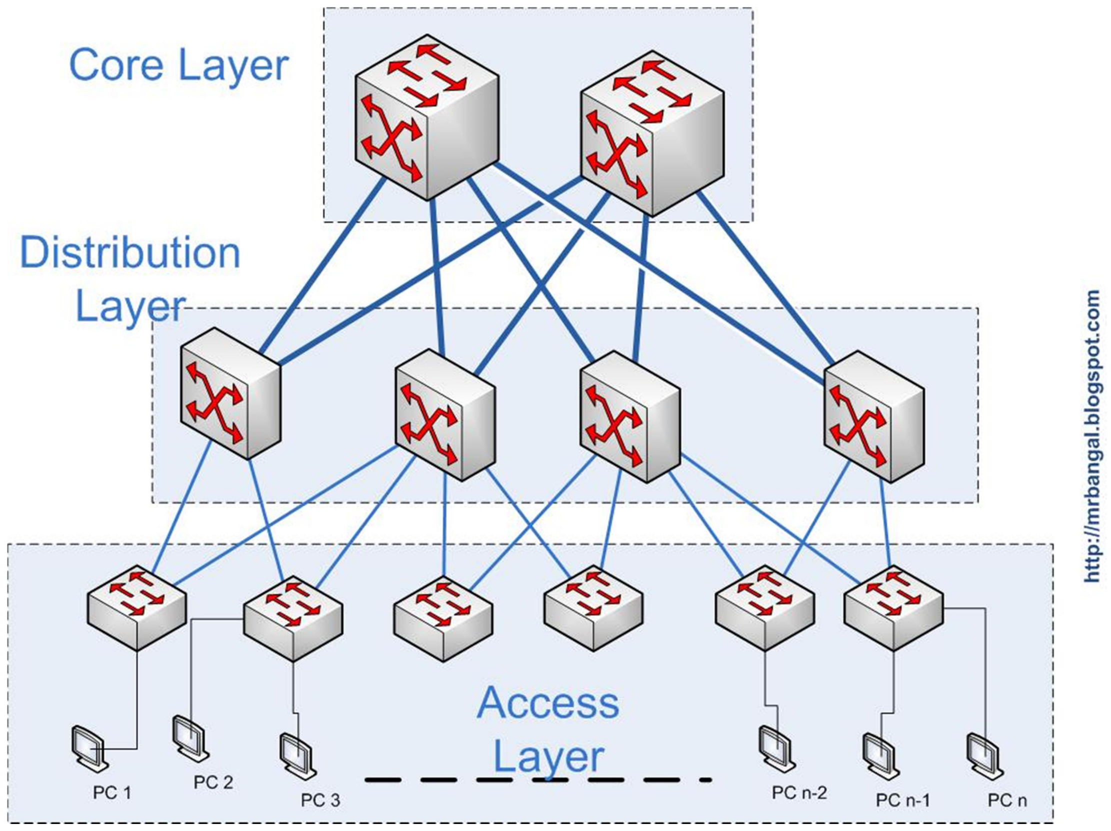


Figure 13: typical layer 2 datacenter design (source: mrbangal.blogspot.com)

As of today, the Layer 2 redundancy has always been achieved by using the spanning tree (STP) protocol. STP has been designed to create a tree in the network by eliminating redundant links that would create a loop topology in order to prevent broadcast or unknown (flooded) unicast to loop endlessly in the network. The loops are removed by blocking a link and this operation transforms the network into a tree. This link would become active again in case the primary link would go down.

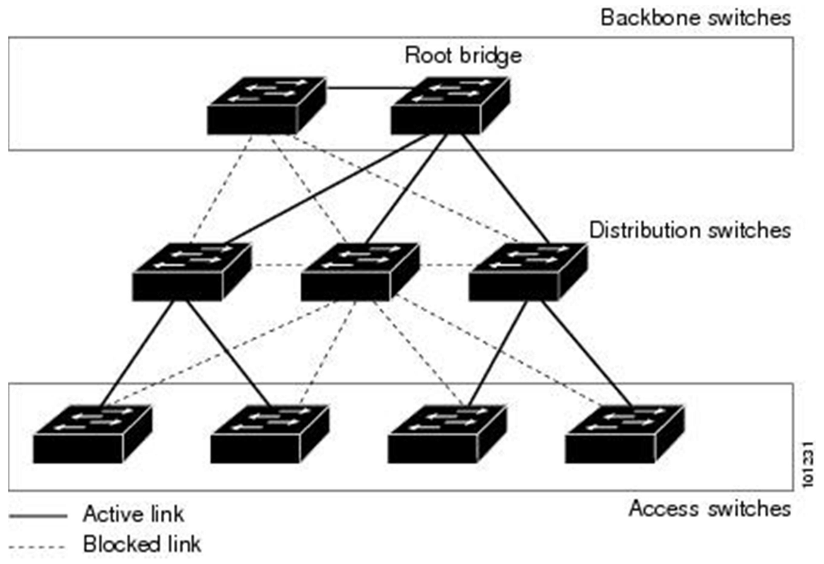


Figure 14: effect of spanning tree blocking redundant links (source: cisco.com)

STP is a protocol that has been engineered in 1985 and while faster versions have been standardized in 1998 and 2004, the protocol still has various limitations:

- STP blocks redundant links, effectively limiting the available bandwidth between two switches to a single logical link (being that a single physical link or an aggregation of links like a port channel)
- STP is a protocol that is not based on negotiation between peers. STP assumes that if no BPDUs are received in a certain amount of time from a port, a host is connected there and the link is moved to forwarding. This causes the STP to converge slowly to changes.
- STP has scalability issues, in the basic version there is no vlan support, in per vlan STP (PVSTP) a bridge PDU (BPDU) has to be sent periodically per each vlan, on top each switch needs to keep information in memory and use CPU cycles to compute the spanning tree on each active vlan. Multiple instance spanning trees (MST) can be used by grouping a pair of vlans to a single spanning tree instance, but this protocol is difficult to configure and to maintain.
- STP is not very stable, loss of BPDU can cause a blocking link to transition to forwarding and cause a loop. Misconfigurations could create root bridge flaps and trigger network wide re-convergences.

In a world where 10 GE is the normal speed for access ports and core interfaces can reach 40 or 100 GE, 3 seconds before a port would to go in forwarding state are not acceptable. Also it is not tolerable anymore to have unutilized links because of loop avoidance: all links should participate to increase the total available bandwidth at all time and connections must be load balanced among all links.

In the latest years, IETF worked with the major networking vendors to engineer the protocol that will substitute STP in the next few years. This protocol is called TRILL [14] [15] [16].

In this thesis I'm not going to discuss TRILL in detail because the standardization of the protocol has not yet finished and there is no commercial product able to run it, but I'm going to talk about the Cisco version of TRILL that is called FabricPath [17]. The major difference between the two is that TRILL supports shared links, while FabricPath assumes that all links are point to point (as actually happens in datacenters based on 10 GE); this has implications on the protocol header.

TRILL and FabricPath are born with the idea of transforming a layer 2 network into a fabric. A fabric can be seen as a cloud where host devices are connected. Those devices are configured with a default gateway used for layer 3 communication, but as soon as the traffic enters the cloud, it will be the fabric to decide which device routes the traffic and when (it could be the first hop or not). A fabric also doesn't require a particular design; a layer 2 network may be inefficient as STP may not always select the best path for forwarding the traffic, while TRILL and FabricPath make sure that always the optimal fastest path is always chosen. Fabric Path and TRILL also allows for the utilization of up to 16 equal cost logical links, so up to 16 logical links between two destinations can be used. Let's remember that a logical link can be a port channel or a single link, TRILL and FabricPath currently allows for the utilization of maximum 16 logical links each made of 16 10 GE links, for a total of 2.56 Tbps bandwidth between two switches.

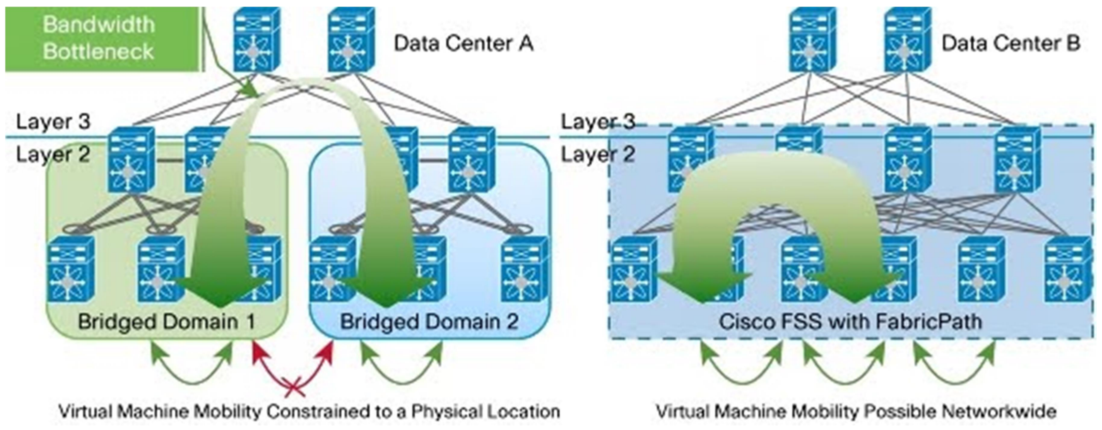

Bandwidth Bottleneck
Data Center A
Data Center B
Layer 3
Layer 2
Layer 3
Layer 2
Bridged Domain 1
Bridged Domain 2
Cisco FSS with FabricPath
Virtual Machine Mobility Constrained to a Physical Location
Virtual Machine Mobility Possible Networkwide

Figure 15: moving from STP to FabricPath and layer 2 extension

Going back to the fabric/cloud and layer 3 routing, Fabric Path behaves exactly like a service cloud where the layer 3 routing is the service offered by the infrastructure and the traffic is the client. The routing is distributed by having multiple devices that are able to perform this functionality and when the traffic enters the network needing layer 3 routing, the first device encountered that offers this functionality will offer it. The client doesn't know which device does it or how many devices are capable of offering this service; it just knows that he can rely on the network to get the most optimized and reliable delivery.

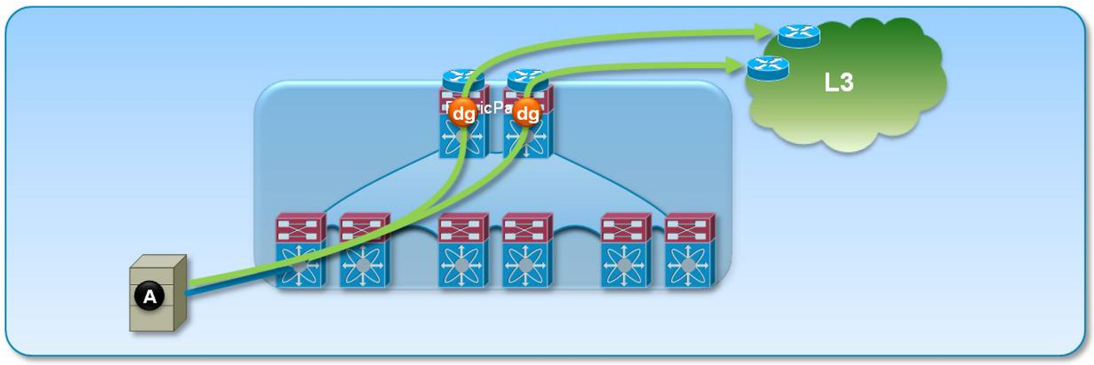


Figure 16: example of multiple default gateway on a Fabric Path enabled network (source: cisco.com)

This could bring to a situation in which each switch in the network is a default gateway by using virtual switched interfaces (SVI) with VRRP/ HSRP/GLBP or IPv6 anycast addresses.

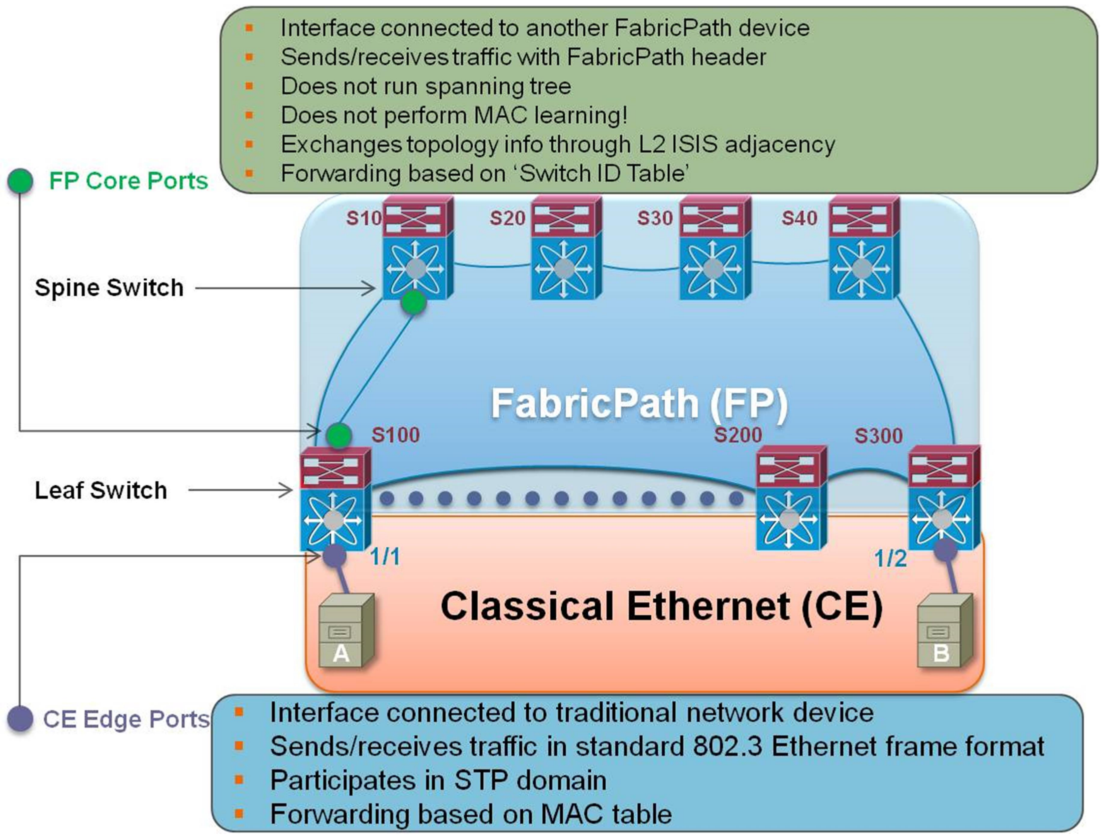
Instead of STP, Fabric Path uses IS-IS to distribute the MAC addresses and create a loop free topology. IS-IS has been chosen because:

- It is an industry standard
- It is not dependent on IP addresses on the interfaces (while OSPF and RIP are)
- It is easy to extend: the protocol is built upon a type/length/value (TLV) format. Cisco had just to define a TLV to transport MAC addresses and now IS-IS can distribute MAC address tables
- Provides shortest path routing, it finds the optimal path to destination
- Fast re-converge

As in the datacenter we usually see virtual machines running on servers and each one has at least one NIC, the number of MAC addresses that are present in the network is way bigger than a normal campus network and it could create overflow of the CAM tables of the switches and unicast flooding. To solve this issue a new feature has been introduced in FabricPath: conversational learning.

We know that a switch learns the MAC via backward learning: when a packet with a unicast source MAC address is received on a port, the switch adds an entry in its CAM table indicating that the MAC is reachable through that port. In FabricPath, MAC addresses are learned only if the destination MAC of the incoming packet is unicast and that destination MAC has been learned on one of the edge interfaces (an edge interface is an interface where a device that doesn't speak Fabric Path is connected like a normal Ethernet switch or a host).

Conversational learning has also another advantage: it enables the manufacturer to build cheaper and faster systems on chip (SoC) containing the full logic to perform forwarding decisions rather than having to engineer multiple ASICs shared by all ports on a linecard with complex arbitration protocols and synchronization. SoC are shared usually among 1-2 interfaces.



Interface connected to another FabricPath device
Sends/receives traffic with FabricPath header
Does not run spanning tree
Does not perform MAC learning!
Exchanges topology info through L2 ISIS adjacency
Forwarding based on 'Switch ID Table'
FP Core Ports
Spine Switch
S10
S20
S30
S40
FabricPath (FP)
S100
S200
S300
Leaf Switch
1/1
1/2
Classical Ethernet (CE)
A
B
CE Edge Ports
Interface connected to traditional network device
Sends/receives traffic in standard 802.3 Ethernet frame format
Participates in STP domain
Forwarding based on MAC table

Figure 17: FabricPath terminology (source: cisco.com)

Similarly to how STP supports multiple trees; one per vlan in case of PVSTP or one per group of vlans as in MST, FabricPath supports multiple topologies. By default all links are part of topology 0, but groups of vlans can be grouped to a separate topology. This could be useful for vlan pruning and traffic engineering. As of now, only topology 0 is supported.

Forwarding in FabricPath is a mix of layer 2 and layer 3 forwarding. For leaf switches, when a packet is received from the FabricPath network, a lookup in the CAM table is done. If the destination MAC has been learned on a classic Ethernet port, the packet is forwarded like normal Ethernet.

When a packet is received from a classic Ethernet port and the destination MAC is not directly connected but known, the result of a lookup in the CAM table will be the switch ID of a FabricPath switch (instead of an interface name like in normal Ethernet). This switch ID will be the switch where the packet has to be forwarded. Once this lookup has been done, a second lookup will start. This second lookup will not be done in the CAM table, but in the routing table built using IS-IS. This operation will tell the interface the switch needs to use to reach the next hop. The leaf switch will encapsulate the original Ethernet packet in the FabricPath header. This header (discussed later) will contain the destination leaf switch where the destination MAC is attached.

If the lookup for the next hop switch returns multiple interfaces (as in case of equal cost multipathing), the switch will choose one of them based on a hashing algorithm.

When the destination leaf receives the packet encapsulated in FabricPath, it will check if the destination MAC has been learned on local interfaces. In this case it will decapsulate the packet and forward it on the edge port, else the lookup result will be a switch ID that will be then looked up again in the RIB and forwarded to the next switch.

Now that we have some more information on how FabricPath works, we can start checking the header structure:

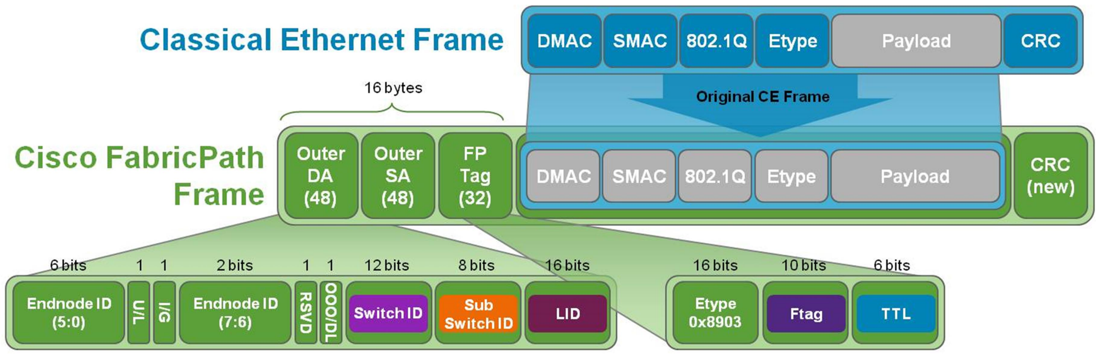


Figure 18: FabricPath header (source: cisco.com)

The first thing we notice is that we encapsulate the native Ethernet frame without changes except the CRC that is recalculated to detect errors in the FabricPath header as well, not only on the original Ethernet frame header.

The second is that the header is composed of a destination address (outer DA) and source address (outer SA) that have the same size as a MAC address (48 bits). The FP Tag is the same size of the 802.1Q TAG; this makes the FabricPath header the same size as the 802.1Q header.

Each source/destination address is comprised of subfields:

- The Endnode ID is a field not utilized at the moment; Cisco keeps this field for future utilization.
- The U/L field is the same as in Ethernet, it indicates if the address is globally unique or locally administered.
- The I/G field is the same as in Ethernet and tells if the address is unicast or multicast.
- The OOO/DL is a field not used at the moment and reserved for future utilization (possibly per packet load sharing when ECMP paths are available).
- The switch ID is an unique identifier of a switch in the fabric
- The subswitch ID is an identifier used when physical switches are virtualized. Each subswitch ID identifies uniquely a virtual switch within the physical switch
- The LID (local ID) is a locally significant value for the switch that destination switch will use to avoid looking up the destination MAC in the CAM. The LID will contain the index of the interface where the MAC has been learned.

The FabricPath tag instead contains the following fields:

- An ethertype of value 0x8903
- And FTAG indicating which topology or tree has to be used for forwarding
- A TTL (time to live) field that as in an IP network is an integer value that is decremented by 1 at each hop. The switch discards the frame if the TTL value is 0. This is used to limit the impact of loops in the network.

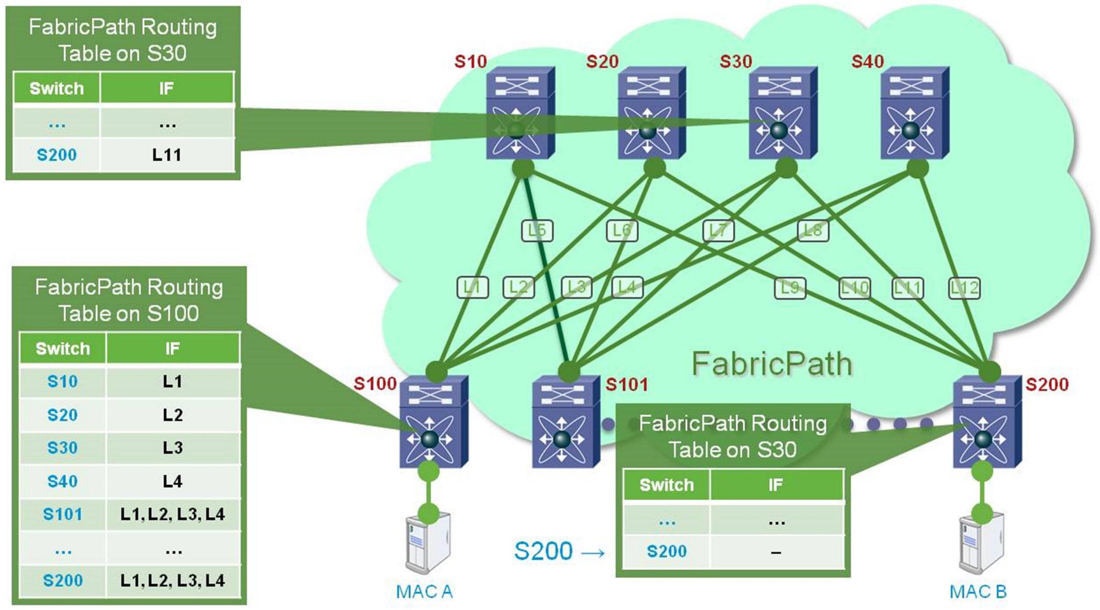
As already anticipated, FabricPath employs multiple trees and topologies for forwarding: trees are loop free and are used to forward broadcast and multicast while topologies are used to forward unicast and may contain loops.

To better understand FabricPath; let's have a look into what happens in the fabric to send a unicast packet from a host to another.

These are the premises:

1) The fabric already built topologies (unicast, possibly with loops) and trees (multi destination, loop free)
2) Because of 1) each leaf and spine tree knows the next hop to reach any other FabricPath enabled switch in the fabric
3) The leaf switches have their MAC address table empty
4) Trees and topology carry all vlans; there is no binding of vlans to topologies/trees nor pruning.

Here is an image of the topology in question with FabricPath routing table already built:



FabricPath Routing Table on S30

Switch	IF
...	...
S200	L11

FabricPath Routing Table on S100

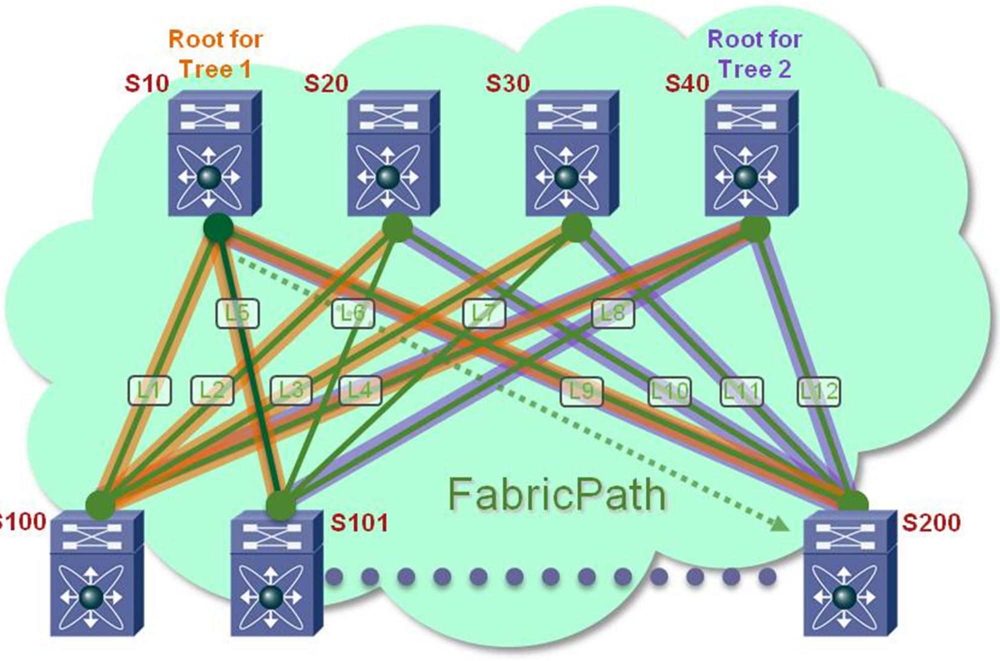
Switch	IF
S10	L1
S20	L2
S30	L3
S40	L4
S101	L1, L2, L3, L4
...	...
S200	L1, L2, L3, L4

FabricPath Routing Table on S30

Switch	IF
...	...
S200	–

Let's assume that host A wants to send a packet to host B and both hosts are in the same IP network and vlan. Host A doesn't know the MAC address of host B, thus it needs to ARP to resolve the MAC address from the IP address.

1) Host A sends an ARP request upstream. ARP request is a broadcast packet addressed to all network elements, the destination MAC address is FF:FF:FF:FF:FF:FF, the source MAC address is host A MAC address
2) The packet is received by S100. S100 is a FabricPath leaf switch so it runs both FabricPath and classic Ethernet (STP in all its versions). The ARP request is received on a classical Ethernet port, and then S100 learns the source MAC of host A on that port, like normal Ethernet.
3) S100 sees that the packet is a broadcast packet, it needs then to forward it out of all its interfaces that are in forwarding state for the vlan where the ARP packet was received from (except the source interface). S100 forwards the packet out of all classical Ethernet interfaces and FabricPath. When forwarding a packet out of FabricPath interfaces, the switch needs to decide which topology or tree to use. As this packet is a multi-destination packet (broadcast) a topology cannot be used because it may contain loops so the switch chooses a multi-destination tree. In the current implementation, 2 trees are supported for load sharing reasons. This is what the multi-destination trees look like in the example topology:

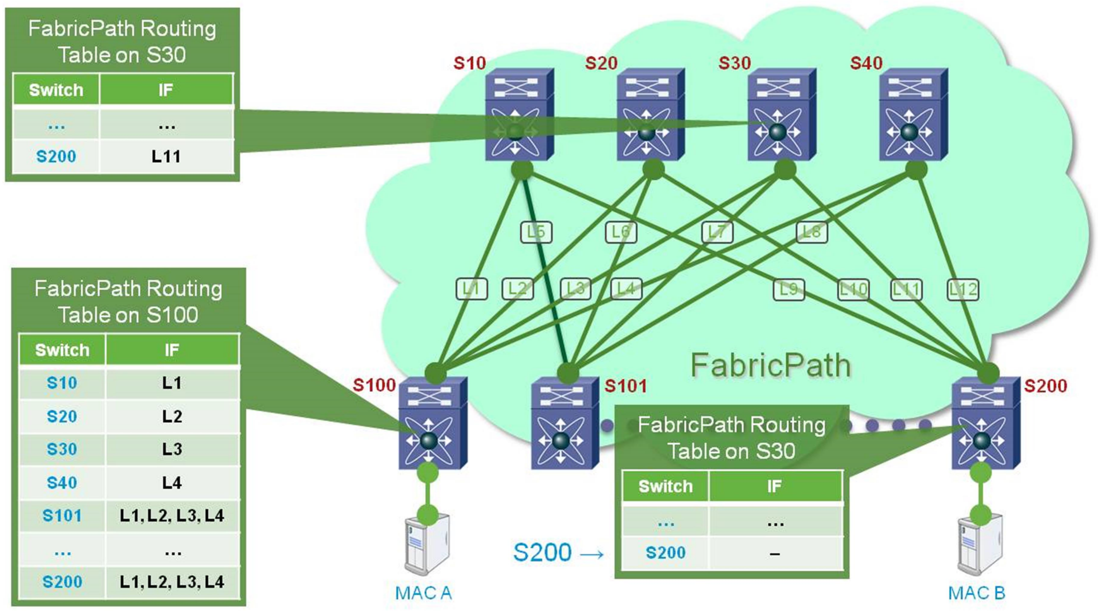


The switch will choose logical tree 1 for broadcast. Tree is used to load balance multicast together with tree 1. Darker orange links are in common between tree 1 and 2.

4) S100 sends the packet out of the FabricPath interface L1 toward S10. To do so, S100 encapsulates the Ethernet frame into the FabricPath header. The outer destination address is filled in with FF:FF:FF:FF:FF:FF. The index of the tree that has been chosen to forward the packet is indicated in the FTAG field, so that all other switches will choose the

same tree. The outer source address will be the Switch ID/Subswitch ID/LID belonging to S100.

5) S10 receives the packet and forwards it out of all the FabricPath interfaces belonging to the tree indicated in the FTAG except on the interface it received it from.
6) All the other switches receiving the packet flood it out on the FabricPath tree and in case of leaf switches, out of the classical Ethernet interfaces in forwarding for the vlan of the packet.
7) All leaf switches will not learn the MAC address of A. This happens because the MAC address of B is unknown and thus no unicast conversation has been initiated yet.
8) The packet eventually reaches S200 that sends it out of its classical Ethernet interfaces where host B will receive it. As for step 7) S200 will not yet learn the MAC address of host A.
9) The packet reaches host B that then replies with an ARP reply. An ARP reply is a unicast packet sourced from MAC B toward MAC A.
10) The ARP reply reaches S200 on a classical Ethernet interface thus the MAC address of host B is learnt.
11) S200 needs to forward the ARP reply, but the destination MAC (host A) is unknown because it has not been learned as discussed in step 7) and 8).
12) S200 needs to treat this packet as an unknown unicast, thus flooding it to the network. The packet is flooded normally over the classical Ethernet interfaces except the source interface.
13) S200 encapsulates the packet in the FabricPath header and chooses the well-known MAC address 01:0F:FF:C1:01:C0. A tree is chosen for forwarding (tree 1 again because this packet is not multicast, but unknown unicast). The outer source address will be the Switch ID/Subswitch ID/LID belonging to S200.
14) S200 sends the packet out all interfaces belonging to tree 1. The packet reaches S10 that floods it out tree 1, but the interface it received it from.
15) The packet reaches S100 that is a leaf switch. S100 will see that the destination MAC address is a MAC address it knows and then decapsulate the packet and sends it out the interface indicated by the MAC address table.
16) At this point, S100 will learn the MAC address of host B because it has been learned over a unicast packet (even if it was an unknown unicast). When learning the MAC address of B, S100 will put the Switch ID/Subswitch ID/LID in the MAC address table and not the interface it was received on. Not that the FTAG is not put into the table because currently we are using a single topology.
17) Let’s imagine now that host A wants to send a TCP SYN packet to host B to initiate a TCP connection. Host A knows the IP and the MAC of host B, so it will build a normal unicast packet and send it towards S100.
18) S100 receives the packet and performs a lookup into its MAC address table. S100 sees that host B MAC address is reachable through FabricPath because the lookup result is a Switch ID/Subswitch ID/LID.
19) S100 will perform a lookup into the FabricPath routing table belonging to topology 1 to find out the best link toward S200.



FabricPath Routing Table on S30

Switch	IF
…	…
S200	L11

FabricPath Routing Table on S100

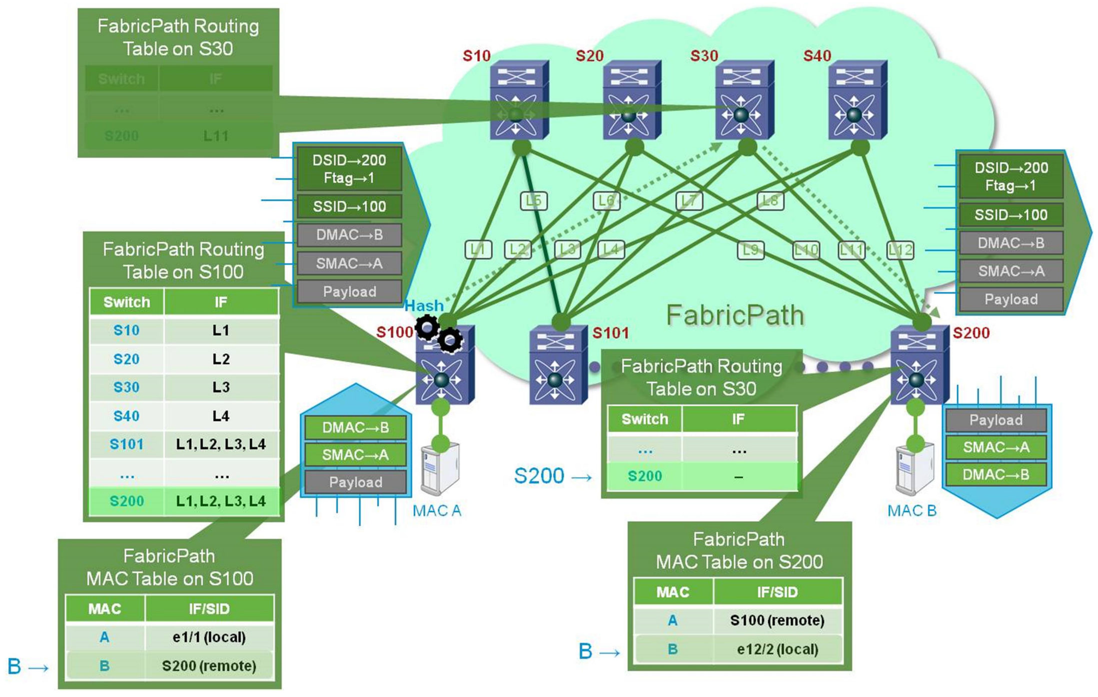
Switch	IF
S10	L1
S20	L2
S30	L3
S40	L4
S101	L1, L2, L3, L4
…	…
S200	L1, L2, L3, L4

FabricPath Routing Table on S30

Switch	IF
…	…
S200	–

20) 4 equal cost paths are available for S200. A hash mechanism is then used to select one. S30 is chosen.
21) Packet is forwarded out of L3. The outer source address will contain Switch ID/Subswitch ID/LID of S100. The FTAG used to forward the packet is 1 (a single topology is available now).
22) S30 receives the packet, it checks if the Switch ID in the outer destination address is itself. The Switch ID is S200, not S30 and then it performs a lookup for S200 and sends the packet out of L1.
23) The packet is received by S200 that checks the outer address Switch ID. The Switch ID is S200, then the packet is decapsulated and a lookup on the MAC address is performed. The lookup result is the classical Ethernet interface where host B is connected.
24) The packet is forwarded toward host B. The MAC address A is learned by using Switch ID/Subswitch ID/LID from S100.

This is what the MAC address table of the network looks like after this operation:



FabricPath Routing Table on S30

Switch	IF
…	…
S200	L11

FabricPath Routing Table on S100

Switch	IF
S10	L1
S20	L2
S30	L3
S40	L4
S101	L1, L2, L3, L4
…	…
S200	L1, L2, L3, L4

FabricPath Routing Table on S30

Switch	IF
…	…
S200	–

FabricPath MAC Table on S100

MAC	IF/SID
A	e1/1 (local)
B	S200 (remote)

FabricPath MAC Table on S200

MAC	IF/SID
A	S100 (remote)
B	e12/2 (local)

Unified Fabric

The term unified fabric is a marketing term that is currently used to identify an infrastructure that is able to transport traffic from both LAN and SAN.

LAN and SAN (as intended in the Fibre Channel world, not in the IP based protocols like iSCSI/CFS/NFS) are very different technologies with almost opposite philosophies behind them: in the LAN environment the main protocol is Ethernet and it is a network where the loss of packets is accepted and in some case necessary. In the SAN environment on the other hand, the network is expected to be lossless as the main protocols operating in this networks (the SCSI protocol encapsulated into the Fibre Channel protocol) do not support retransmission of PDU and a dropped frame usually causes an abort from an Operating System. More details on the SAN protocols will be given in a future chapter.

The Unified Fabric is a solution that allows the SAN traffic to be encapsulated in Ethernet, to create “virtual lanes” (corresponding to one or more specific CoS values) and make them non drop eligible in order to guarantee delivery. The protocol that allows encapsulation of FC traffic into Ethernet is called FCoE and will be discussed later.

As it may be understood, the principal motivation behind the Unified Fabric is that while in a legacy environment a separate LAN and SAN infrastructure will have to be created, in the Unified Fabric the same switches are able to delivery both LAN and SAN traffic and the servers need to be equipped with a single NIC (called a Converged Network Adapter, CNA) instead of an Ethernet NIC and a SAN NIC (called a Host Bus Adapter, HBA).

This created a notable drop in the costs of acquiring and maintaining the network (even more so as often the LAN and SAN team are separate departments in a company, this is called a Silo approach) and the installation. The numbers of cables are lower and cost less, as solutions like Twinax are able to carry both Ethernet and FCoE traffic over copper, while SAN networks use fiber optics with costly transceivers and power costs are almost halved.

The devices that are part of the unified fabric may have the possibility of acting as a gateway between the FCoE traffic and the FC traffic to connect legacy devices like old servers or storage arrays.

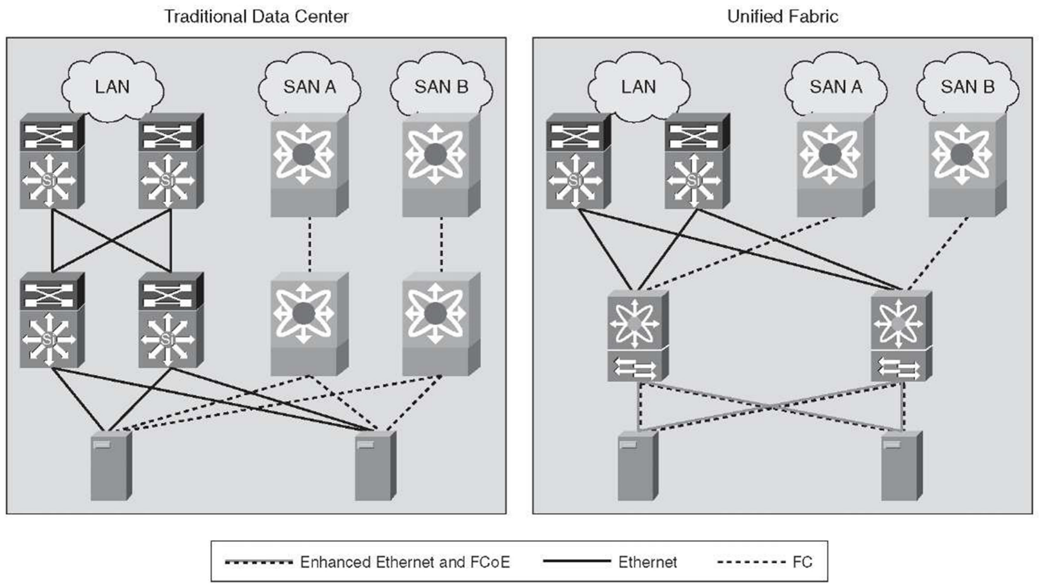


Figure 19: Standard DC vs. Unified Fabric DC (source: what-when-how.com)

To work, the Unified Fabric capability must be built in at least into the Top of Rack switches (TOR) as they connect directly to the servers. As said before, those switches then have the capability of carrying unified traffic (FCoE + Ethernet) to the rest of the fabric or split it back into Ethernet and FC toward legacy networks.

At the moment there is no software implementation available in the field, but in the future the Unified Fabric will move to a Layer 2 Multipath protocol as Fabric Path to benefit from the higher bandwidth.

Design of a modern datacenter

Here we will discuss briefly about the architectures of the Cisco datacenter 3.0 and the general layout of this kind of networks. In future chapters we will go more in details and give a working implementation of a complete datacenter compliant with this architecture.

Cisco’s datacenter 3.0

The passage from a centralized architecture (mainframe) through a decentralized model (multiple machines with a bare-metal Operating System) towards a virtualized environment (multiple servers running a hypervisor and hosting multiple virtual machines) has forced an evolution on the technologies and the designs of the network infrastructure in order to cope with new and unforeseen issue and needs of the business.

A Datacenter 3.0 [18] is similar to a standard campus network composed [19] of an access layer, an aggregation layer and a core layer, but the needs of a datacenter are very different especially when it comes to the services that have to be implemented (in the form of security, accounting, load balancing), the requisite of high availability and the quantity of traffic present in the network at any time.

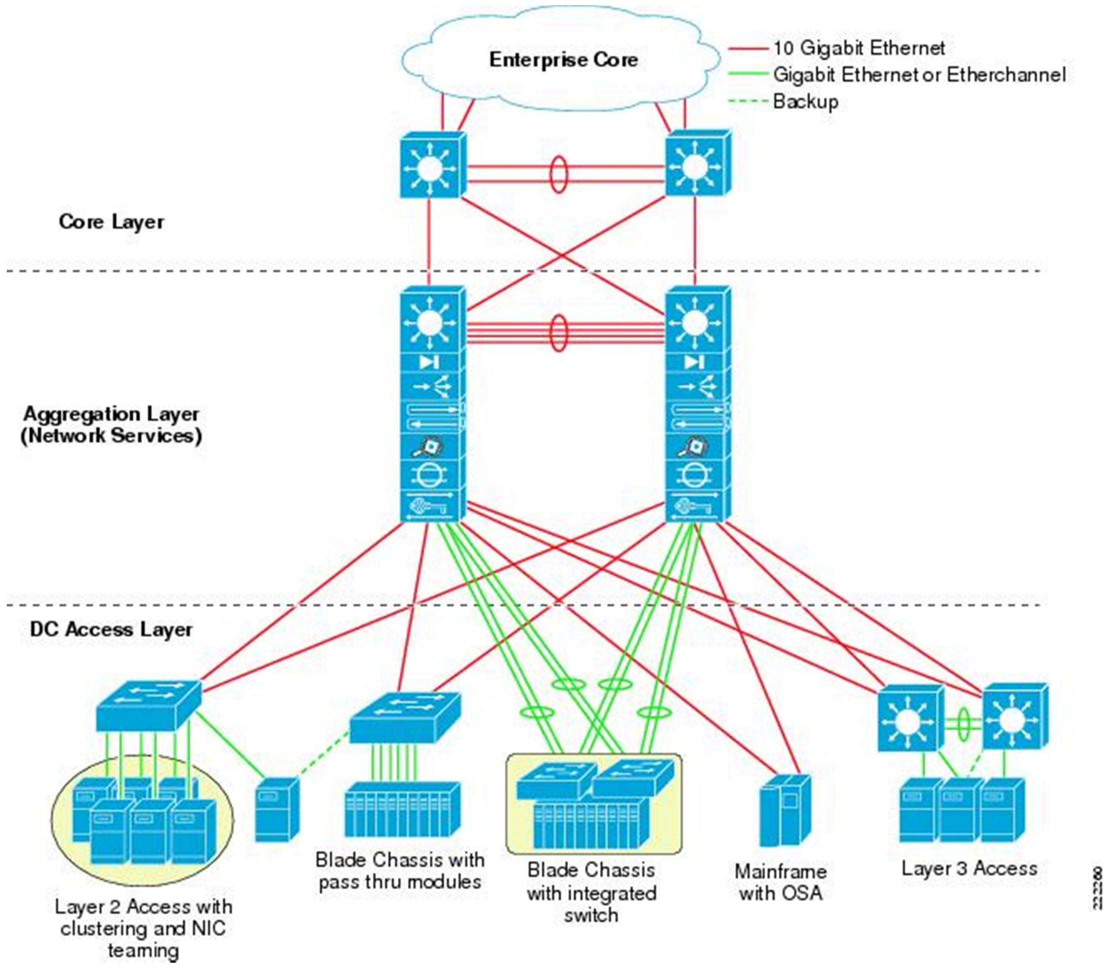

Enterprise Core
10 Gigabit Ethernet
Gigabit Ethernet or Etherchannel
Backup
Core Layer
Aggregation Layer (Network Services)
DC Access Layer
Layer 2 Access with clustering and NIC teaming
Blade Chassis with pass thru modules
Blade Chassis with integrated switch
Mainframe with OSA
Layer 3 Access
222268

Figure 20: simplified representation of DC 3.0 layers (source: cisco.com)

The access layer provides the attachment of the servers to the network. It is usually implemented as a cheap Top of Rack switch (usually 1U dimension). These switches are simple as the features they need to support are very few. The access switches have slow ports toward the servers (1Gbps or 10Gbps) and faster uplink ports connected to the distribution layer (10Gbps or 40Gbps). Depending on the scalability it has to achieve, access switches may implement Switched Virtual Interfaces (SVI) that are used as default gateway for the servers with First Hop Redundancy Protocol (FHRP) like HSRP, VRRP or GLBP. Access switches are always deployed in pairs with interconnection between them to allow dual homed servers to connect to both for maximum redundancy. Routing adjacencies are usually established between access switches and distribution switches to establish routed connectivity between all the pairs of access switches. There is never a connection between multiple pairs of access switches: if traffic needs to flow between servers connected to different access switches, it must go through the aggregation layer.

The access layer should also be low latency to improve performances for the so called “east to west” traffic, so traffic between servers within the datacenter. This kind of traffic has become important recently for Web services where the page requested by the user may be generated using data coming from multiple servers. An example of technology that heavily uses this kind of

approach is Hadoop, HDFS or in general Big Data, where the data is distributed and duplicated in various inexpensive servers and protocols take care of finding the various pieces to be put together to get the whole information.
The access layer is usually running STP towards the servers and the aggregation layer, this means that in standard situations, a link will be forwarding, while the others will be blocking. To avoid this, Cisco invented a technology called Virtual Port Channel (vPC) where two physical switches are configured to appear as a single logical one so that devices connected to it will see it as a single entity instead of two. The two physical switches involved in vPC will use a protocol to synchronize some information (for example STP Bridge ID, LACP ID and MAC address tables) needed for the network to work properly. The links going from the 2 vPC switches to another device will become a bundle (port channel) instead of multiple physical links, so that STP will put it into forwarding state without blocking any link.

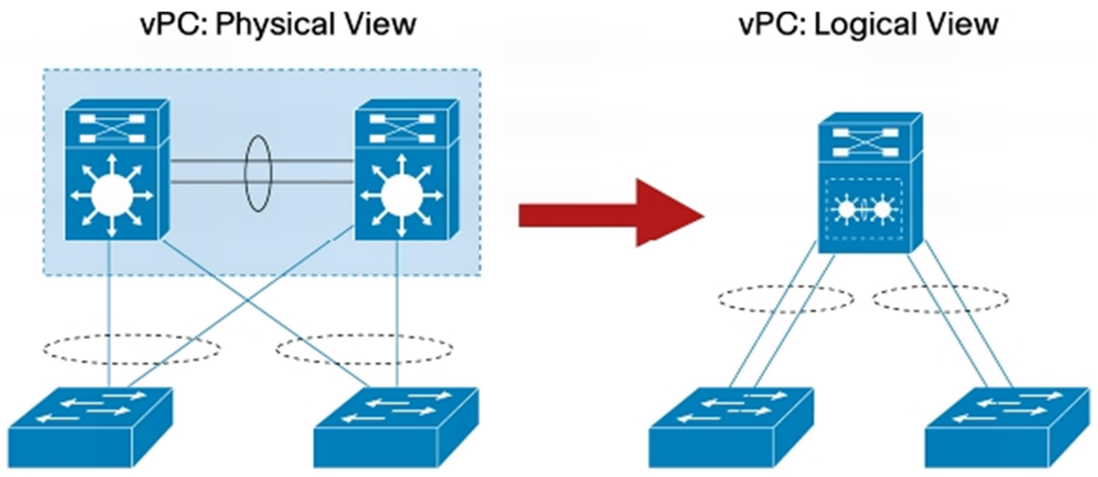


Figure 21: vPC physical vs. logical representation (source: cisco.com)

The aggregation layer is where the uplinks from the access layer are connected to. The switches in this layer are also deployed in pairs for high availability and load balancing. As the name says, the aggregation is where the multitude of sessions in and out the datacenter are grouped. These sessions may need to go through some additional services as firewalling or load balancing before reaching their destination. The services (usually implemented as appliances) are also connected to the aggregation layer. Being connected to the aggregation point, each service is at an equal distance from the servers in the access layer ensuring the same performances and delay for each session. Services are also connected to the aggregation layer because it would be too expensive and cumbersome to have a separate appliance per service per access layer because those appliances are powerful enough to support the traffic from multiple access layers. The aggregation layer is usually a complete layer 3 switch, but due to limitations of some applications or protocols (for example layer 2 extensions between multiple access switches pair), it may be required to extend the layer 2 connectivity towards the access switches. The aggregation layer however is always connected to the core via pure layer 3 connections that are used to establish routing protocol adjacencies. Of all the 3 layers, the aggregation layer is the most complicated one and the devices composing it have to support the highest number of features.

The core layer is the simplest of the layers and it is used to connect the aggregation layers together and give access to the external world via an Internet Service Provider (ISP). The core is a pure layer 3 network implementing features like BGP and OSPF to learn routes from the aggregation layer and exchange them with the ISP. The core however must be very fast and be well designed to not incur in oversubscriptions and drops.

If core and aggregation layers are pure layer 3, a topology involving vPC is not needed, but if the aggregation layer is mixed layer 2 and layer 3, the aggregation switches are usually configured as a vPC pair similarly to what happens in the access layer.

Depending on the size of the network, an alternative topology called collapsed core/aggregation could be implemented. In this topology, as the name says, the core and the aggregation layer are collapsed in one single layer that is used for services and internet connectivity.

In the Cisco vision of the 3.0 Datacenter, we can find 3 layers on the network to be seen as overlapped to the access/distribution and core: the virtualization, the unified fabric and the unified computing.

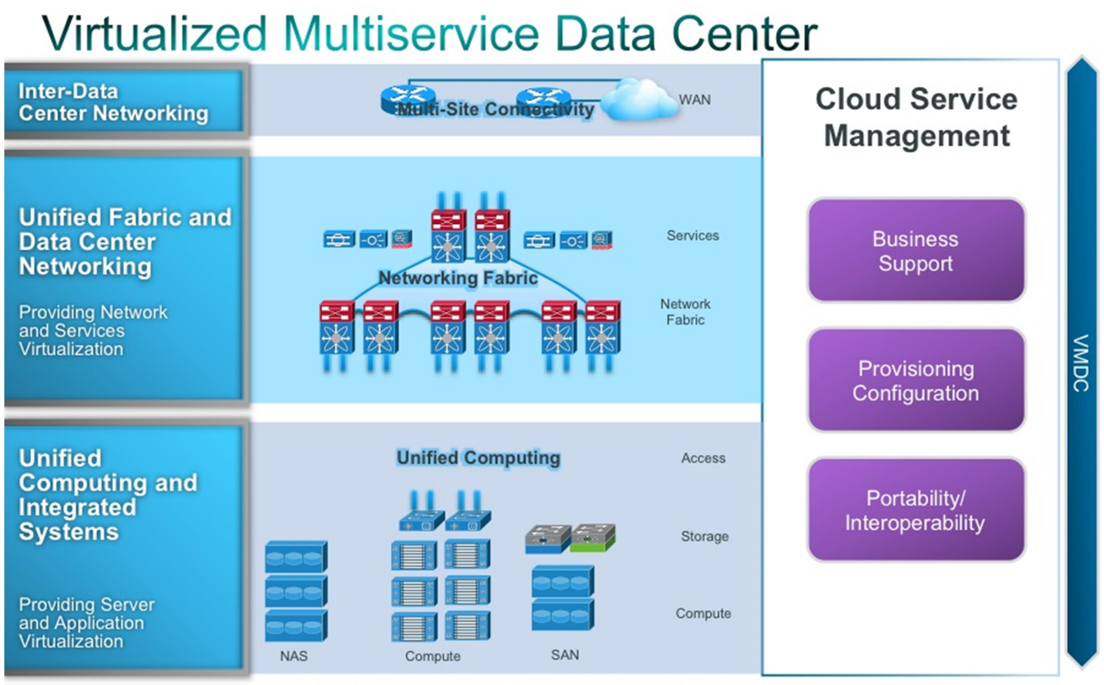

Virtualized Multiservice Data Center
Inter-Data Center Networking
Multi-Site Connectivity
WAN
Cloud Service Management
Unified Fabric and Data Center Networking
Providing Network and Services Virtualization
Networking Fabric
Services
Network Fabric
Business Support
Provisioning Configuration
Portability/ Interoperability
VMDC
Unified Computing and Integrated Systems
Providing Server and Application Virtualization
Unified Computing
Access
Storage
Compute
NAS
Compute
SAN

Figure 22: Virtualized Multiservice DC with FP as transport protocol

The virtualization layer is a layer that is present in the servers (via the hypervisor) and in the network devices as well. It is used to segment and isolate different instances on the network. One of such instances could be a customer renting infrastructure from a provider. You do not want to mix the traffic of different customers in order to maintain privacy and protect their data. Also, a provider may sell infrastructure as a service in the form of switches and not only host virtual machines, so it may happen that customers may have some restriction on routing/IP addressing that may overlap with other customers creating conflicts. To avoid these issues, the virtualization layer offers some technologies to help: VLAN, VSAN, virtual switches, VRF, VDC…

As we know, VLANs are an additional tagging in the Ethernet packet designed to isolate and contain traffic and broadcast packets to improve performances and reliability of the network.

VSANs work similarly to VLAN, but for the FC traffic. In this case the emphasis is on the separation of traffic between main and backup paths (called Fabrics, seen in detail in a later chapter) rather than in restricting broadcast (as FC networks have limited broadcast compared to Ethernet).

Virtual switches are a software switch implemented as a plugin (usually a Linux Kernel module) for a hypervisor Operating System that is able to increase the capability of the hypervisor in network related features, for example improving the QoS customization. Two examples of these virtual switches are the Nexus 1000v [20] and VMWare DVS (built in with ESXi). The virtual switches have also the ability of tapping directly into the hypervisor and learning the MAC/IP addresses of the VMs and their location in order to help with VM mobility.

VDC are a feature of the Nexus 7000 series that allows the creation of up to 8 virtual switches each one with a different configuration, resource limit and processes. Similarly to virtualization, this kind of technology allows splitting a physical device into multiple in order to increase separation and isolation and potentially allocate each virtual device to one of more customers. While very similar to standard virtualization technologies, the VDC virtualization is not achieved through a hypervisor approach, so there are no multiple instances of a base operating system running at once, but processes (like OSPF, STP, BGP…) are spawned multiple time and assigned to a VDC instance.

VRFs are an implementation of multiple routing tables in a router/layer 3 switch. Each table has a descriptor used to differentiate it from the others and it is assigned to a customer. Overlapping routes and IP addresses can be installed in separate VRF and will not conflict. Inter VRF routing and route export can be achieved by means like BGP, MPLS or via Virtual Private Network (VPN). Each logical interface will belong to a VRF and packets arriving on that interface will be forwarded based on that VRF’s routing table.

Differences with a campus network

Datacenter and campus networks are very similar networks when it comes to topology, but with some differences in detailed design.

A campus network can be imagined as the IT infrastructure of a business or organization that is used by the users for their job. The campus network is usually distributed in one of more edifices rather than being concentrated into a single building as a datacenter.

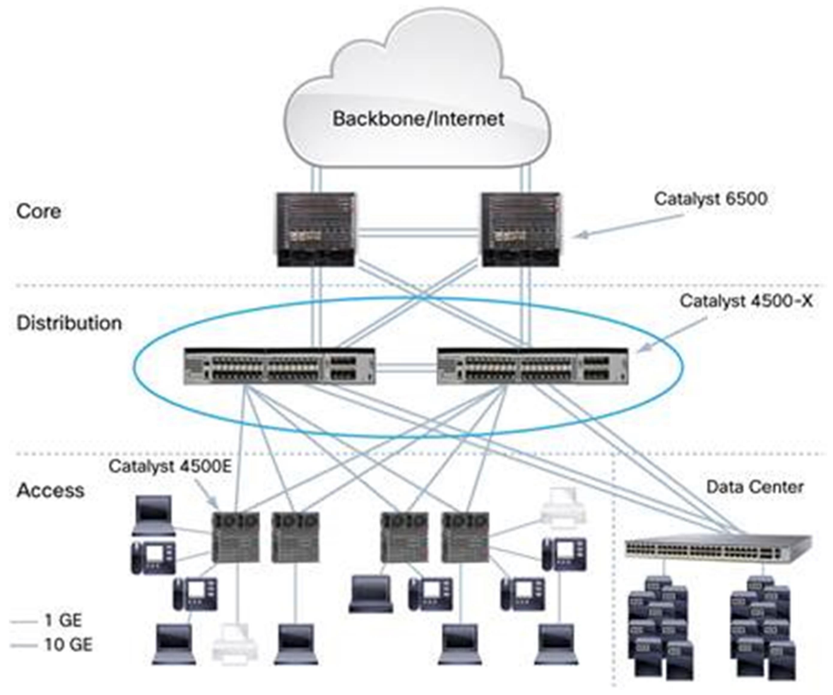


Figure 23: campus network (source: cisco.com)

The campus network has a mix of users' machines (laptops and desktop PC, phones, security cameras…) and servers used for application (email, HTTP and so on).
As in the datacenter networks, servers have redundant connections to the switches while users' machines usually have one cabled connection. This relaxes a bit the requirement on the access layer (for example vPC towards the users' PC is not needed); however the access layer becomes incredibly more complicated due to other requirements, especially security and IP telephony. Access layers need to be able to provide secure access to the users' PC and phones and this often requires the implementation of features like voice vlans, 802.1X, DHCP snooping, dynamic ARP inspection, port security and so on. Some campus networks have wall plugs where users can plug their own device; this is totally opposite of the datacenter philosophy where everything is tightly controlled.
Again regarding security requirements, due to the ability of the user to install software, application monitoring requires particular attention, usually achieved by using security appliances such as IPS/IDS and firewalls that are able to identify potential vulnerabilities.

Users may also connect not directly to the switch ports, but through wireless connections via an access point.
All those features are not needed in the datacenter because the designers and the operators know what is connected to each switch and there are very low amounts of changes (server equipment gets deployed and once installed, it stays there until the end of its useful life span).

A campus network has lower high availability and bandwidth requirement than a datacenter because the average network use of an employee is lower than the one a virtualized server where multiple virtual machines offer services and the traffic out of the physical server is the aggregate of the traffic of the hosted virtual machines: small disruption of network connectivity can be accepted in an office.

On a campus network, layer 2 domains are usually spread across multiple access layers, but do not stretch much away. Workload mobility is limited and usually layer 3 mobility is enough, while on a virtualized environment, the VM that are migrated may need to have layer 2 adjacencies in order for the application to work, this imposes a huge design restriction that can be solved via an increase of the layer 2 domain size (with correlated performance loss due to broadcast) or encapsulation techniques as LISP, NSX and so on, that however tax the network with additional features.

Regarding instead physical connection, the datacenter network offers usually copper for legacy environments, 10/40 Gbps twinax connections for unified fabric and CNA, or fibers (usually between switches or on metro/geographical connections), while campus are still prevalently over wireless or copper (1 Gbps). A particular feature however of campus network is the presence of power over Ethernet connections, where small devices like phones or access points are powered up through the network to allow organizations to save on cabling costs. As the power is provided by the switch, access switches have to be equipped to support this feature and require additional power supplies.

Storage Area Network (SAN)

We are going to discuss briefly about the Storage Area Network. The storage area networks are a special type of network that is used exclusively to transport storage information. The goal of a SAN is to substitute the local storage in a server by concentrating all the disks into a small number of arrays in order to increase security, uptime and performances and reduce maintenance cost.

There are various types of SAN depending on the transport protocol used:

- Fibre Channel (FC) [21]
- Ethernet (iSCSI, NFS, CIFS)

In an operating system, the SCSI protocol is used to communicate with a SCSI hard drive to write and read blocks at a certain address. The SCSI protocol is normally used over a bus physically located within the server where all the SCSI drives are attached. FC and iSCSI allow the SCSI protocol to be encapsulated onto the network to connect to remote drives. This is transparent to the Operating System and so there is no need to modify the code; the remote drive will be seen as a normal drive and the network interface will take care of encapsulating the SCSI PDU into the transport protocol. The biggest limitation of the SCSI protocol is the lack of flow control and retransmission capabilities: being born as a local protocol, the SCSI protocol doesn't account for network error because it is locally arbitrated, so when traversing a potentially congested network, the transport protocol need to take care to either retransmit dropped frames within a short period, or to avoid congestion at all via flow control methods. iSCSI is based on TCP, so it uses TCP retransmission, while FC tries to make the network lossless by using a hop by hop congestion control mechanism called buffer to buffer (B2B) credits. We will go into details of the FC protocol in the next chapter.
The FC network requires an ad-hoc network infrastructure while iSCSI can run on top of a normal Ethernet network.
In order to unify the FC and the Ethernet networks, the FCoE protocol has been invented allowing FC frames to be encapsulated in Ethernet, but reinforcing the transport protocol to allow for FC like hop by hop flow control.

NFS and other Ethernet based protocols instead use TCP/UDP for transport and allow file level access, less performant compared to block level access, but simpler. The operating system however would need modification to be able to access a file as the mapping from file to network path cannot be done by the network interface. These protocols allow for higher level functions are read/write lock, file ownership and permissions.

Why remove the physical disks from the servers and aggregate them into an expensive array? This is done to reduce costs and reduce the security perimeter of the data. By having the data all concentrated into a centralized series of arrays, advanced and expensive caching/backup and redundancies strategies can be applied. Also the lack of physical data into a server or use of a laptop reduces the risks in case of theft: centralized information can be controlled better. Regarding high availability, the disk arrays are usually replicated multiple times in different geographical sites and locally via the use of RAID disks. This greatly reduces the costs because using a RAID 5 or 6 strategy to make an array redundant requires spare disks: parity is striped in 1 or 2 spare disks, if this would have to be used into a server, it would require 1-2 spare disks per server, while on an array it may require 1-2 spare disks per RAID array.

Fibre channel protocol

This chapter will talk about the FC protocol [22] a bit more in detail as it is not widely found in the common network environments, but it is very important as it is used in settings where performances and reliability are very important.

Unlike the ISO/OSI stack, the FC stack is composed of only 5 layers:

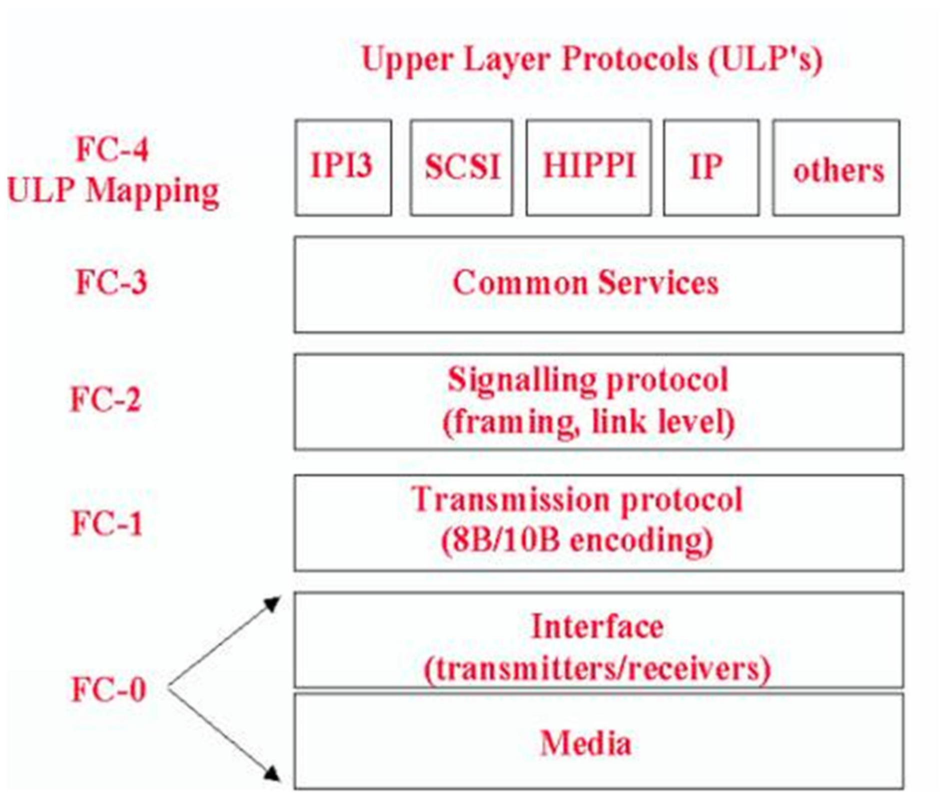


Figure 24: FC protocol layers (source: cisco.com)

The organization presiding FC is not IETF or IEEE, but it is called T11. Documents on FC can be found on the website T11.org. The standards name are numbered with the following schema: FC-

(component, 2 letters)-(version number). Examples of documents name describing the standards are:

- FC-PI-4 – Fibre Channel Physical Interfaces
- FC-FS-3 - Framing and Signaling 2
- FC-LS-2 - Link Services
- FC-GS-6 - Generic Services 6
- FC-SW-5 - Switch Fabric Generation 5
- FC-BB-5 – Fibre Channel Backbone (FCoE, FIP)

Here is a short description of what the FC layers [23] do:

- FC-0 - Signaling rates, cables, connectors, distance capabilities, etc. (FC-FS / FC-LS ordered sets, basic link services, link control protocols, flow control, COS, frame structure…)
- FC-1 defines how characters are encoded/decoded (8b/10b) for transmission
- FC-2 defines how information is transported: frames structure, sequences, exchanges, Login Sessions
- FC-3 is a placeholder for future functions
- FC-4 defines how different protocols are mapped to use Fibre Channel: SCSI, IP, Single byte command code sets (ESCON/FICON) and others

FC is born with very strict ideas on what a network is supposed to do and how. The standard has been implemented with topologies in mind thus the protocol is less flexible than Ethernet. The protocol has been engineered with features needed for the design of those topologies.

These are the possible topologies in FC:

- Point to point: two devices directly connected to each other without any switch in the middle
- Arbitrated loop: all the devices are connected together in a loop. Bandwidth is shared among all devices and a token passing schema is used to decide which device can access the network. Switches and hubs can also participate in a loop topology.
- Switched fabric topology: end hosts connect to a fabric composed of FC switches (similar to what a standard Ethernet network is).

The FC standard also describes a few port types. Similar to an access port vs. a trunk port vs. a layer 3 port, the port type indicates what the port can do and which frames it accepts or drops.
FC port types are:

- N (node) port: used on end host devices to connect to the switch or between end hosts in a point to point topology.
- F (Fabric) port: used on the switches where an N port is attached.
- L (loop) port: generic port belonging to a device attached to an arbitrated loop topology. If the device is an end host, then the port has also N capabilities and is called an NL port.

- E (expansion) port: used for inter switch connection. Can be thought of as a trunk port from the Ethernet world.
- G (generic) port: can operate as E, F or L port via an auto-discovery mechanism.

The FC standard allows for vendor expansion to add functionalities. These port types are Cisco specific and may not be compatible with other vendors:

- FL (fabric loop) port: a switch port connected to a NL port.
- TL (transitive loop) port: used to interface between a FC fabric and a NL port.
- TE (trunk extension): used between switches, allows to carry multiple virtual SAN (VSAN) in a concept very similar to the VLAN in an Ethernet network.
- SD (span destination): port configured on a switch that is used as a destination for a copy of the traffic sourced from another interface. An end host device is connected to these ports and it receives the copy of the traffic. As the SD port can only transmit traffic, this port cannot be used for normal FC communication, but only for monitoring and troubleshooting.
- ST (span tunnel): port configured on a switch that is used as a destination for a copy of the traffic sourced from another interface. In this case, the traffic is first encapsulated in another protocol in order to allow for the end host station capturing the traffic to be remote. The switches treat this encapsulated traffic differently and forward it only over other span ports.
- Fx port: a port that can operate as F or FL.
- B (bridge) port: used to connect special adapters for geographical interconnection.
- Auto mode: discovery of the operational mode. A port in this configuration can operate as F, FL, E or TE.
- TF, TNP (trunking) port: used to trunk multiple VSAN to an end host in normal or NPV mode (discussed later)

Let’s discuss a bit more in detail the FC protocol layers. The FC protocols layers are very different from the Ethernet ones and lots of high level Ethernet/TCP/IP features are implemented in lower FC layers.

We will skip the FC0 as it discusses only which cables and transceivers are allowed, which are the Rx and Tx power needed for a link and the distances for each transceiver/cable pair.

The FC1 defines the information exchanges on a link. Information consists of transmission words. Transmission words are made of 4 transmission characters each.

There are 2 different types of transmission words:

- Ordered sets: special set of characters used for delimiting the frame, primitive signals or primitive sequences. Are not counted towards the content of a frame. Ordered sets are recognized because they start with a special character called K28.5 (only FC allowed character with 5 consecutives 1s)
- Data words: data contained between the start of frame (SOF) and end of frame (EOF).

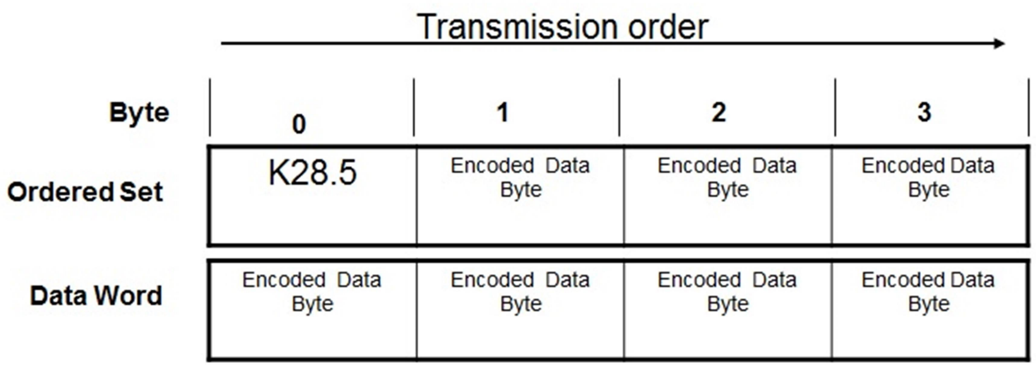


Figure 25: FC ordered sets and data words (source: cisco.com)

The primitive signal indicates that something has happened at the sending port side (port ready to receive a packet, link sync loss, error, closing connection) or that nothing is happening (idle).

The primitive sequence is an ordered set that is transmitted continuously to indicate that a special condition within the sending port is encountered (not operational, offline, link reset…)

The FC2 layer defines the frame header and the device port identifiers among other functions.

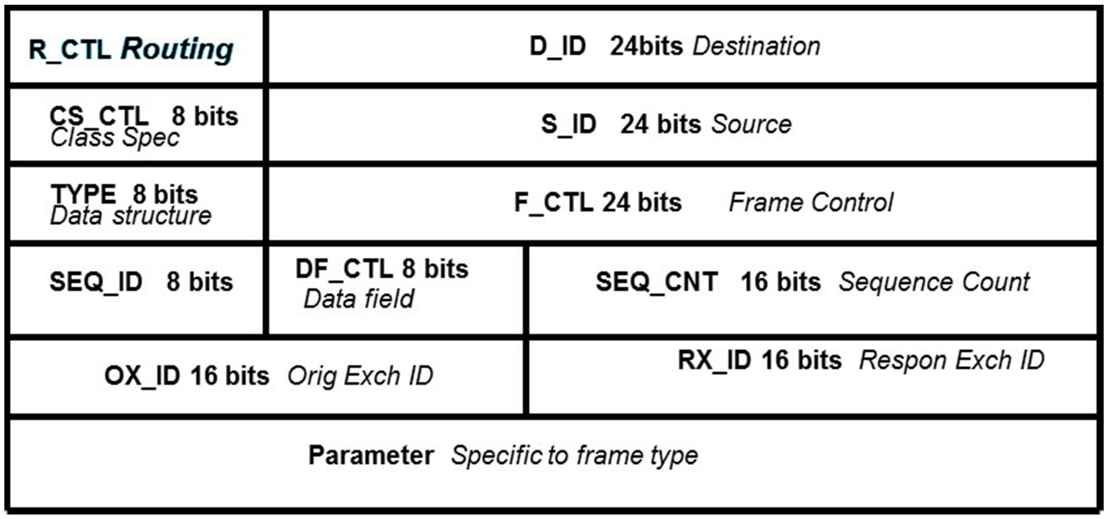

R_CTL Routing
D_ID 24bits Destination
CS_CTL 8 bits Class Spec
S_ID 24 bits Source
TYPE 8 bits Data structure
F_CTL 24 bits Frame Control
SEQ_ID 8 bits
DF_CTL 8 bits Data field
SEQ_CNT 16 bits Sequence Count
OX_ID 16 bits Orig Exch ID
RX_ID 16 bits Respon Exch ID
Parameter Specific to frame type

Figure 26: FC header fields (source: cisco.com)

This is an overview of the fields:

- R_CTL (routing control): identifies if the frame is a link control frame or a data frame.
- S_ID (source ID), D_ID (destination ID): port address source and destination of the frame. Port addresses will be discussed later.
- Type: upper layer protocol code.

- F_CTL (frame control): used to control the flow of the information. Indicates if the source of the packet is the originator or the target of the flow, if it is the first message of an exchange or not, if there is an abort situation and other functions.
- SEQ_ID (sequence identifier): if an upper layer protocol performs an exchange that requires more than one frame, the sequence identifier is used to identify each sequence. All frames belonging to the same exchange will have the same SEQ_ID.
- SEQ_CNT (sequence counter): identifies the frame number of a multiple frames exchange.
- OX_ID (originator exchange ID), RX_ID (responder exchange ID): used to identify multiple sequences belonging to the same upper layer protocol (similar to TCP/UDP ports).
- Parameters: additional information depending on the specific frame type (link control, upper layer protocols…).

In the FC world, each port has a Fabric Address that is a 24 bit identifier. The identifiers are composed in the following way:

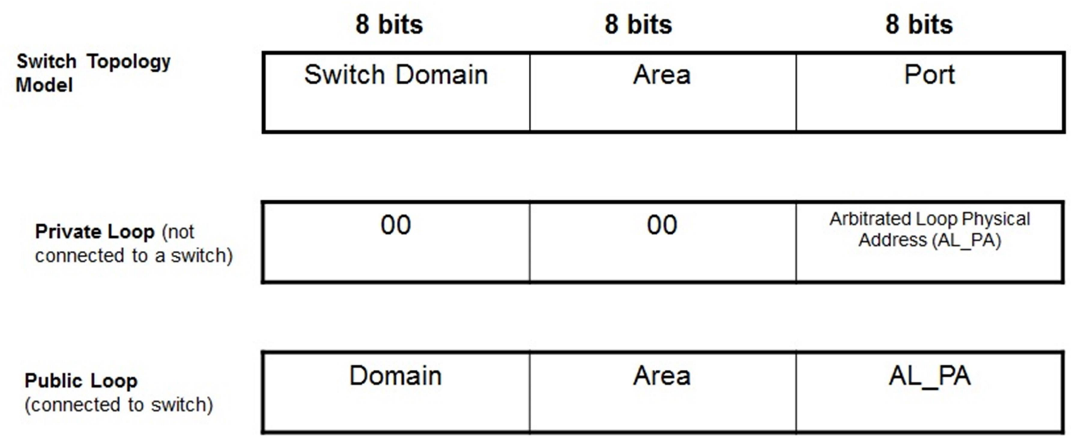

8 bits
8 bits
8 bits
Switch Topology Model
Switch Domain
Area
Port
Private Loop (not connected to a switch)
00
00
Arbitrated Loop Physical Address (AL_PA)
Public Loop (connected to switch)
Domain
Area
AL_PA

Figure 27: port ID schema (source: cisco.com)

We are going to consider only the Switch Topology Model as the Loop topologies are not used in the field anymore.

The Switch Domain is a unique identifier assigned to the switch itself. Each end host connected to the switch will receive a port address (with a mechanism similar to DHCP) that will have the switch's Domain ID. There are 239 available switch domain IDs.

The Area is used to identify a group of ports. There are 256 areas per switch domain ID.

The Port is used to identify the singular attached N or NL port. There are 256 ports usable per Area ID.

By analyzing the port addressing schema, it is possible to see a huge scalability limitation of the FC type networks: each network can be composed of up to 239 switches. This limitation can be lifted by using a feature called NPV/NPIV that acts by having a switch (the NPV switch) to present itself as a host to the uplink switches (the NPIV switches). The NPIV switch will then assign the same Switch Domain ID to all the NPV switches that are connected to it and the NPV switch will somehow act as a NAT gateway towards the end hosts connected to it.

When the switches are initially connected together they exchange information by sending packets from and to a well-known address of 0xFFFFFD called domain controller. This address is used to configure initially the fabric and assign the Switch Domain to each switch and verify that there are no duplicated Switch Domains (Switch Domains can also be statically configured).
Once each switch in the fabric has a Switch Domain, the switches build a topology of the network and distribute the Switch Domain information via a routing protocol called Fabric Shortest Path First (FSPF) that is identical to a single area OSPF scenario, but it is adapted to run over FC (as OSPF is dependent on IP to transport the link state advertisement). As a switch will assign addresses to the N ports that contain its Switch Domain in the first byte of the Fabric Address, the switches in the fabric don't need to have every N port address in the FSPF database, but need to know only to reach the switch that owns that Switch Domain. This is somewhat similar to routing an IP subnet versus routing multiple /32 addresses.

To be able to start sending data, a port needs to login first. The login procedure is a complex procedure composed of various states.

First a port performs a Fabric Login (FLOGI) to exchange the port parameters (port type, speed, establishment of a link…). This causes the port to go into "up" state, but traffic still cannot pass. Also E ports between switches perform this login operation.
During this operation, each port is identified with a port identifier (domain ID + area ID + port ID). A port however has also an equivalent of a MAC address called Port World Wide Name (PWWN) that is sent via the negotiation packets and it is used for the switch to create a mapping between a Port identifier and a physical port. Each device has also a Node World Wide Name (NWWN) that is used to correlate a station (as it can be multihomed) with its ports. NWWN and PWWN are assigned at manufacturing time and are an 8 bytes long identifier.
To perform a FLOGI, the port sends frames to the well-known Fabric address 0xFFFFFE. This address is used for the device to discover the fabric, negotiate service capabilities and receive its Fabric address.
Once a device is logged in and registered into the Fabric, it will perform a Port Login (PLOGI) toward a well-known address of 0xFFFFFC (Fabric name server) and register its capabilities with the fabric, for example to indicate if the port belongs to a server (called initiator) or a storage array (called target).
The device will query the name server to receive information on which other N ports are present in the network and which are the capabilities of those ports.
If a port finds another port offering the service it needs (example a server looking for a storage array port), it will perform another Port Login (PLOGI) towards the Fabric address of that port.
After the PLOGI exchange, the upper layer protocol functions can start.
The Fabric switches will filter the list of the devices a port receives from a name server query based on the zoning information. The zoning information contains a list of which devices are allowed to connect together and it is very similar to an access-list in the IP world.

During the PLOGI operation, the class of service is negotiated. The class of service is used to ensure that the fabric will be lossless. Different classes of service use different methods to achieve this.

The currently defined classes are the following:

- Class 1: dedicated connection (not used)
- Class 2: connectionless multiplexed
- Class 3: datagram
- Class F: Fabric switch to switch

Two end hosts will always choose class 2 or 3 for upper layer protocol exchanges.

The classes work using credits. There are 2 kinds of credits defined: Buffer to Buffer (B2B) and End to End (EE).

B2B credits are connectionless and hop by hop. Each interface has a certain number of B2B credits corresponding to how many packets it can receive. A port will send a R_RDY (receive ready) primitive signal to notify that it is ready to receive a frame. B2B credits are used for class 2 and 3.

EE credits work between source and destination and use an ACK primitive signal to notify that a frame has been received. EE credits are used for classes 1, 2 and F.

As a general rule, a timestamp is applied to a frame when entering a switch. If due to lack of B2B credits, a frame resides in a switch for more than 500ms, it is dropped. This usually causes a catastrophic error (i.e. blue screen of death) or data corruption because many upper layer protocols are not able to cope with drops.

We will have a look to class 2 and 3 only as those are the only ones used in the field. Class 2 is also usually seen only on AIX systems, while class 3 being simpler, it is the most used one.

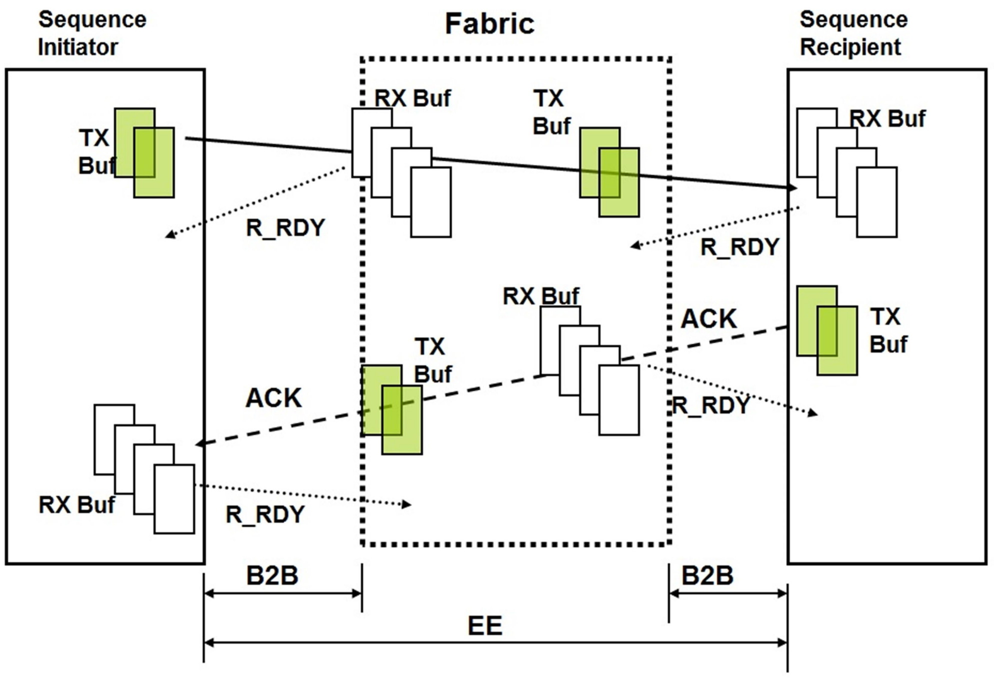


Figure 28: FC credits (source: cisco.com)

A class 2 communication between 2 N ports will use B2B credits hop by hop and EE credits for all data frames exchanged between the 2 end points. This guarantees delivery (as frames are sent only if there are B2B credits available) and non-delivery notification (due to the ACK mechanism) without any guarantee in throughput and in-order delivery.

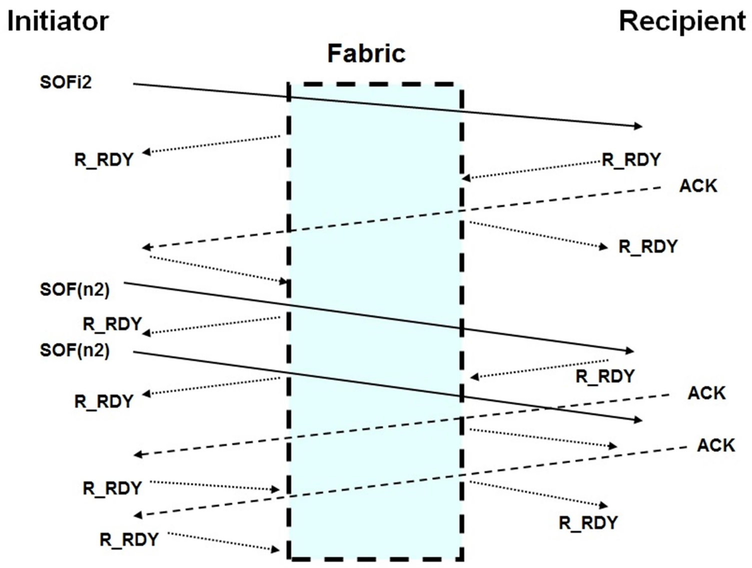


Figure 1: flow control in class 2 (source: cisco.com)

A class 3 communication uses only the B2B credits hop by hop and there is no guarantee delivery (due to the lack of ACK mechanism). In order delivery and throughput are not guaranteed as well.

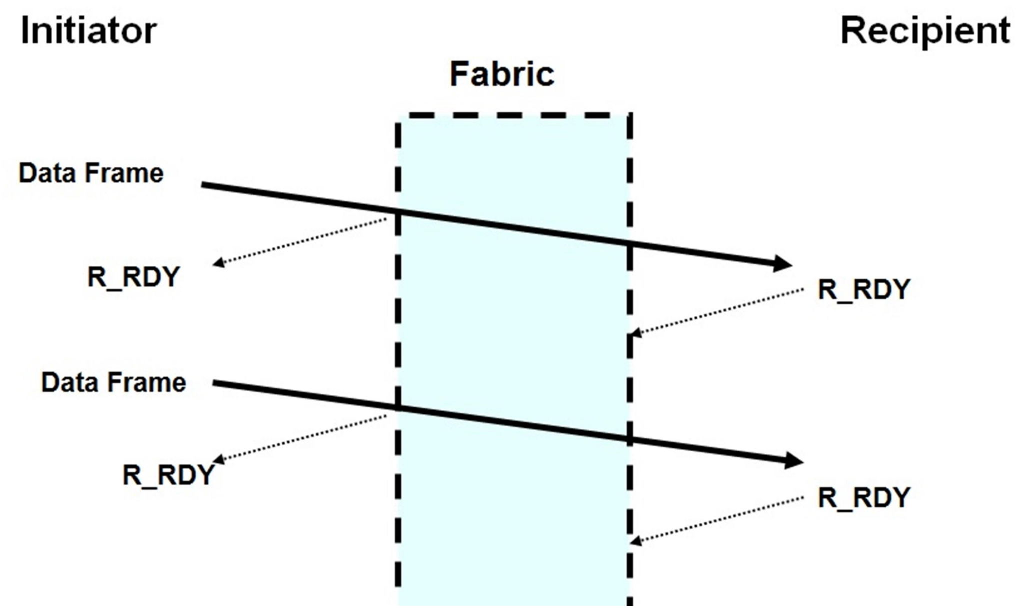


Figure 29: class 3 traffic flow (source: cisco.com)

While a hop by hop congestion control is good as it is used to create a dropless fabric, the B2B credit system has some important issues that as of today haven't been completely addressed yet.

The biggest problem that causes the majority of the issues in the FC world is the so called "slow draining device". When a device cannot cope with the traffic it receives, the R_RDY primitive signals may not be sent back fast enough to allow initiators to send other traffic. This may create a head of line blocking where the whole fabric slows down because inter switch links may start to buffer frames while waiting for a R_RDY coming from the affected device that may never come. At this point the frames that are buffered in the switches start to time out after 500ms and get dropped.

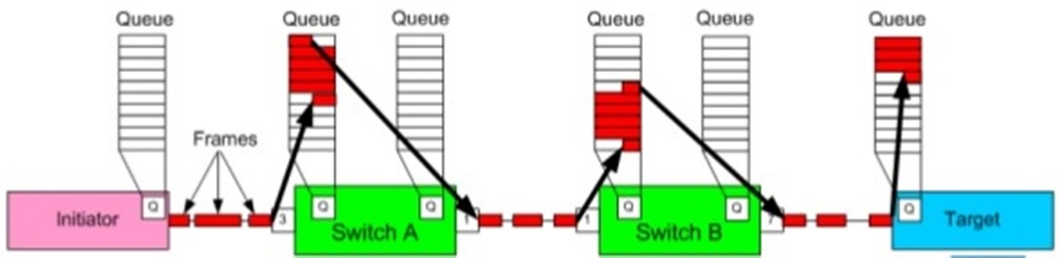


Figure 30: target unable to cope with demand, network starts buffering (source: EMC.com)

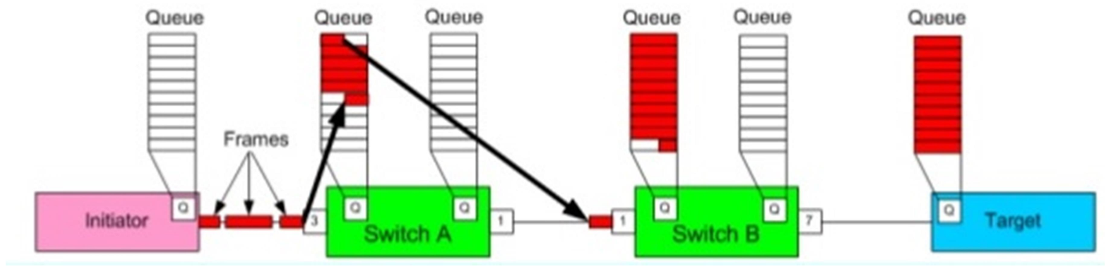


Figure 31: when buffers are full, the switches must drop the traffic (source: EMC.com)

The only way to get rid of this issue is to identify the slow draining device (by for example monitoring the I/O latency on initiators or CPU usage on the targets) and remove it temporarily from the network.

A similar issue may also happen over geographical links or links where the round trip time is higher where the B2B credit number is too low. In this case the R_RDY signal may be delayed and cause excessive buffering in the rest of the network.

Fibre channel over Ethernet (FCoE)

As we have already mentioned, the industry is trying to consolidate the LAN and SAN environment with the goal of removing one of the two networks and have a single infrastructure to carry both types of traffic.

Given the greater penetration of LAN (Ethernet) networks, it is simpler for most organizations to let go of the FC SAN network and migrate it to Ethernet.

The issue encountered here is that upper layer protocols designed to work on top of Ethernet have historically been created with the constraints of possessing some mechanism or resilience to packet drop, while upper layers protocols do not possess these features as we analyzed in the previous chapter.

A packet loss may cause a crash in the Operating System or data corruption, it is then necessary to create mechanisms that transform Ethernet into a (almost) lossless protocol.

FCoE [24] is a suite of protocols born to achieve this goal: allow for encapsulation of FC traffic into an Ethernet frame, allow for discovery of a device capabilities and auto configuration, and hop by hop congestion control to avoid drops.

The first step to move from FC to FCoE is to substitute the Ethernet NIC and the FC HBA and install the FCoE CNA adapters into the devices. A CNA adapter will give access to both Ethernet and FCoE. A CNA is usually bundled with hardware ASIC able to quickly perform encapsulation and decapsulation of FC frames into Ethernet and negotiate parameters via the various protocols composing the FCoE suite (more information later).

A CNA is a network card that accepts usually 2 types of transceivers: SFP+ with fiber optical cables or TWINAX cables that incorporate transceivers. At the moment there is no CNA that is able to use Cat5 or Cat6 Ethernet cables. CNAs also are currently only available at 10 Gbps speed.

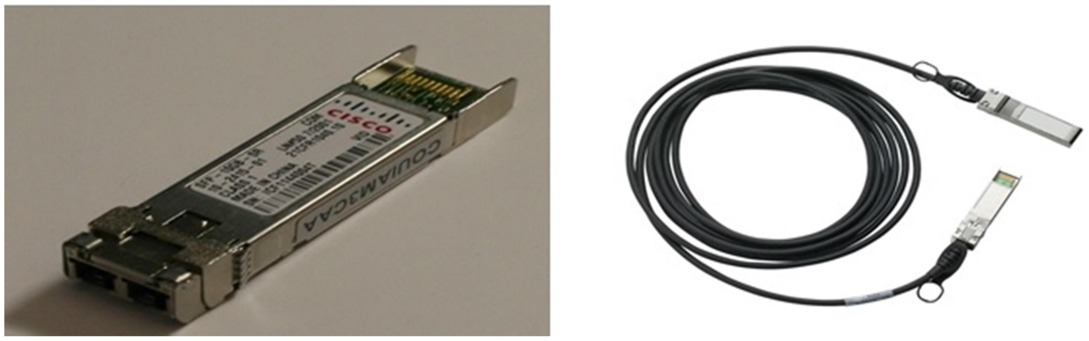


Figure 32: STP and Twinax (source: cisco.com)

FCoE is a joint collaboration between the T11 and the IEEE groups and specifications are split between.

FCoE is fully defined in T11 FC-BB-5 standard, but the protocols that are required to make FCoE work over Ethernet are defined in IEEE 802.1Qbb (priority flow control, PFC), 802.1Qaz (enhanced transmission services, ETS and datacenter bridging exchange, DCBX).

Traditionally, through the 802.3x, Ethernet can stop the transmission of frames via the PAUSE mechanism; however the PAUSE mechanism completely stops the transmission for all frames and has no granularity to stop a single flow or class. As FC has strict rules on how long a frame can be buffered before being dropped, the PAUSE feature may not be indicated for FC encapsulated traffic as it will stop it as well as the normal Ethernet traffic that is lower priority than the FC one. The priority flow control mechanism [25] extends the PAUSE feature and allows for a more granular approach where one or more of the classes defined in Ethernet (via the CoS field in the 802.1p and 802.1Q vlan header). In this case, some more talkative classes can be temporarily stopped to allow FC encapsulated traffic to take priority and be delivered.

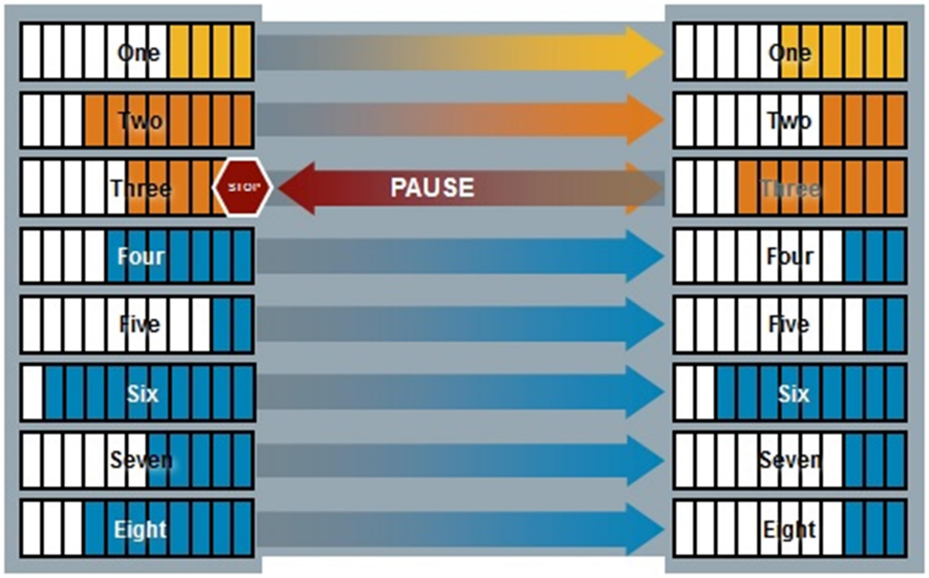


Figure 33: PFC blocking class 3 (source: cisco.com)

The Enhanced Transmission Services instead allow two devices to communicate in order to configure a bandwidth reservation for a certain class of service (even if no FCoE is involved). This reservation is used to guarantee bandwidth in case of oversubscription over a link. When a given traffic class doesn't utilizes the allocated reserved bandwidth, the bandwidth is intelligently shared between the other traffic classes.

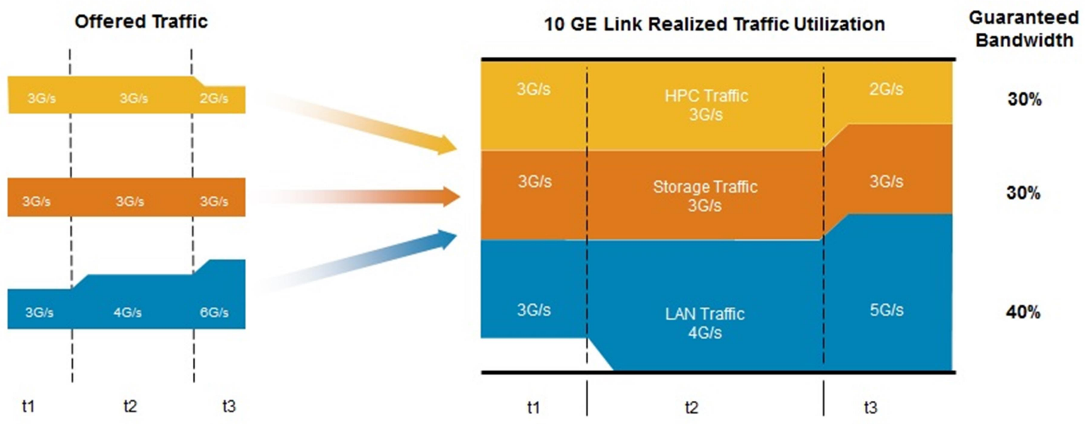


Figure 34: ETS reserving 3 Gbps for Storage Traffic (source: cisco.com)

DCBX is a protocol used for the discovery of directly connected devices, their capabilities (for pre-FIP devices) and performs configuration operation for each of the FCoE related features. It is built as an extension of LLDP (Local Link Discovery Protocol) a vendor neutral layer 2 discovery protocol similar to the Cisco Discovery Protocol (CDP).

The functions that are discovered and handled by DCBX are the following:

- Priority flow control (PFC)
- Bandwidth management (ETS)
- Congestion management
- Applications discovery (FCoE)
- Logical link down

The Logical Link down function is needed because a link down event in FC would also cause a link down event in Ethernet. With this function, the logical FC link can be flapped separately from the Ethernet one. The FC link then can restart its protocol to re-establish the connectivity (FLOGI, PLOGI and so on).

DCBX is used to negotiate the following parameters:

- QoS related parameters (PFC, scheduling parameters, MTU per class)
- CoS value that will be used to mark FCoE parameters
- Ethernet link state (up/down)

This information is pushed from the Fibre Channel Forwarder (FCF), the switch where the host device is connected, to the end device itself to avoid configuring each device singularly and cause possible mistakes.

Once DCBX verifies that two directly connected devices are able to use FCoE, the FCoE Initialization Protocol (FIP) is used to negotiate the remaining parameters needed for FCoE and perform the initial steps of the FCoE protocol (FLOGI).

The FIP protocol is used to exchange information on which vlan will be used for FCoE. Vlans are statically allocated by FCF devices to be Ethernet or FCoE vlans. An FCoE vlan will only carry FCoE traffic, this is important because if the same vlan would be also used to carry normal Ethernet (for example due to misconfiguration where another switch in the network uses the same vlan for normal Ethernet) traffic, the vlan may become congested with Ethernet traffic and the FCoE flow would be disrupted.

When a host device is connected to a switch, DCBX takes care of the capabilities negotiation; if the host device is able to use FCoE, FIP will start and will try to discover which vlan number will be used to encapsulate the FCoE traffic into Ethernet. The vlans number is required to allow for encapsulation and marking of the packets with the CoS values received via DCBX. The vlan discovery is initially performed into the native vlan.

When a vlan has been discovered the FIP protocol will perform a FLOGI to receive a FC ID and then FCoE will be established and start following the rules of the FC protocol.

The resulting FC encapsulated Ethernet packet will have a source mac address composed of a part called FC-MAP concatenated to the FC ID received during FIP FLOGI. The FC-MAP is a 24 bits identifier used in an FCoE fabric as required value for the most significant 24 bits of a MAC address in order to ensure the uniqueness of the produced FC MAC address in the fabric. If a unique vlan is assigned to each FCoE fabric, the FC-MAP configuration is not required, but if different FCoE fabrics are mapped to the same vlan, the FC-MAP is used to differentiate between them. FCF devices will accept frames only if the FC-MAP part of the destination MAC address matches the FC-MAP configured on the FCF itself.

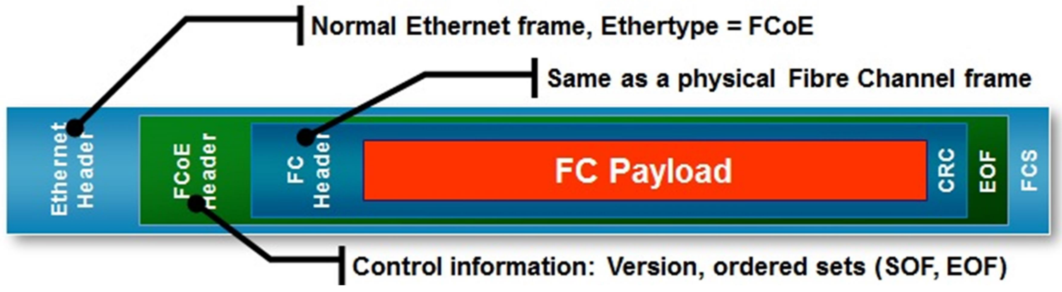
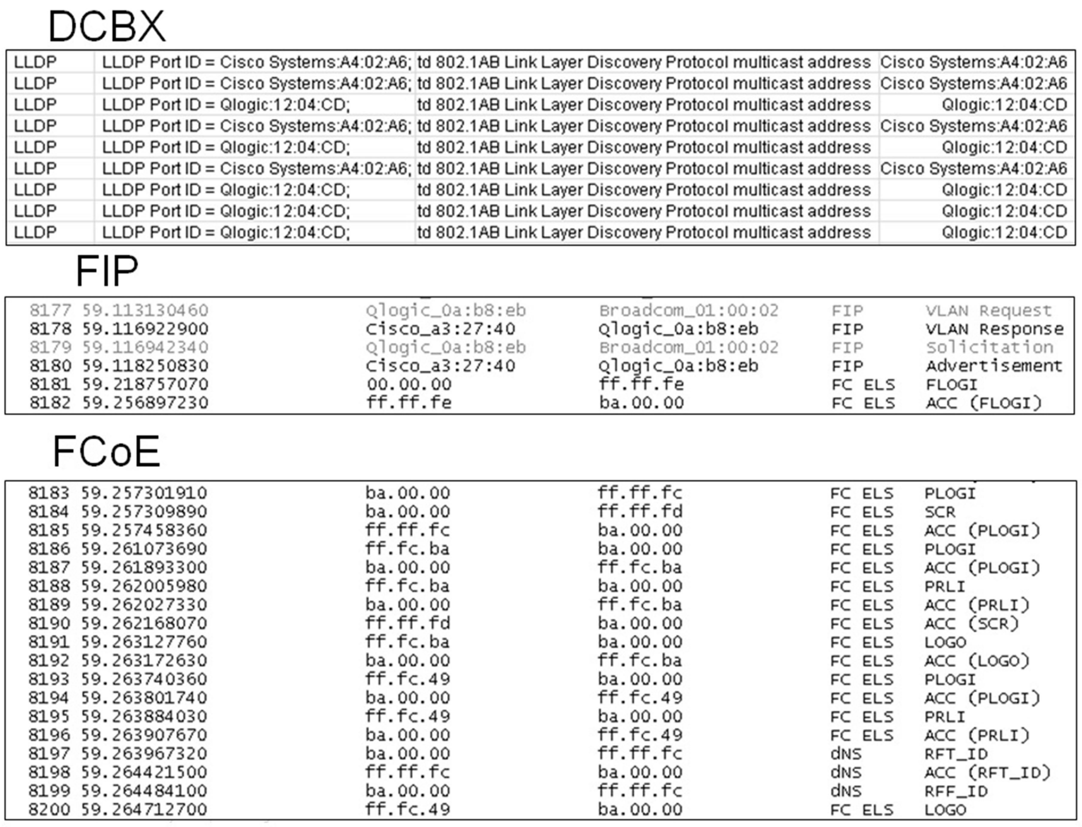

Normal Ethernet frame, Ethertype = FCoE
Same as a physical Fibre Channel frame
Ethernet Header
FCoE Header
FC Header
FC Payload
CRC
EOF
FCS
Control information: Version, ordered sets (SOF, EOF)

Figure 35: FCoE encapsulation (source: cisco.com)

The following picture presents a complete FCoE exchange from DCBX to FCoE:


DCBX

```
LLDP   LLDP Port ID = Cisco Systems:A4:02:A6; td 802.1AB Link Layer Discovery Protocol multicast address  Cisco Systems:A4:02:A6
LLDP   LLDP Port ID = Cisco Systems:A4:02:A6; td 802.1AB Link Layer Discovery Protocol multicast address  Cisco Systems:A4:02:A6
LLDP   LLDP Port ID = Qlogic:12:04:CD;        td 802.1AB Link Layer Discovery Protocol multicast address         Qlogic:12:04:CD
LLDP   LLDP Port ID = Cisco Systems:A4:02:A6; td 802.1AB Link Layer Discovery Protocol multicast address  Cisco Systems:A4:02:A6
LLDP   LLDP Port ID = Qlogic:12:04:CD;        td 802.1AB Link Layer Discovery Protocol multicast address         Qlogic:12:04:CD
LLDP   LLDP Port ID = Cisco Systems:A4:02:A6; td 802.1AB Link Layer Discovery Protocol multicast address  Cisco Systems:A4:02:A6
LLDP   LLDP Port ID = Qlogic:12:04:CD;        td 802.1AB Link Layer Discovery Protocol multicast address         Qlogic:12:04:CD
LLDP   LLDP Port ID = Qlogic:12:04:CD;        td 802.1AB Link Layer Discovery Protocol multicast address         Qlogic:12:04:CD
LLDP   LLDP Port ID = Qlogic:12:04:CD;        td 802.1AB Link Layer Discovery Protocol multicast address         Qlogic:12:04:CD
```

FIP

```
8177 59.113130460     Qlogic_0a:b8:eb    Broadcom_01:00:02   FIP      VLAN Request
8178 59.116922900     Cisco_a3:27:40     Qlogic_0a:b8:eb     FIP      VLAN Response
8179 59.116942340     Qlogic_0a:b8:eb    Broadcom_01:00:02   FIP      Solicitation
8180 59.118250830     Cisco_a3:27:40     Qlogic_0a:b8:eb     FIP      Advertisement
8181 59.218757070     00.00.00           ff.ff.fe            FC ELS   FLOGI
8182 59.256897230     ff.ff.fe           ba.00.00            FC ELS   ACC (FLOGI)
```

FCoE

```
8183 59.257301910     ba.00.00           ff.ff.fc            FC ELS   PLOGI
8184 59.257309890     ba.00.00           ff.ff.fd            FC ELS   SCR
8185 59.257458360     ff.ff.fc           ba.00.00            FC ELS   ACC (PLOGI)
8186 59.261073690     ff.fc.ba           ba.00.00            FC ELS   PLOGI
8187 59.261893300     ba.00.00           ff.fc.ba            FC ELS   ACC (PLOGI)
8188 59.262005980     ff.fc.ba           ba.00.00            FC ELS   PRLI
8189 59.262027330     ba.00.00           ff.fc.ba            FC ELS   ACC (PRLI)
8190 59.262168070     ff.ff.fd           ba.00.00            FC ELS   ACC (SCR)
8191 59.263127760     ff.fc.ba           ba.00.00            FC ELS   LOGO
8192 59.263172630     ba.00.00           ff.fc.ba            FC ELS   ACC (LOGO)
8193 59.263740360     ff.fc.49           ba.00.00            FC ELS   PLOGI
8194 59.263801740     ba.00.00           ff.fc.49            FC ELS   ACC (PLOGI)
8195 59.263884030     ff.fc.49           ba.00.00            FC ELS   PRLI
8196 59.263907670     ba.00.00           ff.fc.49            FC ELS   ACC (PRLI)
8197 59.263967320     ba.00.00           ff.ff.fc            dNS      RFT_ID
8198 59.264421500     ff.ff.fc           ba.00.00            dNS      ACC (RFT_ID)
8199 59.264484100     ba.00.00           ff.ff.fc            dNS      RFF_ID
8200 59.264712700     ff.fc.49           ba.00.00            FC ELS   LOGO
```

Figure 36: complete FCoE exchange (source:cisco.com)

Design considerations for SAN

The design topologies for an Ethernet network and a FC SAN are very similar: both designs aim to increase reliability and bandwidth while keeping the costs as low as possible.

This is a picture of the two different types of network and a consolidation achieved by using FCoE:

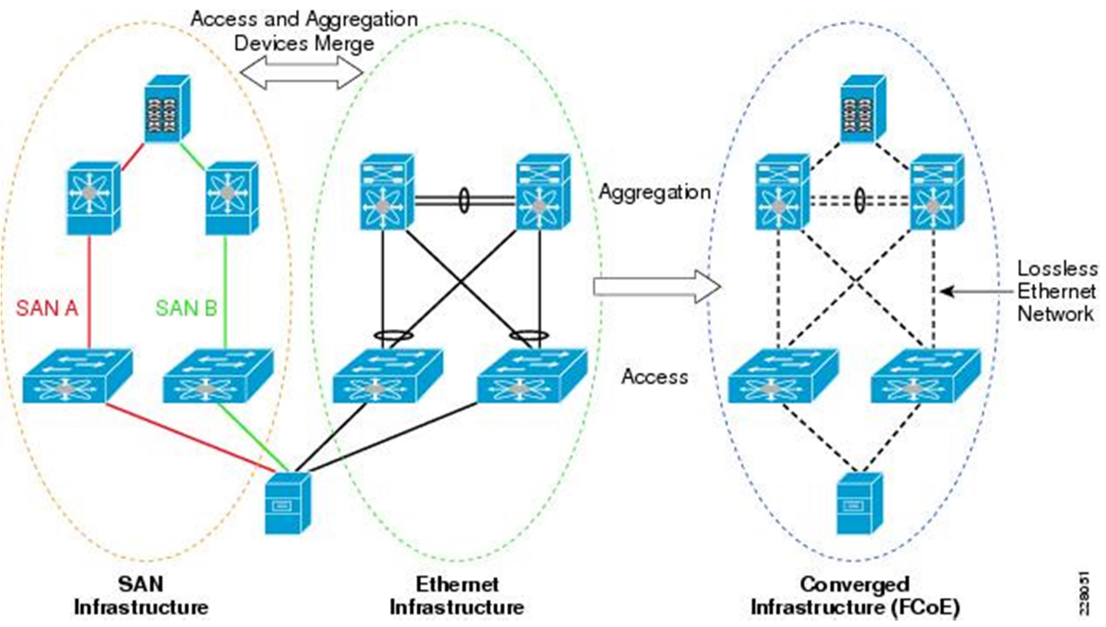

Access and Aggregation Devices Merge
SAN A
SAN B
Aggregation
Lossless Ethernet Network
Access
SAN Infrastructure
Ethernet Infrastructure
Converged Infrastructure (FCoE)
228051

Figure 37: FC vs. Ethernet design and FCoE consolidation (source: cisco.com)

By looking at this picture we can notice that redundancy is achieved in different ways in the SAN and Ethernet world: a device with 2 connections to a upstream switch uses 2 different Fabrics (1 Fabric corresponds to 1 vsan) while the same device will use a link configured with the same vlan to redundantly connect to an Ethernet switch.

As we know, Ethernet uses vlan mostly to isolate the broadcast traffic and reduce the number of packets that get flooded to the network. The FC SAN uses vsans to securely separate devices that need to access different resources. As we remember, when a FC device logs in into the network, he will discover all the devices that are present in the network that he has access to; when a device like a storage arrays logs out or errors, all the devices in the same vsan need to be notified and this may cause unnecessary traffic (there is no broadcast in FC, all packets need to be replicated to each device singularly). This causes a strain on the network and the switches and the administrator needs to avoid it given the criticality of the services involved. In FC SAN networks, “routing” between vsans needs to be enabled via a feature called IVR that is out of the scope of this document. This lack of routing, segregates even better the fabrics.
Multihoming in FC SAN is handled at the operating system level via particular software that will detect multiple paths towards a storage array and take care of the failover if one of the paths will

fail. As a result, each device will have to be connected to each upstream switch via a single physical or logical (port channel) link.

In Ethernet instead, when a device is multihomed, independently of the fact that the configuration of the link could be active/standby or active/active, a single link is chosen to forward the broadcast traffic and the normal unicast traffic is instead load balanced across all the active links. This allows us 3 different choices for redundancy:

- Active/standby: a device will be connected to 2 different upstream switches. The active link will receive and forward broadcast and unicast/multicast traffic. The upstream switch where the device is connected will learn the MAC address of the machine on that port. The switch connected to the standby link will not see any traffic and will not learn the MAC, thus there will be forwarding only of broadcast traffic that will be dropped by the host device as it is received on the standby link.
- Active/Active: a device will be connected to 2 different upstream switches. The two links will act as active/standby for different vlans, achieving a system similar to per-vlan STP, where a link is active for a certain vlan and standby for another vlan.
- Active/Active with bundling: a device will be connected to 2 different upstream switches. In this case the switches will have some feature used to simulate a single virtual switch. The end host device will be able to bundle the two links into a port channel and have all vlans active at all times achieving better load balancing by hashing higher level protocol information (IP addresses/TCP ports). In this case, the end host device will always decide a single link where the broadcast is sent, so the MAC address of the end host device will be learned only by one of the two switches. To avoid unicast flooding, the switches will have to implement a mechanism to synchronize the MAC address tables (among other information). In Cisco world, the feature that allows two switches to act as a single logical one is called VSS in the Catalyst 6500 product family and vPC in the Nexus family.

Virtual Port Channel

One of the biggest limitations of the STP protocol is that when redundant links are used to connect a device to 2 upstream switches, one of those links will be blocked and will be unutilized. When multiple links are needed between two devices, they can be bundled into a port channel to increase available bandwidth and reliability, but the achieved reliability will still be less than the one of a device connected to two upstream devices.

To get the best of both solutions (increased bandwidth and reliability), Cisco introduced the Virtual Port Channel (vPC) [26] feature into the Nexus family of switches.

The vPC allows two switches to appear as a single logical switch to the devices downstream so that links connected from the downstream device to the upstream switches in vPC configuration can be bundled together as a port channel. A port channel will allow all links to be forwarding and will load balance flows based on hash calculated on traffic parameters like source/destination MAC addresses, IP addresses and TCP/UDP ports.

If one of the links will go down, nothing will happen in the port channel and the other link will transparently take over the traffic while with separate links, a STP event will trigger and a TCN will be sent towards the root bridge then downstream to all switches and cause limited unicast flooding for all the vlans that were carried over the link that went down.

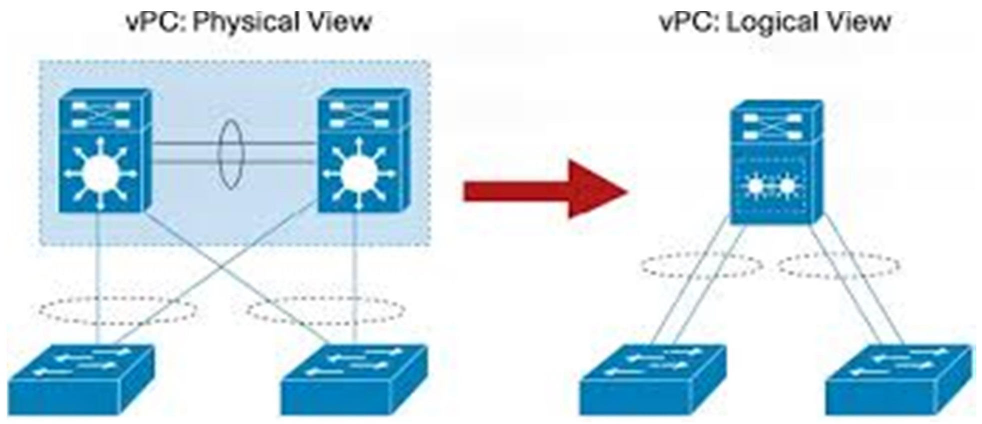


Figure 38: vPC physical vs. logical view (source: cisco.com)

Pairs of switches configured in vPC can also be connected together however in this scenario a little precaution needs to be taken: if two switches are seen as a single logical one, then all the links between the two switch pairs needs to be bundled together as a single port channel. If each physical switch in the downstream vPC pair would connect with a port channel to the logical upstream switch, there would be 2 different port channels between the 2 vPC pairs, and one of them would be blocked because of STP!

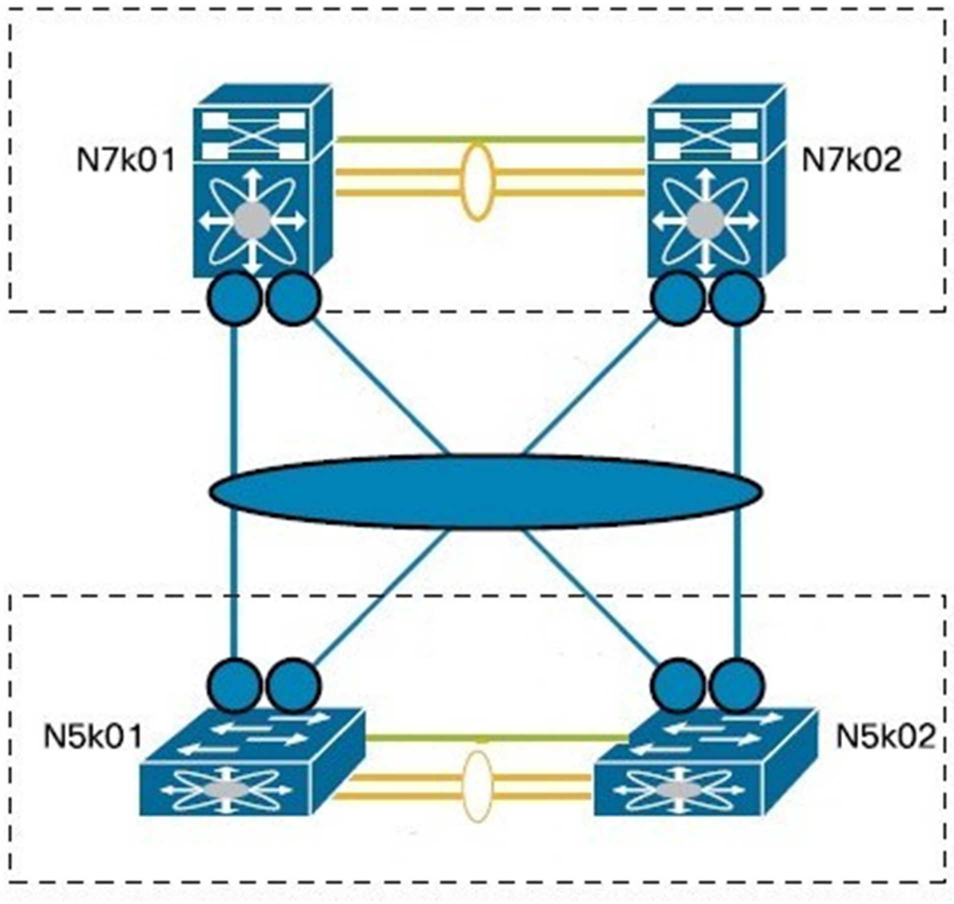


Figure 39: correctly configured double sided vPC (source: cisco.com)

When a downstream device (let's call this SRC) as a host is connected to a vPC pair, it doesn't know that those are two physical switches, but it still thinks that it is a single device. Let's say that there is a single unicast traffic flow outbound the end host device. One of the two port channel links will be chosen by the hashing algorithm to be used for this flow. The switch (let's call this switch A) that receives this traffic will then learn the MAC address of SRC, but the other switch in the vPC pair will not (let's call this switch B). Switch A will forward upstream the traffic toward the destination (let's call this DST) and let's say that DST is another end host device connected to the vPC pair via a port channel in the same way the source end host device is.

If DST will load balance the return traffic to go through switch B, switch B will then forward it to SRC, but SRC's MAC address will not be in switch B MAC address table because switch B never saw any traffic coming from SRC. Switch B will unicast flood.

To avoid this problem, the vPC switches are connected via a special protocol that runs over a link called the vPC peer link. This special protocol (called CFS) is used to synchronize some parameters between the two switches to make them behave as one. A non-exhaustive list of the parameters that are synchronized are:

- LACP ID
- MAC address tables (new MACs learned are propagated)
- Optionally the switch configuration
- IGMP groups
- HSRP/VRRP virtual MAC addresses for virtual IP

Additionally, CFS is also used to verify the compatibility of the configuration between the two vPC switches. If the configuration would become incompatible for some core parameters (as vlan list trunked on a port, or STP configuration) an inconsistency would be triggered and the ports connected to the secondary switch would be suspended.

While strictly not needed, SPT should not be disabled in vPC configuration as it is used as a guard safe mechanism in case of misconfigurations or split brain situations.

The two vPC switches are connected via a port channel (vPC peer link) that is used to carry traffic between the peers in case the destination device for a flow was single connected to one of the two vPC switches. The peer link carries also the CFS packets.

Another link connects the two vPC switches and it is used for keepalives. If the keepalive link or the vPC peer links are down, the secondary vPC switch will suspend his port channels because if also the other link between the two vPC switches would go down, we would be in a split brain situation where the switches would have no possibility of synchronizing.

It is important to understand that vPC is a layer 2 solution: only layer 2 links can be vPC port channels and due to limitations, if a layer 3 routing protocol adjacency has to be formed with the two vPC switches, it should happen on a separate layer 3 link. This limitation is due to the fact that packets forwarded from a vPC switch to the peer switch over the vPC peer link will be dropped if received from a vPC port channel connecting to a host to ensure that the MAC address of the host will be learned over the vPC port channel connecting that host to the vPC pair (via CFS MAC learning) and not over the vPC peer link because it would create suboptimal forwarding.

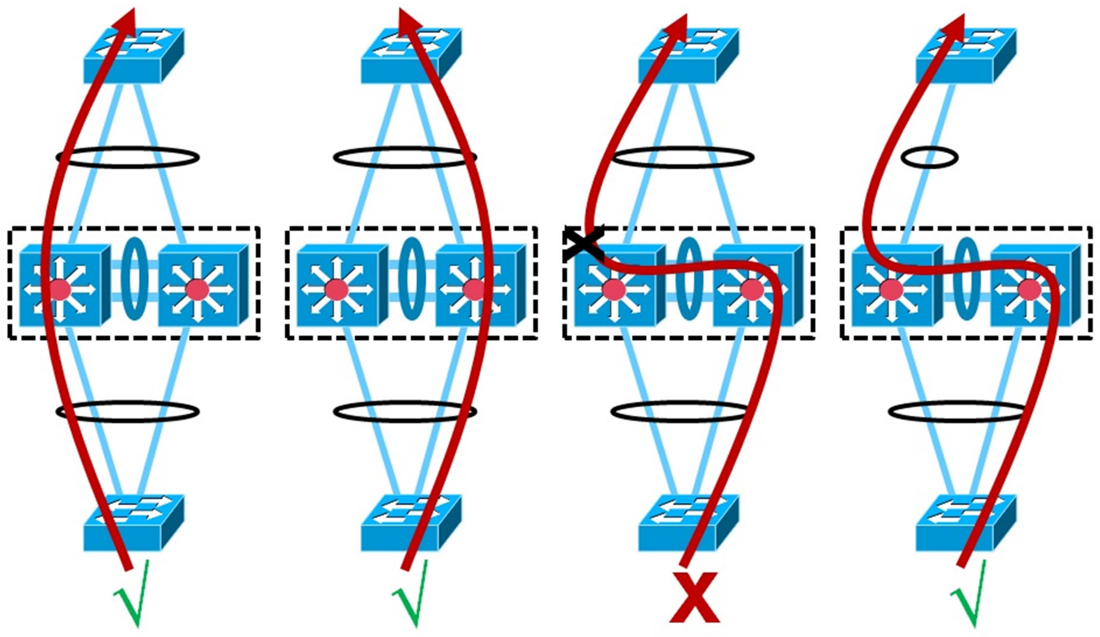


Figure 40: vPC forwarding rules (source: cisco.com)

In case of layer 3 adjacencies, if a routing protocol packet (as an OSPF hello) coming from a device that established a routing adjacency with one of the two vPC pair switches is received on the other vPC switch, it will contain a destination MAC address belonging to the other vPC switch and will be forwarded over the vPC peer link, thus it will be dropped by the other vPC switch due to the forwarding rules that state: “do not accept a packet over the vPC peer link, if there is a vPC port channel connecting to the source of that packet”.

To avoid this (and other similar problems with certain devices not compliant with some RFC), the peer-gateway feature has been created and this feature allows a packet received from the peer link to go through and not being dropped. This is however an unsupported configuration for routing adjacencies and should not be used in production.

Unified Computing System

The majority of the hardware equipment in a datacenter is computing. The issues with computing are the faster life cycle compared to the other equipment and the higher probability of hardware failures.

The scalability in terms of operations of a huge datacenter will require lots of people to swap failed hardware and lots of other employees to reconfigure the servers and the services.

Number of failures is linear with the number of devices and complex devices made of multiple parts (like a server composed of storage, networking, CPU, RAM and so on) have a higher chance of breaking.

While a centralized storage solution can be used to boot the servers from a remote disk and avoid reinstalling an operating system if a machine dies, this solution is far away from being perfect. In the case of a FC/FCoE SAN deployment, in particular, disks and security policies (zoning) are tied to NWWN and PWWN burn in into the CNA/HBA and modifying those addresses to match existing policies will require an operating system to run on the machine, so this is a catch-22. The other possible solution is to reconfigure the zoning in the SAN switches and the LUN masking in the storage array, but this requires time and human intervention.

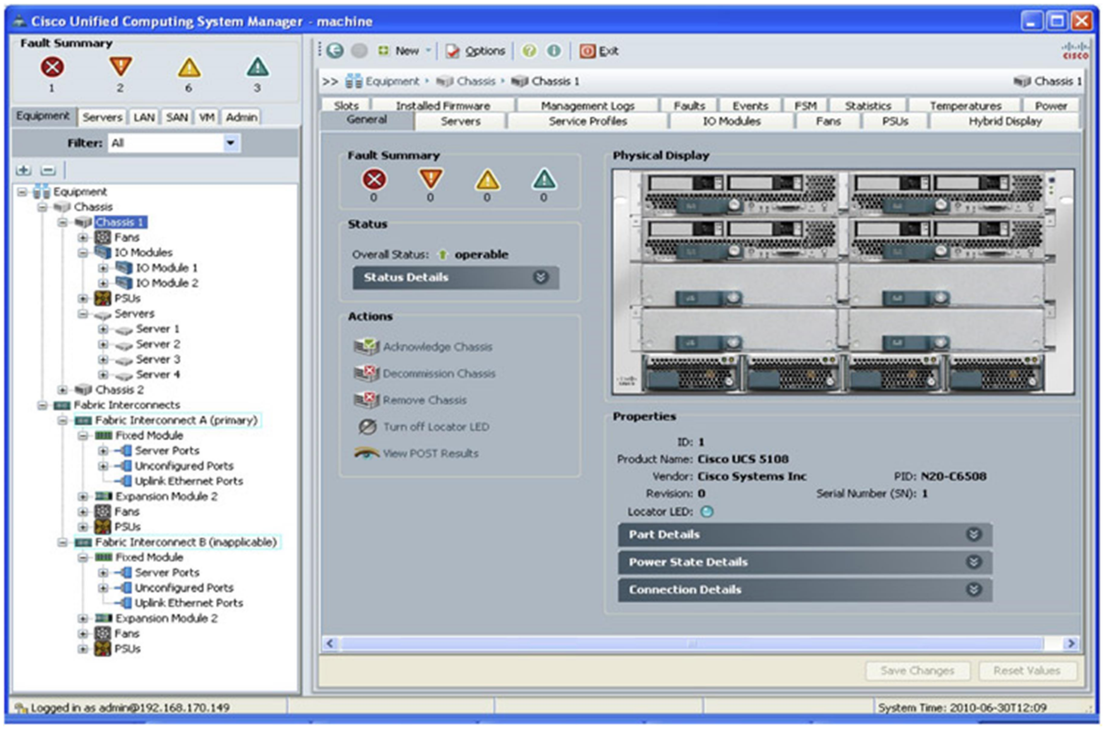
Issues may also arise in case of other features that may be tied to other identifiers of the machine (e.g. MAC addresses for port security), in this case, the reconfiguration in case of hardware failure is even more complicated.

The secondary issue is deployment: nowadays the deployment of hardware and network equipment has become the slowest part of the global datacenter life cycle.

A third issue is a lack of central management that gives a complete view of all the servers and their health.

To solve these issues, Cisco invented the Unified Computing System, a system consisting of a pair of redundant switches (the Fabric Interconnects) that control chassis where server blades are installed. The UCS allows a user to configure a pool of elements (MAC addresses, PWWN, NWWN, boot policies and so on) that are automatically applied to the servers installed in the chassis. When a server is booted, a lightweight modified Linux operating system is pushed and installed in the server. This operating system is used by the UCS to burn the hardware identifiers and the boot policies in the server, so that at the next reboot, the server will assume the identity configured by the administrator.

A server configured to use those policies and suffering a hardware failure can be substituted by another server that will inherit the same policies and identifiers (MAC, WWN…) and will de facto assume the identity of the previous server and (if configured to boot from a remote disk), boot the same operating system and have the same data as the previous device.



Overall Status: operable

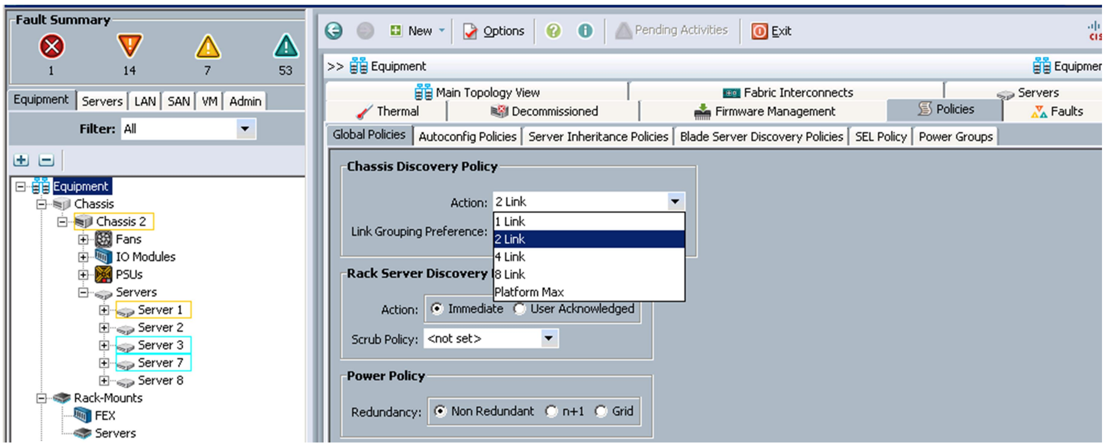
ID: 1
Product Name: Cisco UCS 5108
Vendor: Cisco Systems Inc PID: N20-C6508
Revision: 0 Serial Number (SN): 1

Logged in as admin@192.168.170.149
System Time: 2010-06-30T12:09

Figure 41: UCS manager (source: cisco.com)

The UCS Manager (UCSM) runs as part of the OS running on the Fabric Interconnect. Once the FI are configured to act as a cluster, a virtual IP (VIP) is used to intercept connections to the UCSM. Pointing a web browser to the VIP allows the user to download a Java applet and start the connection to the UCS.

Once connected, the UCSM will show the FI. The user will then be able to configure a policy to discover the chassis and the servers that are installed.



Chassis Discovery Policy

Action: 2 Link
Link Grouping Preference:

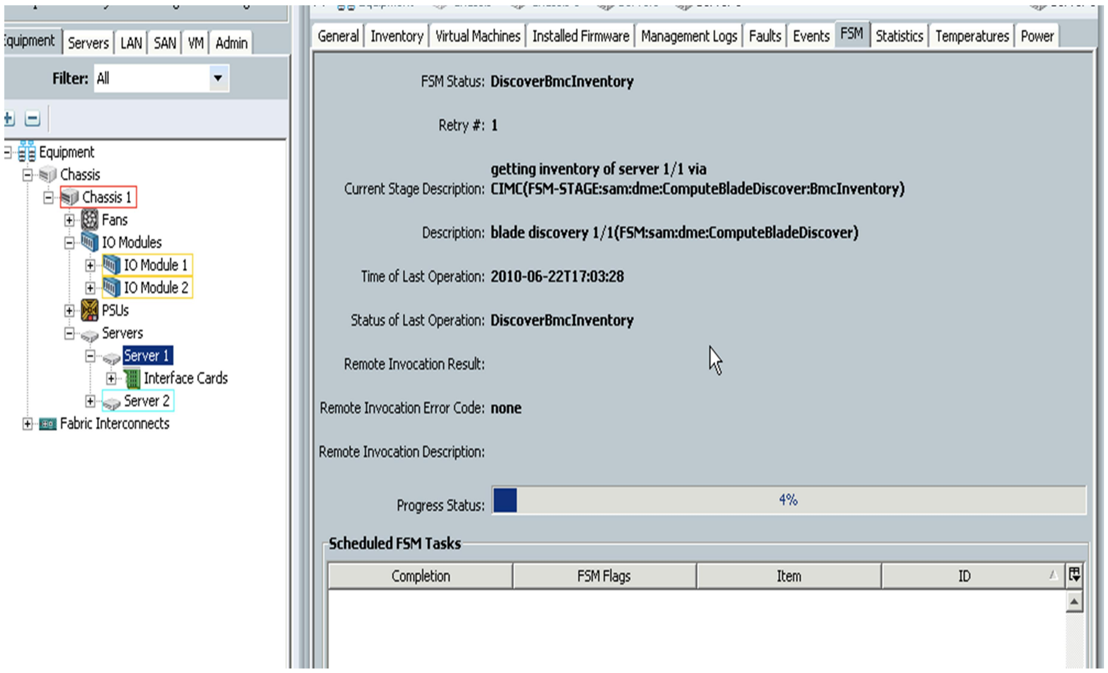
Rack Server Discovery

Action: [✓ Immediate] User Acknowledged
Scrub Policy: <not set>

Power Policy

Redundancy: [✓ Non Redundant] n+1 Grid

Figure 42: Chassis discovery policy (source: cisco.com)



FSM Status: DiscoverBmcInventory

Retry #: 1

Current Stage Description: getting inventory of server 1/1 via CIMC(FSM-STAGE:sam:dme:ComputeBladeDiscover:BmcInventory)

Description: blade discovery 1/1(FSM:sam:dme:ComputeBladeDiscover)

Time of Last Operation: 2010-06-22T17:03:28

Status of Last Operation: DiscoverBmcInventory

Remote Invocation Result:

Remote Invocation Error Code: none

Remote Invocation Description:

Progress Status: 4%

Figure 43: Server discovery (source: cisco.com)

When the servers have been discovered, the administrator can start configuring the policies that are used to program them.

The policies that are normally configured are:

- List of the vlans to be trunked to the servers
- List of the VSAN that the server has access
- A MAC address pool to be associated with the vNIC
- vNIC interfaces that will be presented to the host OS on the server
- VLAN list on each virtual NIC
- PWWN and NWWN address group to be associated with the vHBA
- vHBA interfaces that will be presented to the host OS on the server
- VSAN list on each HBA
- A boot policy to indicate to the server where to boot from

Datacenter interconnect (OTV)

Due to the businesses' complete dependency on the IT infrastructure to accomplish their goals, downtime has become unacceptable. Well organized IT organizations prepare tests to verify whether their hardware and software are able to cope with catastrophic incidents and properly recover.

This brought the IT organizations to not only look for active/standby software and hardware mechanisms of redundancy in the same datacenter, but also redundant datacenters in different geographical locations in order to survive natural events (i.e. earthquakes) or severe power events.

Among the redundancy exercises, the most important (and extreme) is called disaster recovery (DR).

A DR is a test where a whole datacenter is shut down and applications are restarted (usually in an almost stateless way) in a different datacenter in another part of the world.

IT organizations need to plan for this kind of event because a failure would mean a total downtime of the business with economical and brand damages. However it is expensive to build a whole datacenter in another location just to function as a standby site without running any useful activity.

Additionally there are some serious networking problems that arise when planning: the most complex of them is the layer 2 mobility issue.

It often happens that legacy software applications written in the past and still being used, assume that the network is a layer 2 network where all the machines part of a redundant cluster are in the same IP network. If those applications run on a VM, we expect the VM to be able to move to a different location and still find the same (layer 2) network in order for the application running on the guest OS to perform properly.

This limitation burdens the network designer with the problem of extending layer 2 between the top of the rack switches from the main datacenter to the DR one.

Various solutions are possible, simple layer 2 extensions via spanning tree, VPLS, EoMPLS, L2VPN, LISP…

Cisco recently came out with a technology called Overlay Transport Virtualization (OTV) [27] [28] where Nexus devices can exchange layer 2 packets (both multicast and unicast) by encapsulating them into GRE packets.

Standard L2 VPN techniques have issues because they require a full mesh to be built between sites, they cause flooding due to unknown unicast and MAC propagation or have STP related constraints (STP failures in one site affect all).

OTV uses a multicast address allocated for control plane functions to discover neighbors on the network.

This control plan is used to advertise which MAC addresses are present at which sites to avoid flooding.

Packets destined for another site are encapsulated in a unicast GRE packet and sent over the WAN.

Multicast packets that need to reach multiple sites are encapsulated in GRE using a configurable set of multicast IP addresses.

As no state of the site is exposed outside to the other sites, local failures are contained.

Before MAC addresses can be discovered, the switches need to discover each other. This can be done by either configuring static unicast neighbors or having the switches to listen and advertise their presence over a multicast address (called Control Group).

Adjacencies are maintained over the multicast group allowing for maximum efficiency and reducing the amount of WAN traffic occupied by control plane protocols.

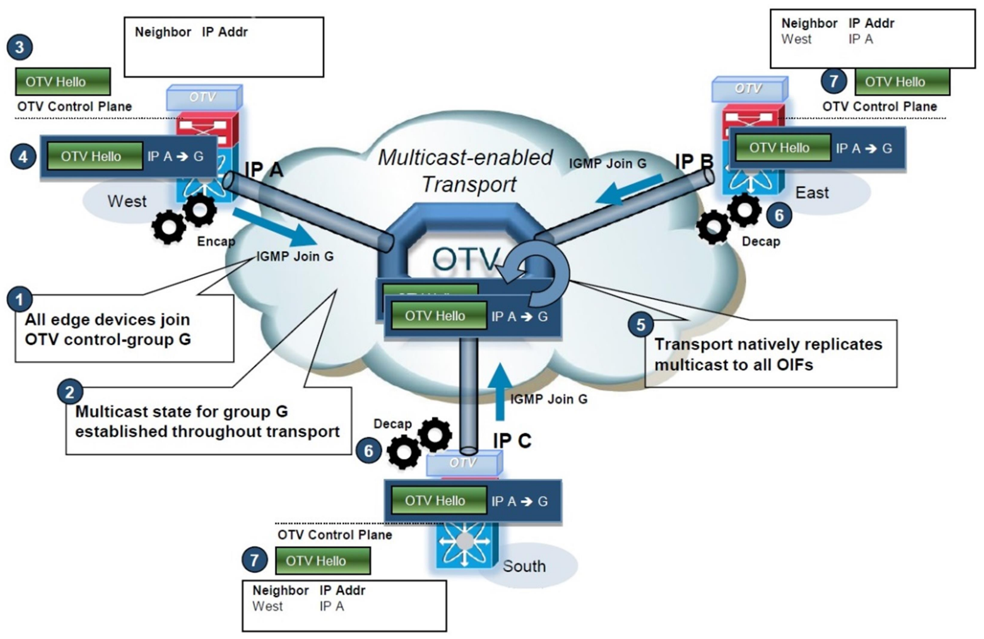


Figure 44: OTV control plane establishment (part 1) (source: cisco.com)

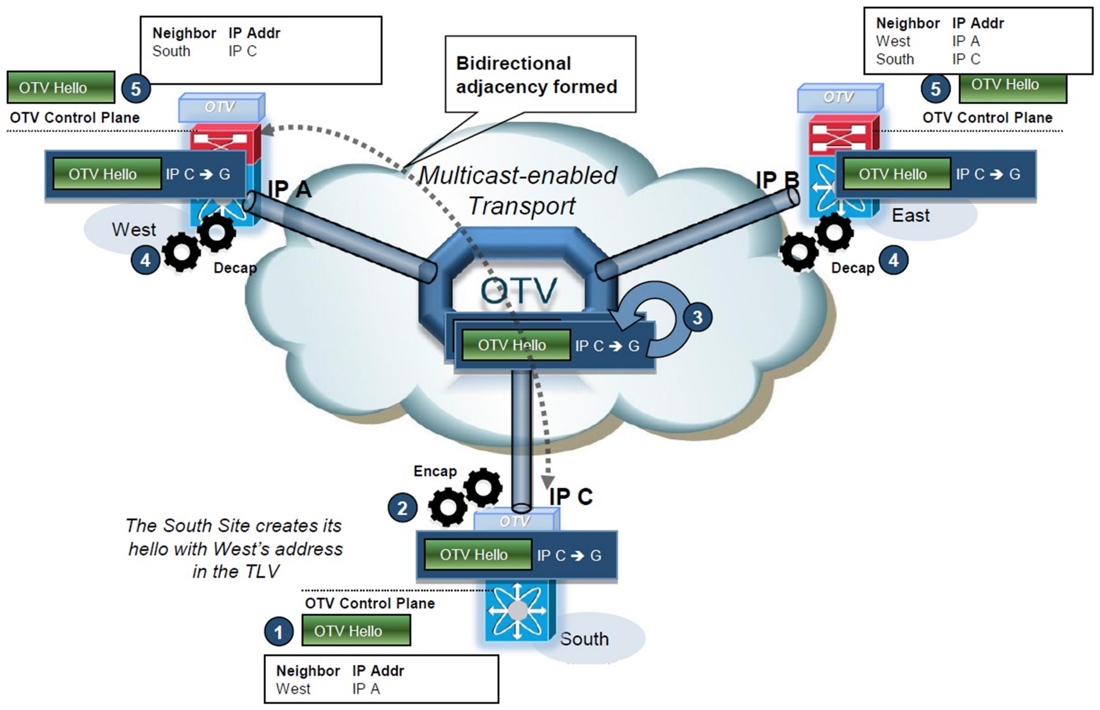


Figure 45: OTV control plane establishment (part 2) (source: cisco.com)

Once adjacencies are formed, ISIS [29] [30] is used to advertise the MAC addresses present at each site.

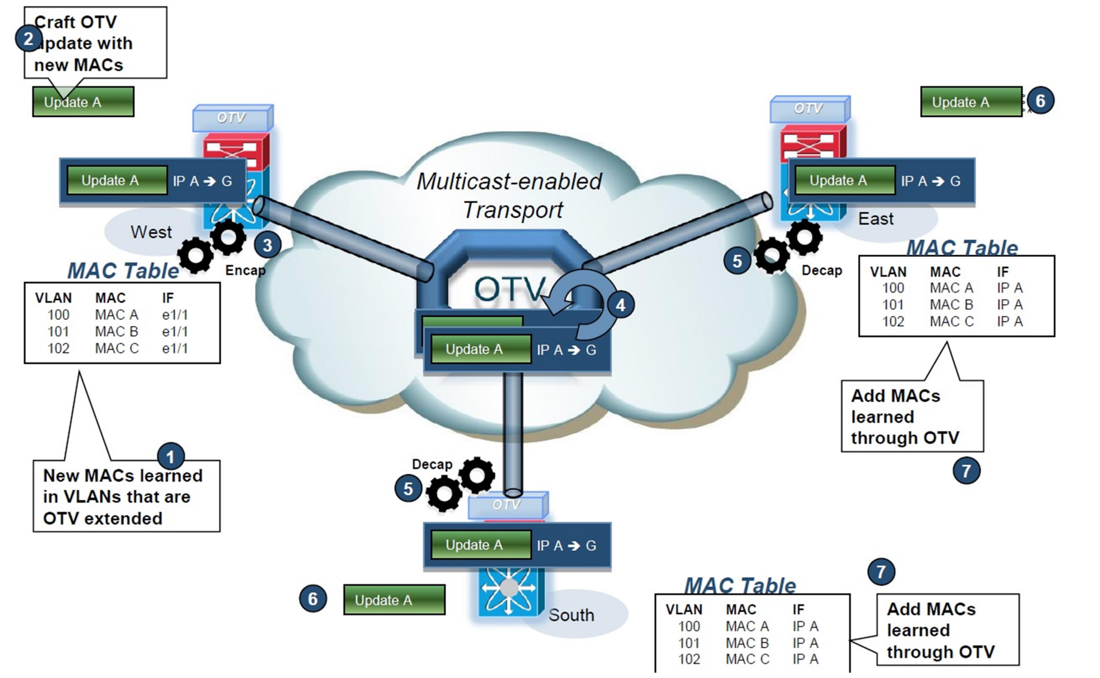


Figure 46: MAC address learning (source: cisco.com)

If a unicast packet must be forwarded to a MAC address belonging to another site, the OTV device will encapsulate it into an OTV packet and send it unicast to that site.

The OTV encapsulation has this format:

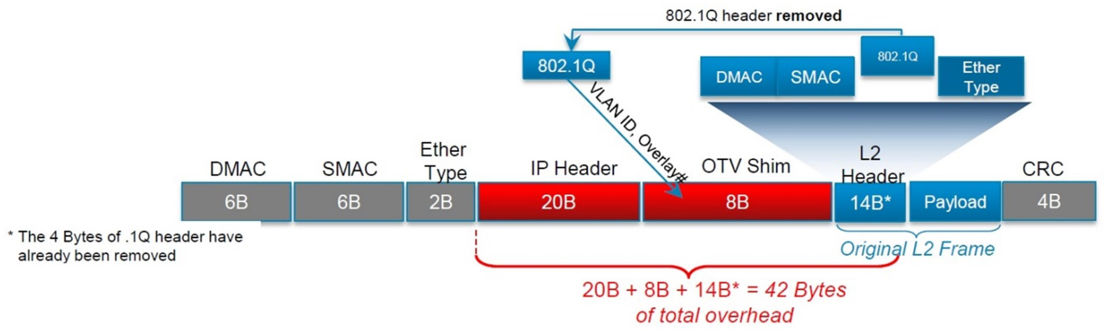


Figure 47: OTV encapsulation (source: cisco.com)

Layer 2 multicast packets are instead sent to different sites using a series of multicast addresses (data group) configured on the OTV devices on each site.

Multicast groups local to each site are mapped using round robin to a data group address and the packets are encapsulated as they were unicast, but using that data group as the external address.

Because the data group addresses can be less than the multicast groups used in all the sites, more than one multicast group can be assigned to the same data group and this can cause a lower efficiency because a site with listeners for group A will receive multicast for group B as well through the data group even if there are no listeners for group B in the local site.

Not all the layer 2 packets are forwarded by OTV: STP packets, in particular are never forwarded outside a site, so that failures are contained within and there is no risk of having root devices local to another site.

Same applies for unknown unicast: in OTV it is assumed that hosts are not just listeners or silent. Unknown unicast is forwarded in the local site, but never sent to the other sites. OTV devices can be also configured to spoof ARP response packets to avoid wasting WAN bandwidth.

The configuration for OTV mostly consists in having two special interfaces: the join interface and the overlay interface.

The join interface is a point to point layer 3 interface used to join the overlay network. It is the interface that sends and receives MAC reachability information and the data packets (both unicast and multicast).

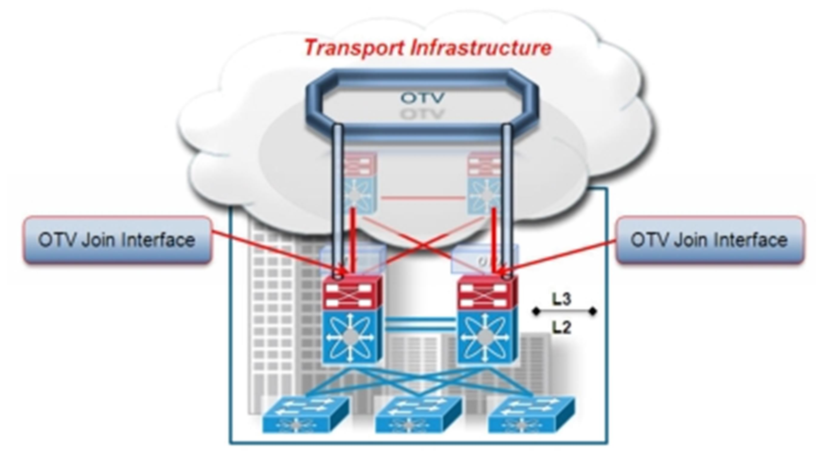


Figure 48: OTV join interface (source: cisco.com)

The overlay interface is a multi-access layer 2 interface containing the whole OTV configuration. The overlay interface is used to receive the layer 2 packets and forward them to the join interface for encapsulation.

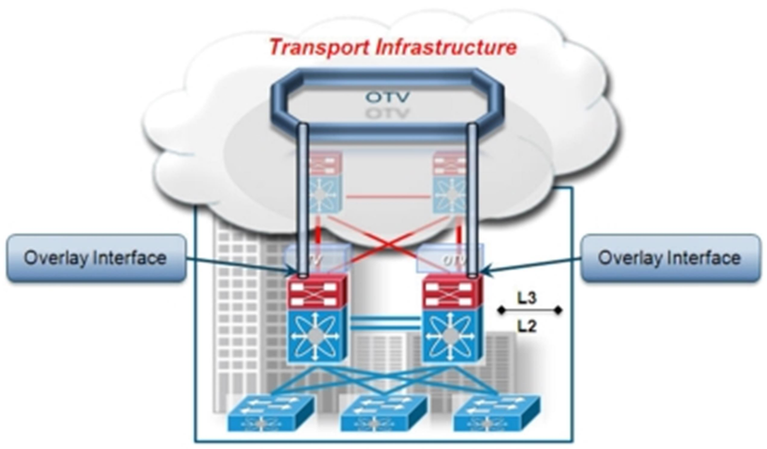


Figure 49: OTV overlay interface (source: cisco.com)

Example of a design for a datacenter

Now we are going to see an implementation of a design and we will go through the most common best practices regarding network layers and protocols.

The example will contain a fully working configuration example.

Topology

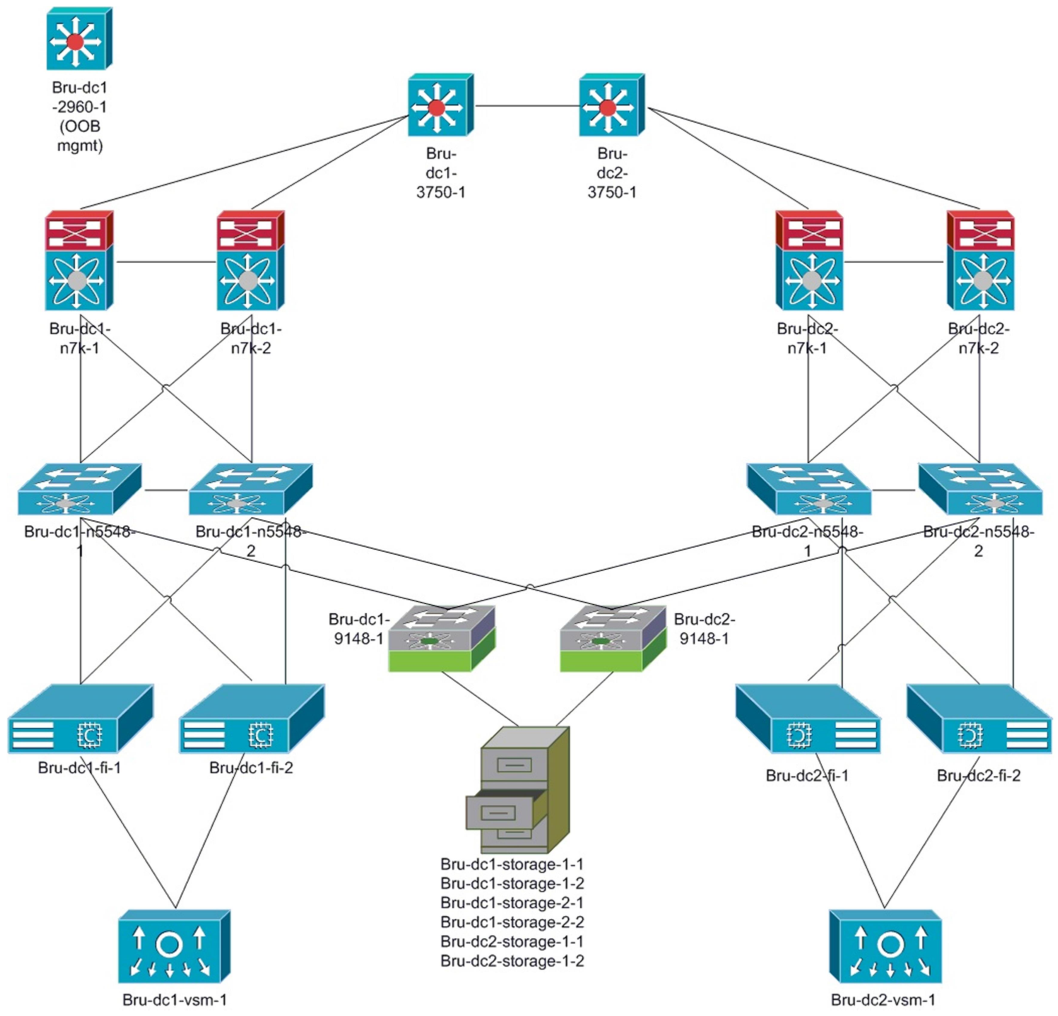


Figure 50: example topology

The detailed topology is the following:

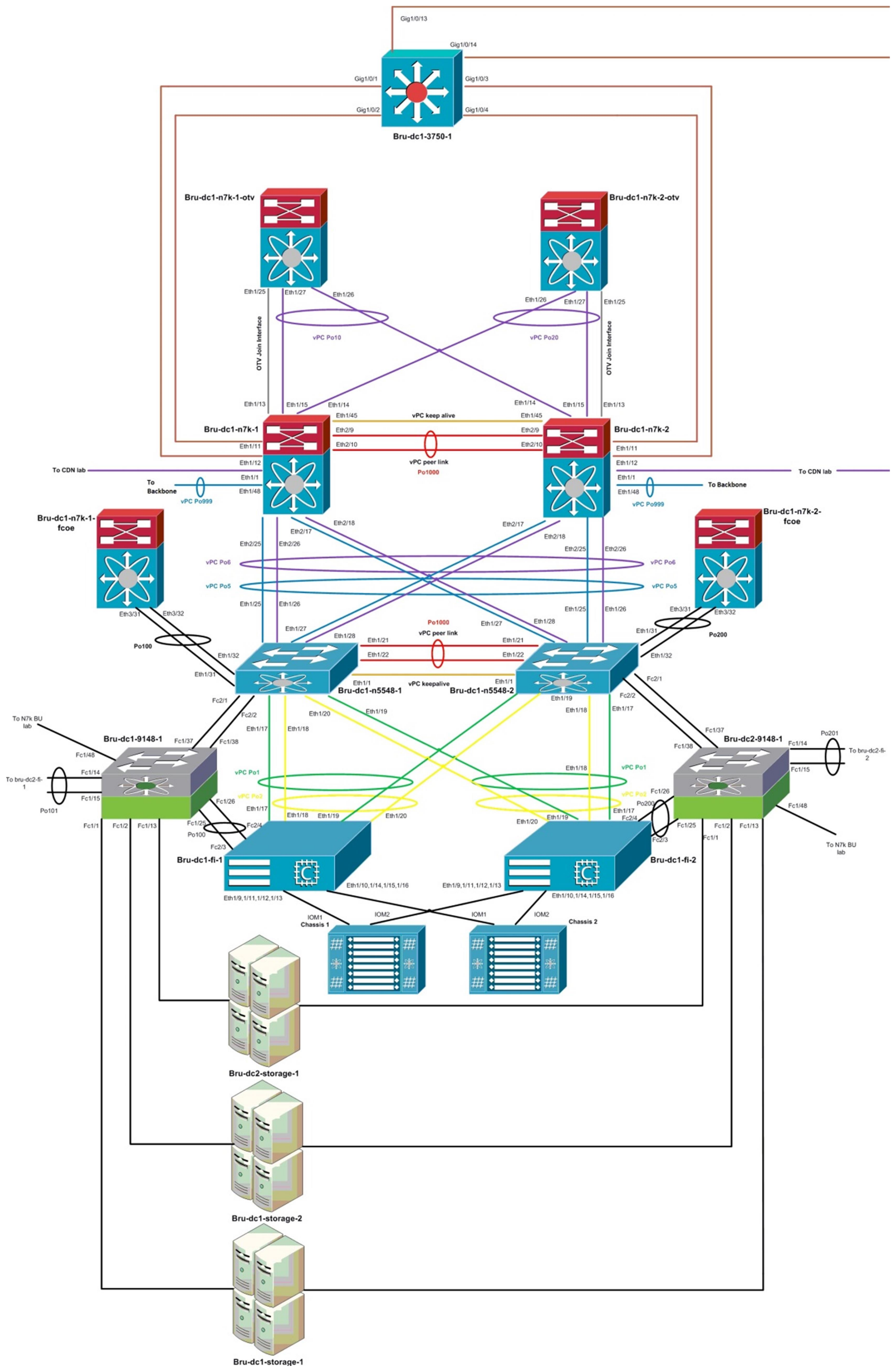
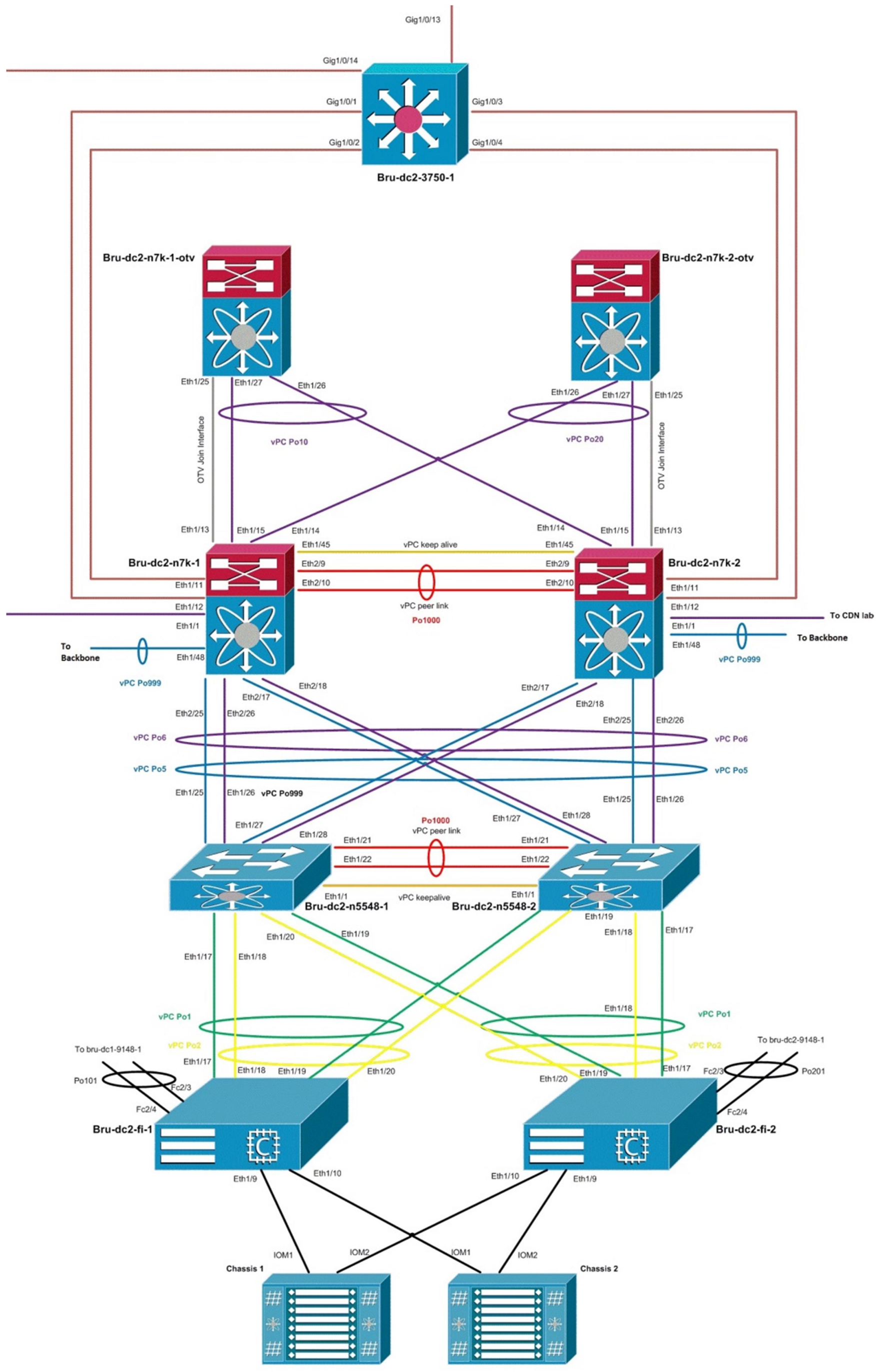


Figure 51: dc1 detailed topology



	Name	Notes
🟥	vPC peer link	Cannot be changed
🟧	vPC keepalive	Cannot be changed
🟦	Management links	Cannot be changed
🟪	Data links	
🟩	Mixed Links (temporary)	Mgmt vlans cannot be changed
🟨	Inactive links	
⬛	Fibre channel links (SAN)	Cannot be changed
🟫	L3 data links	
⬜	L3 mgmt links	Cannot be changed

Figure 52: dc2 detailed topology

General design considerations

Due to the restrictions in scale and the equipment, this example datacenter will be composed of Nexus 7000 [31] as core/aggregation devices, Nexus 5000 as top of the rack switches and UCS will act as end host devices. Two datacenter will be configured and will be connected using an OTV encapsulation to simulate traversal of an internet cloud (potentially owned by a third party ISP).

The servers will be ESXi machines with multiple VM. The servers will be connected to an UCS and will not possess hard disks, the storage will be reached via FCoE between the UCS and the Nexus 5000 and from there, the Nexus 5000 will act as a gateway to decapsulate FCoE and go towards a MDS [32] switch as legacy FC to a storage array containing both disks used to boot ESXi and disks where the VM are physically stored.

For test purposes, the FCoE will also be extended to the Nexus 7000 even if no targets are attached to it.

A Nexus 1000v will be installed to substitute ESXi's distributed vSwitch (DVS).

Management of ESXi and integrated lights out management of servers will be performed inband by using separated vlans that will be trunked towards the servers over virtual NIC interfaces configured on the UCS and the Nexus 1000v. The management vlan for ESXi will use a normal vSwitch, not the Nexus 1000v.

We are not going to detail the configuration of the storage array as it is not a Cisco device and it is not a network device.

General best practices

The first best practice is to decide on a naming convention that is easy to remember and possibly working as a DNS name.

For this example I've chosen to use the following naming convention to identify the devices:

(Geographical location)-(datacenter number)-(device family type)-(device number).

The name of the first core Nexus 7000 switch in datacenter 1 in Brussels will then be "bru-dc1-n7k-1". The switch will have its management IP registered in a DNS name as "bru-dc1-n7k-1.cisco.com".

Some organizations that own particularly big datacenters, may want to include building/floor/aisle number into the name of the devices. A designer however should pay particular attention to not make the naming convention too complicated. Device names that differ for a single number or letter in the middle of their name may cause numerous human errors where configuration changes may be performed on the wrong device.

All devices should be configured with a name and a domain server. This is important because all other servers (NTP, TACACS+, Radius, etc.) can now be referenced by name instead of IP. If one of those servers should be decommissioned or moved away, the IP may change, but the change wouldn't affect the network functionalities as long as the DNS resolution would just return a name. This is particularly important with AAA servers, because a non-responding server may prevent an administrator from logging into the device.

To avoid this issue, there should always be a backdoor mechanism of authentication configured as a console access using local credentials stored on the device or a fallback mechanism in case the device couldn't reach the AAA server. This is configured by default in the Nexus family while it was not the case in the standard IOS.

To protect access to the management of the switch, the IP address used for inband or out of band access to the switch should belong to a management virtual routing and forwarding table (VRF) so that routing problems in other VRF would not impact the management VRF. This VRF should also be used to reach all management services such as NTP, logging, AAA and so on.

Vlans should be configured with names for ease of understanding the configuration

It is strongly recommended to activate and configure the following services:

- SYSLOG server: used to store the logs from a device. Logs should have at least millisecond precision. Logs should also be configured to save a copy of the messages on disk.
- NTP server: used to have time synchronization between devices, very important if doing post mortem troubleshooting.
- DNS server: used to access name resolution services.
- AAA server: used for authentication, authorization and accounting.
- SNMP server: used for traps. It is recommended to configure an access list to restrict the SNMP poll access from specific management IP. SNMP v3 should be used if possible due to the enhanced security features.
- Control plane policing (CoPP [33]): used to rate limit the amount of control packets reaching the CPU.
- Scheduled jobs to save the configuration to a server.

Example of a configuration that can be used for management:

```
username admin password 5 $1$WRZWWCNL$eKzQhJJvHVxfz4aqpi/UA/  role network-admin
username waas password 5 $1$jZ/hHoM8$//YjZnb5L1KN52RQncxJJ1  role WAAS_user
username ACE password 5 $1$Y6D8Dx9t$uem0SfQ7KEFp0pWBgp8vE.  role vdc-admin
username dcnm-admin password 5 $1$7O17KVck$w8ywfQfc2pguqIK137si9/  role network-
admin
no password strength-check
ip domain-lookup
ip domain-name cisco.com
radius distribute
radius-server retransmit 5
radius-server host bru-dc1-ise-1.cisco.com key 7 "Lxq-levr1" authentication
accounting
```

```
radius commit
radius-server host bru-dc1-ise-1.cisco.com test username cisco password cisco
idle-time 3
aaa group server radius ISE
    server bru-dc1-ise-1.cisco.com
    deadtime 3
    use-vrf management
    source-interface mgmt0
ip access-list copp-system-acl-bgp
  10 permit tcp any gt 1024 any eq bgp
  20 permit tcp any eq bgp any gt 1024
ipv6 access-list copp-system-acl-bgp6
  10 permit tcp any gt 1024 any eq bgp
  20 permit tcp any eq bgp any gt 1024
ip access-list copp-system-acl-cts
  10 permit tcp any any eq 64999
  20 permit tcp any eq 64999 any
ip access-list copp-system-acl-dhcp
  10 permit udp any eq bootpc any
  20 permit udp any eq bootps any
  30 permit udp any any eq bootpc
  40 permit udp any any eq bootps
ip access-list copp-system-acl-eigrp
  10 permit eigrp any any
ip access-list copp-system-acl-ftp
  10 permit tcp any any eq ftp-data
  20 permit tcp any any eq ftp
  30 permit tcp any eq ftp-data any
  40 permit tcp any eq ftp any
ip access-list copp-system-acl-glbp
  10 permit udp any eq 3222 224.0.0.0/24 eq 3222
ip access-list copp-system-acl-hsrp
  10 permit udp any 224.0.0.0/24 eq 1985
ipv6 access-list copp-system-acl-hsrp6
  10 permit udp any ff02::66/128 eq 2029
ip access-list copp-system-acl-icmp
  10 permit icmp any any echo
  20 permit icmp any any echo-reply
ipv6 access-list copp-system-acl-icmp6
  10 permit icmp any any echo-request
  20 permit icmp any any echo-reply
ipv6 access-list copp-system-acl-icmp6-msgs
  10 permit icmp any any router-advertisement
  20 permit icmp any any router-solicitation
  30 permit icmp any any nd-na
  40 permit icmp any any nd-ns
  50 permit icmp any any mld-query
  60 permit icmp any any mld-report
  70 permit icmp any any mld-reduction
ip access-list copp-system-acl-igmp
  10 permit igmp any 224.0.0.0/3
mac access-list copp-system-acl-mac-cdp-udld-vtp
  10 permit any 0100.0ccc.cccc 0000.0000.0000
mac access-list copp-system-acl-mac-cfsoe
  10 permit any 0180.c200.000e 0000.0000.0000 0x8843
mac access-list copp-system-acl-mac-dot1x
  10 permit any 0180.c200.0003 0000.0000.0000 0x888e
mac access-list copp-system-acl-mac-fabricpath-isis
```

```
  10 permit any 0180.c200.0015 0000.0000.0000
  20 permit any 0180.c200.0014 0000.0000.0000
mac access-list copp-system-acl-mac-flow-control
  10 permit any 0180.c200.0001 0000.0000.0000 0x8808
mac access-list copp-system-acl-mac-gold
  10 permit any any 0x3737
mac access-list copp-system-acl-mac-l2pt
  10 permit any 0100.0ccd.cdd0 0000.0000.0000
mac access-list copp-system-acl-mac-lacp
  10 permit any 0180.c200.0002 0000.0000.0000 0x8809
mac access-list copp-system-acl-mac-lldp
  10 permit any 0180.c200.000c 0000.0000.0000 0x88cc
mac access-list copp-system-acl-mac-otv-isis
  10 permit any 0100.0cdf.dfdf 0000.0000.0000
mac access-list copp-system-acl-mac-stp
  10 permit any 0100.0ccc.cccd 0000.0000.0000
  20 permit any 0180.c200.0000 0000.0000.0000
mac access-list copp-system-acl-mac-undesirable
  10 permit any any
ip access-list copp-system-acl-msdp
  10 permit tcp any gt 1024 any eq 639
  20 permit tcp any eq 639 any gt 1024
ip access-list copp-system-acl-ntp
  10 permit udp any any eq ntp
  20 permit udp any eq ntp any
ipv6 access-list copp-system-acl-ntp6
  10 permit udp any any eq ntp
  20 permit udp any eq ntp any
ip access-list copp-system-acl-ospf
  10 permit ospf any any
ipv6 access-list copp-system-acl-ospf6
  10 permit 89 any any
ip access-list copp-system-acl-otv-as
  10 permit 4 any any
ip access-list copp-system-acl-pim
  10 permit pim any 224.0.0.0/24
  20 permit udp any any eq pim-auto-rp
  30 permit ahp any 224.0.0.13/32
ip access-list copp-system-acl-pim-reg
  10 permit pim any any
ipv6 access-list copp-system-acl-pim6
  10 permit pim any ff02::d/128
  20 permit udp any any eq pim-auto-rp
ip access-list copp-system-acl-radius
  10 permit udp any any eq 1812
  20 permit udp any any eq 1813
  30 permit udp any any eq 1645
  40 permit udp any any eq 1646
  50 permit udp any eq 1812 any
  60 permit udp any eq 1813 any
  70 permit udp any eq 1645 any
  80 permit udp any eq 1646 any
ipv6 access-list copp-system-acl-radius6
  10 permit udp any any eq 1812
  20 permit udp any any eq 1813
  30 permit udp any any eq 1645
  40 permit udp any any eq 1646
  50 permit udp any eq 1812 any
  60 permit udp any eq 1813 any
  70 permit udp any eq 1645 any
  80 permit udp any eq 1646 any
ip access-list copp-system-acl-rip
  10 permit udp any 224.0.0.0/24 eq rip
ipv6 access-list copp-system-acl-rip6
  10 permit udp any ff02::9/64 eq 521
ip access-list copp-system-acl-sftp
  10 permit tcp any any eq 115
  20 permit tcp any eq 115 any
ip access-list copp-system-acl-snmp
  10 permit udp any any eq snmp
  20 permit udp any any eq snmptrap
ip access-list copp-system-acl-ssh
  10 permit tcp any any eq 22
  20 permit tcp any eq 22 any
ipv6 access-list copp-system-acl-ssh6
  10 permit tcp any any eq 22
  20 permit tcp any eq 22 any
ip access-list copp-system-acl-tacacs
  10 permit tcp any any eq tacacs
  20 permit tcp any eq tacacs any
ipv6 access-list copp-system-acl-tacacs6
  10 permit tcp any any eq tacacs
  20 permit tcp any eq tacacs any
ip access-list copp-system-acl-telnet
  10 permit tcp any any eq telnet
  20 permit tcp any any eq 107
  30 permit tcp any eq telnet any
  40 permit tcp any eq 107 any
ipv6 access-list copp-system-acl-telnet6
  10 permit tcp any any eq telnet
  20 permit tcp any any eq 107
  30 permit tcp any eq telnet any
  40 permit tcp any eq 107 any
ip access-list copp-system-acl-tftp
  10 permit udp any any eq tftp
  20 permit udp any any eq 1758
  30 permit udp any eq tftp any
  40 permit udp any eq 1758 any
ipv6 access-list copp-system-acl-tftp6
  10 permit udp any any eq tftp
  20 permit udp any any eq 1758
  30 permit udp any eq tftp any
  40 permit udp any eq 1758 any
ip access-list copp-system-acl-traceroute
  10 permit icmp any any ttl-exceeded
  20 permit icmp any any port-unreachable
ip access-list copp-system-acl-undesirable
  10 permit udp any any eq 1434
ip access-list copp-system-acl-vpc
  10 permit udp any any eq 3200
ip access-list copp-system-acl-vrrp
  10 permit 112 any 224.0.0.0/24
ip access-list copp-system-acl-wccp
  10 permit udp any eq 2048 any eq 2048

class-map type control-plane match-any copp-system-class-critical
```

```
  match access-group name copp-system-acl-bgp
  match access-group name copp-system-acl-pim
  match access-group name copp-system-acl-rip
  match access-group name copp-system-acl-vpc
  match access-group name copp-system-acl-bgp6
  match access-group name copp-system-acl-igmp
  match access-group name copp-system-acl-msdp
  match access-group name copp-system-acl-ospf
  match access-group name copp-system-acl-pim6
  match access-group name copp-system-acl-rip6
  match access-group name copp-system-acl-eigrp
  match access-group name copp-system-acl-ospf6
  match access-group name copp-system-acl-otv-as
  match access-group name copp-system-acl-mac-l2pt
  match access-group name copp-system-acl-mac-otv-isis
  match access-group name copp-system-acl-mac-fabricpath-isis
  match protocol mpls router-alert
class-map type control-plane match-any copp-system-class-exception
  match exception ip option
  match exception ip icmp unreachable
  match exception ipv6 option
  match exception ipv6 icmp unreachable
class-map type control-plane match-any copp-system-class-important
  match access-group name copp-system-acl-cts
  match access-group name copp-system-acl-glbp
  match access-group name copp-system-acl-hsrp
  match access-group name copp-system-acl-vrrp
  match access-group name copp-system-acl-wccp
  match access-group name copp-system-acl-hsrp6
  match access-group name copp-system-acl-pim-reg
  match access-group name copp-system-acl-mac-lldp
  match access-group name copp-system-acl-icmp6-msgs
  match access-group name copp-system-acl-mac-flow-control
class-map type control-plane match-any copp-system-class-l2-default
  match access-group name copp-system-acl-mac-undesirable
  match protocol mpls
class-map type control-plane match-any copp-system-class-l2-unpoliced
  match access-group name copp-system-acl-mac-stp
  match access-group name copp-system-acl-mac-gold
  match access-group name copp-system-acl-mac-lacp
  match access-group name copp-system-acl-mac-cfsoe
  match access-group name copp-system-acl-mac-cdp-udld-vtp
class-map type control-plane match-any copp-system-class-management
  match access-group name copp-system-acl-ftp
  match access-group name copp-system-acl-ntp
  match access-group name copp-system-acl-ssh
  match access-group name copp-system-acl-ntp6
  match access-group name copp-system-acl-sftp
  match access-group name copp-system-acl-snmp
  match access-group name copp-system-acl-ssh6
  match access-group name copp-system-acl-tftp
  match access-group name copp-system-acl-tftp6
  match access-group name copp-system-acl-radius
  match access-group name copp-system-acl-tacacs
  match access-group name copp-system-acl-telnet
  match access-group name copp-system-acl-radius6
  match access-group name copp-system-acl-tacacs6
  match access-group name copp-system-acl-telnet6
```

```
class-map type control-plane match-any copp-system-class-monitoring
  match access-group name copp-system-acl-icmp
  match access-group name copp-system-acl-icmp6
  match access-group name copp-system-acl-traceroute
class-map type control-plane match-any copp-system-class-normal
  match access-group name copp-system-acl-dhcp
  match access-group name copp-system-acl-mac-dot1x
  match redirect dhcp-snoop
  match protocol arp
class-map type control-plane match-any copp-system-class-redirect
  match redirect arp-inspect
class-map type control-plane match-any copp-system-class-undesirable
  match access-group name copp-system-acl-undesirable
policy-map type control-plane copp-system-policy
  class copp-system-class-exception
    set cos 1
    police cir 360 kbps bc 250 ms conform transmit violate drop
  class copp-system-class-critical
    set cos 7
    police cir 39600 kbps bc 250 ms conform transmit violate drop
  class copp-system-class-important
    set cos 6
    police cir 1060 kbps bc 1000 ms conform transmit violate drop
  class copp-system-class-management
    set cos 2
    police cir 10000 kbps bc 250 ms conform transmit violate drop
  class copp-system-class-normal
    set cos 1
    police cir 680 kbps bc 250 ms conform transmit violate drop
  class copp-system-class-redirect
    set cos 1
    police cir 280 kbps bc 250 ms conform transmit violate drop
  class copp-system-class-monitoring
    set cos 1
    police cir 130 kbps bc 1000 ms conform transmit violate drop
  class copp-system-class-l2-unpoliced
    police cir 8 gbps bc 5 mbytes conform transmit violate transmit
  class copp-system-class-undesirable
    set cos 0
    police cir 32 kbps bc 250 ms conform drop violate drop
  class copp-system-class-l2-default
    police cir 100 kbps bc 250 ms conform transmit violate drop
  class class-default
    set cos 0
    police cir 100 kbps bc 250 ms conform transmit violate drop
control-plane
  service-policy input copp-system-policy
snmp-server contact fmarabot@cisco.com
snmp-server user ACE vdc-admin auth md5 0x793b99565cb099502bba124c7849bd58 priv
0x793b99565cb099502bba124c7849bd58 localizedkey
snmp-server user waas WAAS_user auth md5 0x488de185dda3d5f2b98b5dfa06c2834a priv
0x488de185dda3d5f2b98b5dfa06c2834a localizedkey
snmp-server user admin network-admin auth md5 0x33a403f8087721c148af5f576c01c921
priv 0x33a403f8087721c148af5f576c01c921 localizedkey
snmp-server user dcnm-admin network-admin auth md5
0xf94e7016148fb201d84b39995253292b priv 0xf94e7016148fb201d84b39995253292b
localizedkey
snmp-server host 172.22.29.51 traps version 2c public  udp-port 2162
```

```
rmon event 1 log trap public description FATAL(1) owner PMON@FATAL
rmon event 2 log trap public description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap public description ERROR(3) owner PMON@ERROR
rmon event 4 log trap public description WARNING(4) owner PMON@WARNING
rmon event 5 log trap public description INFORMATION(5) owner PMON@INFO
snmp-server community public group network-operator
ntp server 10.48.56.1 use-vrf management
aaa authentication login default group ISE
aaa accounting default group ISE
logging logfile messages 6
logging server 10.48.56.53 5 use-vrf management
logging timestamp milliseconds
scheduler job name backup_config
copy running-config startup-config vdc-all
 copy startup-config tftp://bru-dc1-ad-1.cisco.com/$(SWITCHNAME)-config-
$(TIMESTAMP).cfg vrf management vdc-all
end-job
scheduler schedule name midnight
  job name backup_config
  time daily 00:00
```

vPC and Spanning Tree Protocol best practices

vPC is a complicated feature and has numerous tricks to be deployed properly. Without going too much into details we are going to talk about the minimum amount of best practices needed for the implementation into this example design. As vPC interacts with STP, we are going to speak about the two together also because in a vPC free scenario, the STP best practices would hold true.

A vPC has a domain that is an integer number that identifies a pair of switches. Layer 2 adjacent vPC pairs, must use different vPC domain numbers.

vPC requires two connections between the switches composing the pair: a vPC peer link and a vPC keep alive link. The peer link is the one where the CFS messages are passing and where the data will pass in case of a link failure of a vPC port channel. The keepalive link is a safety net used to avoid split brain scenarios and to bring up the vPC pair.

The vPC peer link must be a port channel composed of 10 Gbps interfaces, while the vPC keepalive link could be a normal 1 Gbps link. For maximum redundancy both links are usually made of port channels aggregating ports belonging to different modules, so that the failure of a line card wouldn’t bring down the whole port channel.

The vPC keepalive link is a layer 3 link, so it is best practice to put it into a separate VRF in order to protect it from routing issues that may happen in the network. Because the keepalive link is a layer 3 link, the keepalive traffic may be carried over an IP cloud: this is not the best practice because the devices should be directly connected to avoid being reliant on the transport network, but if this is not possible (Nexus 5000 without layer 3 module using the management interface connected to an out of band management IP network), then it is acceptable.

A feature called vPC peer gateway allows for a vPC switch to route traffic destined for the physical MAC address of an interface belonging to the other vPC switch. This traffic would be normally dropped due to vPC forwarding rules, but with this feature it will be forwarded. This feature is used when certain devices that are not completely compliant to the IP standard are in the network (a common example is a Netapp filer). When this feature is enabled and routing protocols adjacencies are built between the two vPC switches over the vPC peer link between SVI, those vlans should be excluded from the peer gateway feature due to very specific hardware limitations too complicated to be discussed here.

When vPC is used, STP configuration must be consistent between the two peers; else the in case of type 1 inconsistency vPC will be suspended or in case of type 2 inconsistency forwarding will be changed (for example by suspending vPC links on the secondary switch). Inconsistencies could be caused by global configuration or per interface configuration mismatches.

Type 1 inconsistencies are caused by:

- Mismatch in the VLAN to MST instance
- STP global settings mismatch (root guard, loop guard and bridge assurance)
- Per interface STP setting mismatch
- Switch port type (trunk vs. access)
- Port channel mode (active/passive/static)

Type 2 inconsistencies are caused by:

- QoS mismatches
- Mismatch in VLAN list configured on ports

As said before, even with vPC in place, STP is still needed as a safeguard mechanism in case of human error.

The best practices for STP do not differ when used in vPC vs. when used in normal scenarios.

- Spanning tree global options are risky and only bridge assurance should be used (only on the vPC peer link port).
- Ports that connect to hosts should be configured as edge if in access mode or edge trunk if in trunk mode.
- Ports connecting to other switches should be normal mode.
- UDLD could be used instead of bridge assurance.
- Root guard could be used on designated ports to prevent them from becoming root ports.
- BPDU guard should be enabled on host facing ports when edge trunk is configured to prevent connections of switches on ports that would go immediately in forwarding.
- All switches should be configured with explicit STP priorities, but vPC switches should use the same priority.

vPC configuration example on a vPC switch pair:

bru-dc1-n7k-1:

```
feature vpc
vrf context vPC_keepalive
interface Ethernet1/45
  description vPC keepalive interface
  vrf member vPC_keepalive
  ip address 10.0.0.1/30
  no shutdown
spanning-tree vlan 1-55,57-59,61-3967,4048-4093 priority 4096
spanning-tree vlan 56,60 priority 40960
vpc domain 10
  peer-switch
  peer-keepalive destination 10.0.0.2 source 10.0.0.1 vrf vPC_keepalive
  peer-gateway exclude-vlan 620,627
  ip arp synchronize
interface port-channel1000
  description vPC peer link
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-103,105,619-623,626-627
  switchport trunk allowed vlan add 666,843
  spanning-tree port type network
  vpc peer-link
```

bru-dc1-n7k-2:

```
feature vpc
vrf context vPC_keepalive
interface Ethernet1/45
  description vPC keepalive interface
  vrf member vPC_keepalive
  ip address 10.0.0.2/30
  no shutdown
spanning-tree vlan 1-55,57-59,61-3967,4048-4093 priority 4096
spanning-tree vlan 56,60 priority 40960
vpc domain 10
  peer-switch
  peer-keepalive destination 10.0.0.1 source 10.0.0.2 vrf vPC_keepalive
  peer-gateway exclude-vlan 620,627
  ip arp synchronize
interface port-channel1000
  description vPC peer link
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-103,105,619-623,626-627
  switchport trunk allowed vlan add 666,843
  spanning-tree port type network
  vpc peer-link
```

Example or STP configuration on different port type:

Port connected to a switch:

```
interface port-channel5
  description L2 port channel to the n5548, double sided vPC, mgmt traffic
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-104
  spanning-tree port type normal
  spanning-tree guard root
  vpc 5
```

Port connected to a host (or a UCS in end host mode):

```
interface port-channel1
  description Port channel to bru-dc1-fi-A, mgmt traffic
  switchport mode trunk
  vpc 1
  switchport trunk allowed vlan 56,60,100-106,208,619-621,627,666
  spanning-tree port type edge trunk
  spanning-tree bpduguard enable
```

Core/aggregation layer

In the example design I implemented the collapsed core aggregation layer with two Nexus 7000 in a vPC pair.

The layer connects to the access layer (Nexus 5000) via two sets of layer 2 port channels: the first set is for data traffic, the second is for inband management vlans that are used on the UCS servers to access the management infrastructure (DNS, vCenter, Active Directory and so on). The two sets of port channels are vPC port channels so every link is forwarding.

The core part of the layer connects to an IP cloud that is simulated using two 3750 (out of the scope for this design). The IP core is multicast enabled; we need this for OTV that is used to interconnect the two datacenters. The multicast RP is an anycast RP address configured on a loopback. Another loopback is used to allow for MSDP to propagate the (S, G) multicast groups between the two switches.

Connectivity with the IP cloud is done via an OSPF single area topology (for simplicity area 0), but a complete design would use eBGP, but eBGP was unusable due to limitation in the 3750 software, so OSPF was used instead. Loopbacks used for multicast and connected interfaces are distributed via those routing protocols as well.

The OTV router used to connect the two datacenters is a Virtual Device Context (VDC) installed on the same physical Nexus 7000 and connected to the primary VDC by loopbacked cables between linecards. To simulate that the OTV system may be a separate device, an EIGRP routing adjacency is brought up between the core and the OTV VDC.

This is the core part configuration of bru-dc1-n7k1:

```
vdc bru-dc1-n7k-1 id 1
  limit-resource module-type m1 f1 m1xl m2xl
  allocate interface Ethernet1/1-24,Ethernet1/28-48
  allocate interface Ethernet2/1-32
  allocate interface Ethernet3/1-28
vdc OTV id 2
  limit-resource module-type m1 f1 m1xl m2xl
  allocate interface Ethernet1/25-27
  boot-order 1

feature ospf
feature bgp
feature pim
feature pim6
feature msdp
feature eigrp
feature interface-vlan
feature hsrp
feature lacp
feature vpc
feature lldp
feature bfd

vrf context cloud
vrf context vPC_keepalive

ip pim rp 10.127.1.1
ip pim ssm range 232.0.0.0/8
ip msdp originator-id loopback1
ip msdp peer 200.200.200.1 connect-source loopback1

ip prefix-list ALL-NETWORKS seq 5 permit 0.0.0.0/0 le 32
route-map ALL-NETWORKS permit 10
  match ip address prefix-list ALL-NETWORKS

vpc domain 10
  peer-switch
  peer-keepalive destination 10.0.0.2 source 10.0.0.1 vrf vPC_keepalive
  peer-gateway exclude-vlan 620,627
  ip arp synchronize

interface port-channel10
  description L2 port channel to the OTV vdc of bru-dc1-n7k-1
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 622-623,627,666,843
  vpc 10

interface port-channel20
  description L2 port channel to the OTV vdc of bru-dc1-n7k-2
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 622-623,627,666,843
  vpc 20

interface Ethernet1/11
  description BRU-DC1-3750-1 gi 1/0/1 cloud link
  bfd interval 50 min_rx 100 multiplier 3
  no bfd echo
  vpc orphan-port suspend
  ip address 10.1.0.1/30
  ip router ospf cloud area 0.0.0.0
  ip pim sparse-mode
  ip igmp version 3
  no shutdown

interface Ethernet1/12
  description BRU-DC1-3750-1 gi 1/0/2 cloud link
  bfd interval 50 min_rx 100 multiplier 3
  no bfd echo
  vpc orphan-port suspend
  ip address 10.1.1.1/30
  ip router ospf cloud area 0.0.0.0
  ip pim sparse-mode
  ip igmp version 3
  no shutdown

interface Ethernet1/13
  description OTV L3 link eth 1/25
  vpc orphan-port suspend
  no ip redirects
  ip address 10.254.1.1/24
  ip router eigrp 1
  ip pim sparse-mode
  ip igmp version 3
  no shutdown

interface Ethernet1/14
  description Link to bru-dc1-n7k-2 OTV vdc eth1/26
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 622-623,627,666,843
  channel-group 20 mode active
  no shutdown

interface Ethernet1/15
  description Link to OTV vdc eth1/27
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 622-623,627,666,843
  channel-group 10 mode active
  no shutdown

interface Ethernet1/45
  description vPC keepalive interface
  vrf member vPC_keepalive
  ip address 10.0.0.1/30
  no shutdown

interface Ethernet2/9
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-103,105,619-623,626-627
  switchport trunk allowed vlan add 666,843
  channel-group 1000
  no shutdown

interface Ethernet2/10
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-103,105,619-623,626-627
  switchport trunk allowed vlan add 666,843
  channel-group 1000
  no shutdown

interface loopback0
  description Loopback for EIGRP
  ip address 10.10.10.1/32
  ip router eigrp 1
  ip pim sparse-mode

interface loopback1
  description Loopback for OSPF routing process for the cloud
  ip address 100.100.100.1/32
  ip router ospf cloud area 0.0.0.0
  ip pim sparse-mode
```

```
interface loopback2
  description mcast rp anycast
  ip address 10.127.1.1/32
  ip router ospf cloud area 0.0.0.0
  ip pim sparse-mode

router eigrp 1
  bfd
  autonomous-system 1
  router-id 10.10.10.1
  redistribute direct route-map ALL-NETWORKS
  redistribute ospf cloud route-map ALL-NETWORKS

router ospf cloud
  bfd
  router-id 100.100.100.1
  redistribute direct route-map ALL-NETWORKS
  redistribute eigrp 1 route-map ALL-NETWORKS
```

This is the aggregation part of the configuration of bru-dc1-n7k-1:

```
feature lacp
feature vpc

vlan 1,56,60,100-103,105,612,619-623,627,666,843,900

vrf context vPC_keepalive

vlan 56
  name BackBone
vlan 60
  name UC
vlan 100
  name n1kv_Control
vlan 101
  name VM_HA
vlan 102
  name N1kv_Packet
vlan 103
  name vMotion
vlan 105
  name VDI_Internal_Network
vlan 612
  name WAAS_interconnect
vlan 619
  name ACE_clients
vlan 620
  name vWAAS
vlan 621
  name WAAS_servers
vlan 622
  name vWAAS_vpath
vlan 623
  name vWAAS_vpath_servers
vlan 627
  name ACE_servers
```

```
vlan 666
  name VM_test
vlan 843
  name OTV_site-vlan
vlan 900
  name TEST

spanning-tree vlan 1-55,57-59,61-3967,4048-4093 priority 4096
spanning-tree vlan 56,60 priority 40960

vpc domain 10
  peer-switch
  peer-keepalive destination 10.0.0.2 source 10.0.0.1 vrf vPC_keepalive
  peer-gateway exclude-vlan 620,627
  ip arp synchronize

interface Vlan1

interface Vlan56
  no shutdown
  ipv6 address 2001:420:44ff:ff38:1:38:0:1/64

interface Vlan105
  description VDI Desktop vlan
  no shutdown
  ipv6 address 2001:420:44ff:ff80:1:69:0:1/64

interface port-channel5
  description L2 port channel to the n5548, double sided vPC, mgmt traffic
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-104
  spanning-tree port type normal
  spanning-tree guard root
  vpc 5

interface port-channel6
  description L2 port channel to the n5548, double sided vPC, data traffic
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 105,619-621,627,666
  spanning-tree port type normal
  spanning-tree guard root
  vpc 6

interface port-channel1000
  description vPC peer link
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-103,105,619-623,626-627
  switchport trunk allowed vlan add 666,843
  spanning-tree port type network
  vpc peer-link

interface Ethernet1/45
  description vPC keepalive interface
  vrf member vPC_keepalive
  ip address 10.0.0.1/30
  no shutdown

interface Ethernet2/9
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-103,105,619-623,626-627
  switchport trunk allowed vlan add 666,843
  channel-group 1000
  no shutdown

interface Ethernet2/10
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-103,105,619-623,626-627
  switchport trunk allowed vlan add 666,843
  channel-group 1000
  no shutdown

interface Ethernet2/17
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-104
  channel-group 5 mode active
  no shutdown

interface Ethernet2/18
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 105,619-621,627,666
  channel-group 6 mode active
  no shutdown

interface Ethernet2/25
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-104
  channel-group 5 mode active
  no shutdown

interface Ethernet2/26
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 105,619-621,627,666
  channel-group 6 mode active
  no shutdown
```

Access layer

The access layer is composed of two Nexus 5548 switches in vPC. Compared to the core/aggregation layer, the access layer is way simpler. It is used for the UCS and other end host devices to connect to the rest of the network in a pure layer 2 say, default gateway functions are delegated to the aggregation layer and achieved via SVI with first hop redundancy protocol (FHRP) features like HSRP.

In this case the access layer is also an access layer for FC traffic. The USC uses FCoE to the Nexus 5548 and then the Nexus translates it to FC towards the storage arrays.

Regarding the best practices and implementation details, same rules from the aggregation layer apply, with a small difference concerning the vPC keepalive link: as the Nexus 5548 by default doesn't support layer 3 features (if the layer 3 module is not installed), the only usable layer 3 interface is the mgmt0 management interface; to have then the switch reachable from the external world and able to exchange keepalives towards another Nexus 5548, the mgmt0 interface has to be connected to an out of band network.

Another possibility is to use layer 2 SVI interfaces keeping in mind that a major STP issue will impact those interfaces as well.

In our example, we have the layer 3 modules installed into the Nexus 5548 so we use a single interface (due to lack of 1 Gbps ports) for keepalive exchange. As in the previous chapters, this is not the optimal solution as a port channel of multiple interfaces would be preferred, but due to restrictions, we couldn't configure it.

The Ethernet configuration on this layer is very similar to the aggregation one with some minor changes regarding the access ports directly connected to an UCS or a host device.

```
interface Ethernet1/17
  description Port channel to bru-dc1-fi-A, mgmt traffic
  switchport mode trunk
  switchport trunk allowed vlan 56,60,100-106,208,619-621,627,666
  spanning-tree port type edge
  spanning-tree bpduguard enable
  channel-group 1 mode active

interface Ethernet1/18
  description Port channel to bru-dc1-fi-A, data traffic
  switchport mode trunk
  switchport trunk allowed vlan 105,666
  spanning-tree port type edge
  spanning-tree bpduguard enable
  channel-group 2 mode active
```

These are two ports part of two vPC port channels connected to the UCS. As we can see, the ports are configured as edge trunks, so the ports will go in forwarding state immediately without passing through the learning state of STP. The BPDU guard feature enabled will force the port to shut down in case of a BPDU received on that port, this is to prevent a switch from being attached there and cause a loop. As the UCS is a switch that by default acts as a host, but can be enabled to run STP as a normal switch, these features are there to prevent a possible issue.

Server connectivity

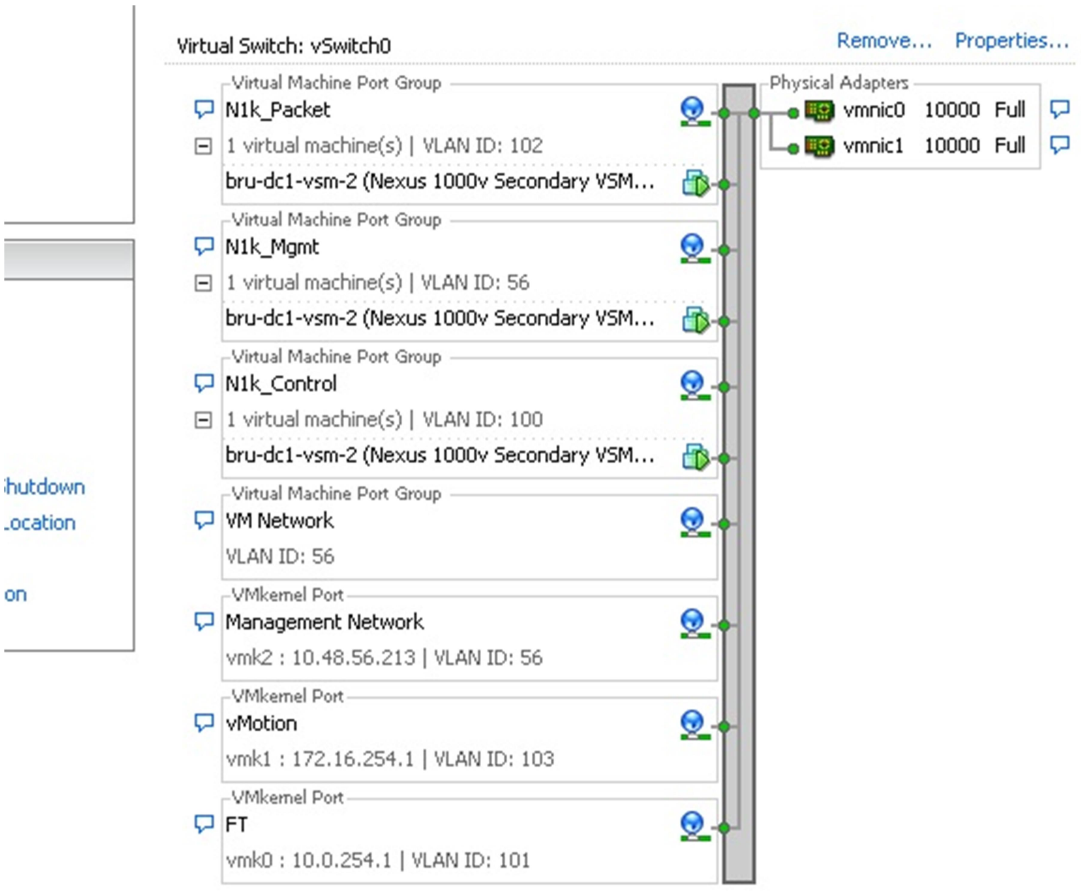
The fabric interconnects are running in end host mode and are connected to the Nexus 5000 vPC pair via two vPC port channels, one carries data vlans and the other inband management vlans and other management-like vlans (as the Nexus 1000v control and packet vlans). Because the UCS in end host mode works differently than a switch and the ports have restricted configuration possibilities, there are no interconnections between the two FI switches (with the exception of the links used to form a cluster), so to create a redundant environment, every vlan that is used on servers hosted in different physical chassis, must be carried over towards the access layer.

The servers are physically hosted in chassis (maximum of 8 half width blades or 4 full widths). The chassis are connected to the Fabric Interconnect via twinax cables from the IO Modules. These cables carry both Ethernet and FCoE. The links are carrying all vlans, there is no vlan pruning possibility on these links and to perform a sort of pruning, virtual NIC are created and bound to IOM. In the version used, the IOM links towards the FI are statically pinned to a server, so if a link goes down, the associated servers will also see a NIC down event (if no failover to the other IOM is configured). In newer versions of the software, it is possible to bundle IOM links into a port channel and pin the servers to the port channel interface itself and inherit the load balancing and reliability features of the bundle.

Virtual NIC

The servers will be configured to have multiple virtual NIC in order to assign them to different virtual switches inside ESXi [34]. We do this to separate data from management traffic, if all vlans would be trunked into the same vNIC handled by the same vSwitch, a malfunction of the vSwitch would prevent us from accessing the management of the ESXi and force us to go through out of band connectivity (that may not be available) [35].

This explains how the standard ESXi virtual switch is configured:



Virtual Switch: vSwitch0 Remove... Properties...

Virtual Machine Port Group
N1k_Packet
1 virtual machine(s) | VLAN ID: 102
bru-dc1-vsm-2 (Nexus 1000v Secondary VSM...

Virtual Machine Port Group
N1k_Mgmt
1 virtual machine(s) | VLAN ID: 56
bru-dc1-vsm-2 (Nexus 1000v Secondary VSM...

Virtual Machine Port Group
N1k_Control
1 virtual machine(s) | VLAN ID: 100
bru-dc1-vsm-2 (Nexus 1000v Secondary VSM...

Virtual Machine Port Group
VM Network
VLAN ID: 56

VMkernel Port
Management Network
vmk2 : 10.48.56.213 | VLAN ID: 56

VMkernel Port
vMotion
vmk1 : 172.16.254.1 | VLAN ID: 103

VMkernel Port
FT
vmk0 : 10.0.254.1 | VLAN ID: 101

Physical Adapters
vmnic0 10000 Full
vmnic1 10000 Full

Figure 53: management vlans on servers

There are 2 vNIC bundled together to allow redundancy (vmnic0 and vmnic1). Those 2 vNIC are pinned to links on different IOM.

Distributed virtual switches

The data vlans are carried over 2 other vNIC (again bundled together in a port channel for redundancy) pinned to different IOM links.

ESXi Management vlans can also be carried over this link if the virtual machines need to be in the same layer 2 domain as the ESXi server for example if vCenter is hosted as a VM on top of an ESXi that it manages (this is not recommended).

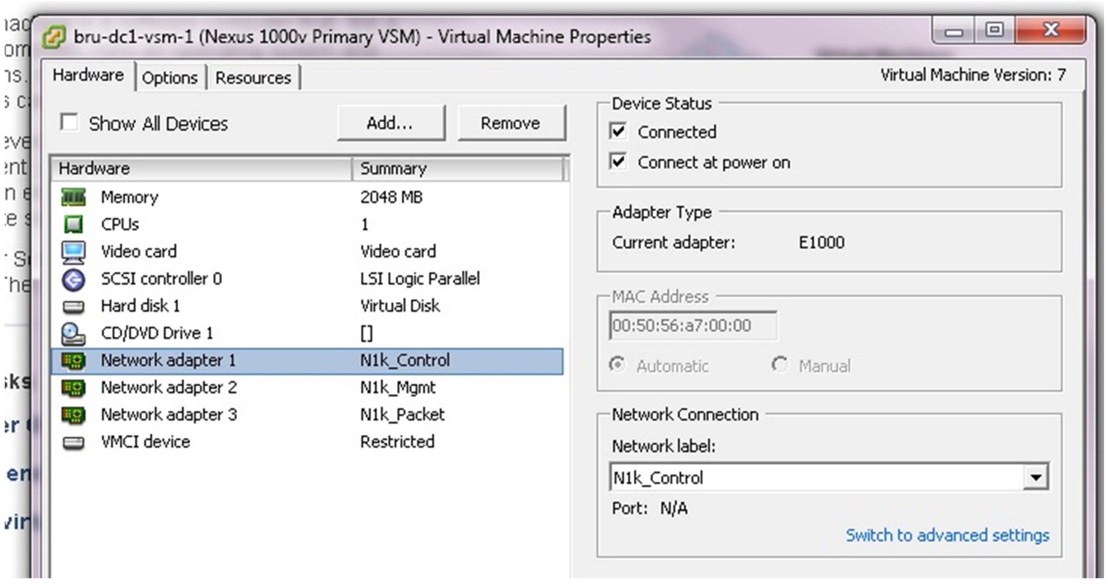
The virtual machines interfaces are all allocated into the Nexus 1000v that is a distributed virtual switch running as a kernel module into ESXi giving advanced networking features to the host OS. The switch is called distributed because every ESXi where the Nexus 1000v agent is installed (called a VEM) is able to communicate with a central control module (called a VSM) that configures it. All VEM share state information with the VSM and get programmed by it, so that the state of the network is distributed and synchronized among the various agents. To make a parallel with a physical switch, the VSM would be a supervisor engine and the VEM a linecard.

The VSM and the VEM need three vlans to correctly communicate: the management, the packet and the control vlan.

The control vlan is used to establish communication and synchronize VEM and VSM status and configuration, the packet vlan is used to punt particular packets to the VSM (as LLDP, IGMP and CDP) and the management vlan is used for the VSM to communicate to vCenter.

For the VSM-VEM cluster to work, layer 2 reachability between the VEM and the VSM via the control and the packet vlans must be available. More recent versions of the software allow for layer 3 communication between VSM and vCenter.

A VSM has three NIC configured as follows (order is important):


bru-dc1-vsm-1 (Nexus 1000v Primary VSM) - Virtual Machine Properties

Virtual Machine Version: 7

Hardware	Summary
Memory	2048 MB
CPUs	1
Video card	Video card
SCSI controller 0	LSI Logic Parallel
Hard disk 1	Virtual Disk
CD/DVD Drive 1	[]
Network adapter 1	N1k_Control
Network adapter 2	N1k_Mgmt
Network adapter 3	N1k_Packet
VMCI device	Restricted

Device Status
☑ Connected
☑ Connect at power on

Adapter Type
Current adapter: E1000

MAC Address
00:50:56:a7:00:00
(●) Automatic ( ) Manual

Network Connection
Network label: N1k_Control
Port: N/A

Figure 54: VSM NIC configuration

Once a Nexus 1000v is connected to vCenter, every port profile configured in the VSM will be reflected in vCenter with a port group.

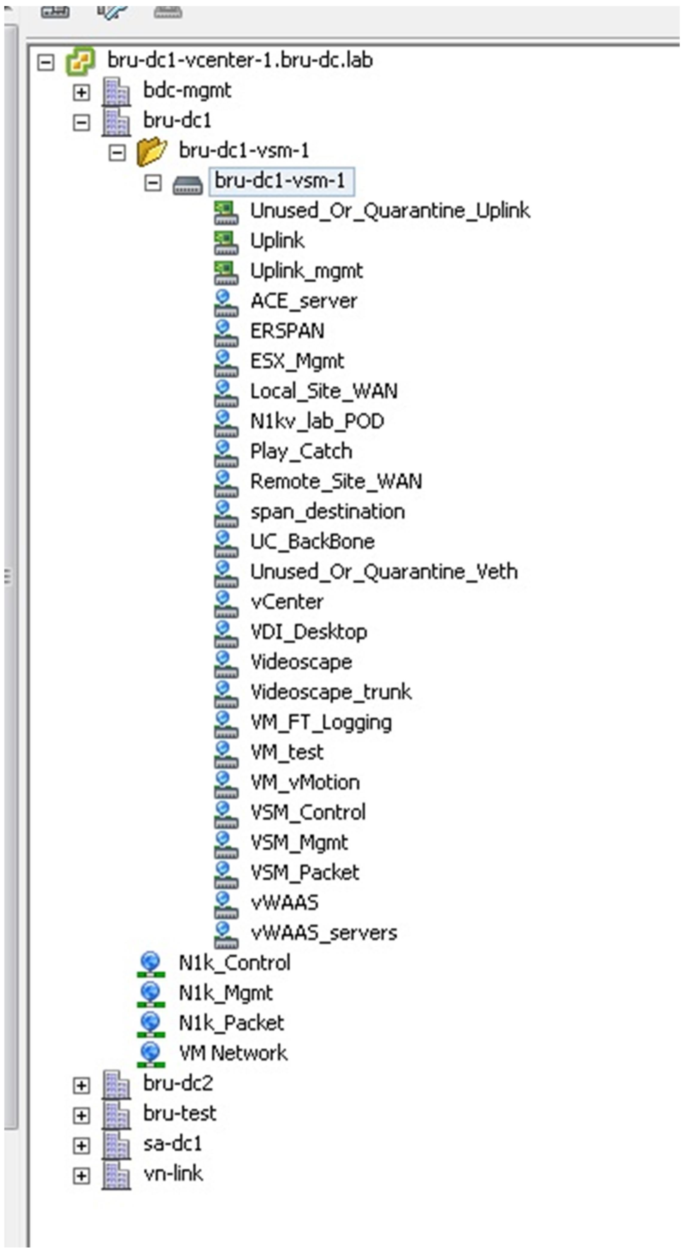


Figure 55: vCenter port groups from VSM configuration

The VM will have their interfaces assigned to one of the port groups displayed above. In case of a VM due to a restart or a vMotion, the Nexus 1000v will take care of moving the MAC address from the VEM where the VM used to reside to the new one.

Also all port profile changes in the Nexus 1000v will be distributed to all ESXi in the cluster, this is especially useful when the cluster is expanded as new ESXi servers are able to download the configuration and be ready to host the VM with no network interruption due to a missing part of the configuration.

The port profiles are configured as either a type vEthernet or a type Ethernet in the VSM. The vEthernet type is used for VM, while the Ethernet is used to configure the physical interfaces that are used for ESXi to communicate with the rest of the world. It doesn't matter if the interfaces are physical (as in a ESXi installed on a rack server) or virtual (as a vNIC configured on the UCS), for the VSM/VEM and vCenter, those interfaces are always seen as physical.

Once VM are assigned to port groups in vCenter, the VSM will show a virtual interface that can be used to monitor the network performances of the VM. It is best practice to not change the configuration of the vEthernet interface assigned to the VM as changes will override the inherited port profile configuration and the vEthernet interface will not move with the VM (a new vEthernet interface may be assigned): instead it is better to create a new port profile, inherit the original there and make the changes (e.g. configure QoS or Netflow) and then assign this new vEthernet port profile to the VM.

To connect the VSM to vCenter in layer 2 mode, the following configuration is needed:

```
vlan 56
  name BackBone
vlan 100
  name N1kv_Control
vlan 102
  name N1kv_Packet


svs-domain
  domain id 10
  control vlan 100
  packet vlan 102
  svs mode L2
svs connection vcenter
  protocol vmware-vim
  remote ip address 10.48.56.225 port 80
  vmware dvs uuid "7e 9d 0a 50 0b 8b 39 78-74 b1 2e 4b 7c 81 86 ca" datacenter-
name bru-dc1
  connect
```

In the "svs-domain" configuration the Domain ID is a number to identify a cluster of VEM/VSM.

In the "svs connection vcenter" configuration we need to specify the IP address or name of the vCenter server.

The connectivity between VM and the rest of the world will be established with the following commands:

```
vlan 105
  name VDI_Internal_Network
```

```
vlan 107
  name N1kv_lab_POD
vlan 208
  name videoscape
vlan 612
  name WAAS_interconnect
vlan 619
  name ACE_clients
vlan 620
  name vWAAS
vlan 621
  name WAAS_servers
vlan 627
  name ACE_servers
vlan 666
  name VM_test
vlan 2500
  name local_site_WAN
vlan 2501
  name remote_site_WAN
port-profile type ethernet Uplink
  vmware port-group
  switchport mode trunk
  switchport trunk allowed vlan 105,208,612,619-621,627,666
  mtu 9000
  channel-group auto mode on mac-pinning
  no shutdown
  description Generic Uplink port-profile
  state enabled
interface Ethernet4/5
  inherit port-profile Uplink
  no shutdown

interface Ethernet4/6
  inherit port-profile Uplink
  no shutdown
port-profile type vethernet VDI_Desktop
  vmware port-group
  switchport access vlan 105
  switchport mode access
  ip flow monitor IP_monitoring output
  ip flow monitor IP_monitoring input
  no shutdown
  max-ports 50
  description Internal VDI Network
  state enabled



interface Vethernet11
  inherit port-profile VDI_Desktop
  description VDI-4,Network Adapter 1
  no shutdown
  vmware dvport 992 dvswitch uuid "7e 9d 0a 50 0b 8b 39 78-74 b1 2e 4b 7c 81 86
ca"
  vmware vm mac 0050.568A.0012

interface Vethernet12
  inherit port-profile VDI_Desktop
  description Visual Studio Test,Network Adapter 1
  vmware dvport 988 dvswitch uuid "7e 9d 0a 50 0b 8b 39 78-74 b1 2e 4b 7c 81 86
ca"
  vmware vm mac 0050.568A.0000
```

After defining the vlans and the port profiles for the uplink (that will carry all the data vlans), we create a vEthernet port profile that will define the network configuration of the VM NIC.

The line “channel-group auto mode on mac-pinning” will take care of bundling automatically all interfaces assigned to that port profile.

When the VMs are configured to use those port groups in vCenter, the VSM will create a virtual interface (vEthernet 11 and 12 in this case) for each VM NIC assigned to that profile. Information on the VM (their MAC address and vCenter CM unique identifier) is added automatically.

Access to internet (WAN) and OTV

A datacenter needs to have access to the internet; for this example, we’ve simulated an IP cloud with a pair of 3750 switches.

Each switch of the pair is inter-connected and attached to the corresponding Nexus pair on each DC site via two layer 3 interfaces point to point interfaces in order to create some possibility for equal cost multipathing (ECMP).

In a real scenario, the Nexus devices would be connected to the internet usually via a different device acting as a Data Center Interconnect (DCI). This device is usually a router that is able to host the whole BGP table, has MPLS capabilities, advanced QoS features and bigger sized buffers.

A Nexus would connect to that device and exchange routes usually via an IGP as OSPF or less commonly ISIS or with an eBGP peering. In this case, due to lack of features in the 3750, the Nexus will use OSPF to exchange routing information with the 3750. Only area 0 is used.

Because we require layer 2 extension between the two sites, we need to use a technique that allows some kind of layer 2 encapsulation over an IP cloud as VPLS because we don’t want to extend the vlans over the internet, but we want to use transit services offered by an ISP. Also extending layer 2 over a WAN will mean extending the STP failure domain to multiple sites and we don’t want TCN generated in one site to cause unicast flooding over to the other site. Technologies like VPLS also require a full mesh between all sites that can be expensive and difficult to manage.

In this example we used OTV to extend the layer 3 domain in a scalable way. An OTV VDC is created and is connected to the core/aggregation VDC via a layer 2 vPC port channel that carries the data to encapsulate and a layer 3 interface that is used for the OTV VDC to join a control multicast group that is used for all the sites to establish an ISIS adjacency and announce information about the MAC addresses of the sites. Layer 2 connections between OTV VDC and core/aggregation VDC are needed due to limitations that don't allow the OTV context to perform routing for the vlans that it extends.

This is the configuration that is needed for OTV to start communicating with the other sites. For simplicity we've created an EIGRP routing adjacency between the core/aggregation and the OTV VDC for exchanging routing information (mostly PIM RP address), but this is facultative.

```
feature pim

interface Overlay1
  otv join-interface Ethernet1/25
  otv control-group 239.1.1.2
  otv data-group 239.192.1.0/24
  otv extend-vlan 627, 666
  no otv suppress-arp-nd
  no shutdown

interface Ethernet1/25
  description OTV L3 interface main VDC eth 1/13
  mtu 9216
  ip address 10.254.1.254/24
  ip router eigrp 1
  ip igmp version 3
  no shutdown

interface loopback0
  ip address 10.10.10.253/32
  ip router eigrp 1

router eigrp 1
  autonomous-system 1
  router-id 10.10.10.253
otv-isis default
otv site-identifier 0x1
```

On the core/aggregation VDC this is the configuration needed:

```
feature pim
feature eigrp

ip pim rp 10.127.1.1
ip pim ssm range 232.0.0.0/8
ip msdp originator-id loopback1
ip msdp peer 200.200.200.1 connect-source loopback1

interface Ethernet1/13
  description OTV L3 link eth 1/25
  vpc orphan-port suspend
  no ip redirects
  ip address 10.254.1.1/24
  ip router eigrp 1
  ip pim sparse-mode
  ip igmp version 3
  no shutdown
```

```
interface loopback0
  description Loopback for EIGRP
  ip address 10.10.10.1/32
  ip router eigrp 1
  ip pim sparse-mode

interface loopback1
  description Loopback for OSPF routing process for the cloud
  ip address 100.100.100.1/32
  ip router ospf cloud area 0.0.0.0
  ip pim sparse-mode

interface loopback2
  description mcast rp anycast
  ip address 10.127.1.1/32
  ip router ospf cloud area 0.0.0.0
  ip pim sparse-mode

router eigrp 1
  bfd
  autonomous-system 1
  router-id 10.10.10.1
  redistribute direct route-map ALL-NETWORKS
  redistribute ospf cloud route-map ALL-NETWORKS
```

The configuration is simple, the only remark is that the IP cloud must be multicast enabled and you need to have allocated multicast addresses for control and data groups.

If multiple OTV contexts are hosted on the same site they will elect an Authoritative Edge Device (AED) that will become the “encapsulator” for a certain vlans. Vlans will be split equally among peers.

When a new MAC address will come online and is learned by the OTV device, it will be announced via ISIS to the other sites over the control plane multicast group to avoid sending unicast notification. The OTV data group will be used instead, similarly to a MVPN, by mapping multicast groups with receivers are in different sites to one data group so that only the sites where receivers are present will join that data group and receive the multicast that will then be decapsulated and forwarded in the local network. While this seems superfluous, it is important in order to:

- Keep the number of the multicast entries small
- Offload the task of the multicast replication to the provider network instead of the Nexus

Security

Securing the datacenter means securing the access from the internet towards the servers and securing the elements of the datacenter themselves.

To secure the traffic coming from the internet the simplest solution is to install a stateful firewall as a service into the aggregation layer. The firewall will be a default gateway for the servers that needs to be protected and will route packets through between the vlans assigned to the different servers/applications and the internet. The choice of using a firewall as default gateway is the simplest solution, but another possibility is to use a default gateway on the access or aggregation switches and then establish a routing adjacency with the firewall and threat it as an element “on a stick” so traffic will be sent to the firewall, checked and routed out towards the aggregation switch and then routed out of the switch towards the destination. For this to work, the switch will have to put the networks connecting to the firewall in different VRF or else the routing will happen on the switch and the firewall will be bypassed.

A layer 2 firewall (like a transparent bump in the wire firewall) is not really a possibility due to the interaction with STP and the limitation that this kind of configuration poses.

In this example unfortunately there is no firewall due to lack of devices.

To protect the devices themselves possibilities are various and all should be implemented:

- Control plane policing protects against attacks targeted to overload the CPU of the switch like pings, fragmented packets or TCP SYN flooding.
- AAA controls which users can access the router, which operations can perform and logs every command issued
- Access lists limit the access to the switch from a well-known group of IP
- Out of band management blocks any possible attack from the outside world because there is no connectivity between internet and the management access of the devices
- SNMP access lists, community and v3 users restrict the IP addresses that can poll the switch via SNMP and encrypt the SNMP traffic
- Logging adds an additional level of security by storing additional information on the status of the switch and who accesses it
- Additional features as Cisco Trust Sec (CTS) are able to encrypt at line rate speed the traffic to prevent sniffing via TAP devices.

Not all features were available at the time of the design of the example.

This is an example on how to configure AAA servers:

```
feature radius
radius distribute
radius-server retransmit 5
```

```
radius-server host bru-dc1-ise-1.cisco.com key 7 "Lxq-levr1" authentication
accounting
radius commit
radius-server host bru-dc1-ise-1.cisco.com test username cisco password cisco
idle-time 3
aaa group server radius ISE
    server bru-dc1-ise-1.cisco.com
    deadtime 3
    use-vrf management
    source-interface mgmt0
aaa authentication login default group ISE
aaa accounting default group ISE
vrf context management
  ip domain-name cisco.com
  ip name-server 10.48.56.249 144.254.71.184
  ip route 0.0.0.0/0 10.48.56.1
```

This is how to configure command authorization when this feature is not available or implemented with limitation in the AAA server:

```
role distribute
role name WAAS_user
  description Role for vWAAS users
  rule 34 permit command copp copy *
  rule 33 permit command conf t ; policy-map *
  rule 32 permit command conf t ; class-map *
  rule 31 permit command copp *
  rule 30 permit command ethanalyzer *
  rule 29 permit command conf t ; no ip router *
  rule 28 deny command conf t ; no ip router eigrp 1
  rule 27 permit command conf t ; router * ; *
  rule 26 deny command conf t ; router eigrp 1
  rule 25 deny command conf t ; no ip access-list copp*
  rule 24 permit command conf t ; vrf context vWAAS ; *
  rule 23 permit command copy run start vdc
  rule 22 permit command conf t ; no ip access-list *
  rule 21 permit command conf t ; ip access-list * ; *
  rule 20 deny command conf t ; no ip access-list copp*
  rule 19 deny command conf t ; ip access-list copp*
  rule 18 permit command conf t ; no interface vlan *
  rule 17 permit command conf t
  rule 16 permit command config t ; interface * ; *
  rule 15 permit read-write feature wccp
  rule 14 permit command traceroute6 *
  rule 13 permit command traceroute *
  rule 12 permit command telnet6 *
  rule 11 permit command telnet *
  rule 10 permit command ping6 *
  rule 9 permit command ping *
  rule 8 permit command ssh6 *
  rule 7 permit command ssh *
  rule 6 permit command enable *
  rule 5 permit command no debug logfile *
  rule 4 permit command undebug *
  rule 3 permit command debug *
  rule 2 permit command terminal *
  rule 1 permit read
  vlan policy deny
    permit vlan 612-612
    permit vlan 620-623
  interface policy deny
    permit interface Vlan612
    permit interface Vlan620
    permit interface Vlan621
    permit interface Vlan622
    permit interface Vlan623
    permit interface port-channel6
    permit interface Ethernet1/1
  vrf policy deny
    permit vrf vWAAS
role name read-only
  description Role for read only users
  rule 19 permit command switchback *
  rule 18 permit command switchto vdc *
  rule 17 permit command switchto *
  rule 16 permit command slot *
  rule 15 permit command show *
  rule 14 permit command traceroute6 *
  rule 13 permit command traceroute *
  rule 12 permit command telnet6 *
  rule 11 permit command telnet *
  rule 10 permit command ping6 *
  rule 9 permit command ping *
  rule 8 permit command ssh6 *
  rule 7 permit command ssh *
  rule 6 permit command enable *
  rule 2 permit command terminal *
  rule 1 permit read
role commit
username waas password 5 $1$jZ/hHoM8$//YjZnb5L1KN52RQncxJJ1  role WAAS_user
```

Limiting access to the management of the switch can be done like this:

```
ip access-list mgmt-access
permit tcp 10.48.56.0/24 any eq 22
permit udp 10.48.56.0/24 any eq 161
line vty
ip access-class mgmt-access
```

SNMP can be protected via access lists and v3 usernames:

```
ip access-list mgmt-access
permit tcp 10.48.56.0/24 any eq 22
permit udp 10.48.56.0/24 any eq 161
snmp-server user ACE vdc-admin auth md5 0x51abad087a67f75c8cedaae25c4bd1e9 priv
0x51abad087a67f75c8cedaae25c4bd1e9 localizedkey
snmp-server user waas network-operator auth md5
0x488de185dda3d5f2b98b5dfa06c2834a priv 0x488de185dda3d5f2b98b5dfa06c2834a
localizedkey
snmp-server user waas WAAS_user
```

```
snmp-server user admin network-admin auth md5 0xbe163cbc5f6680325240f776645b8dbf
localizedkey
snmp-server user cisco read-only auth md5 0xbca9b64384472edbf90e588fd266b40e priv
0xbca9b64384472edbf90e588fd266b40e localizedkey
snmp-server host 172.22.29.51 traps version 2c public  udp-port 2162
snmp-server community public use-acl mgmt-access
```

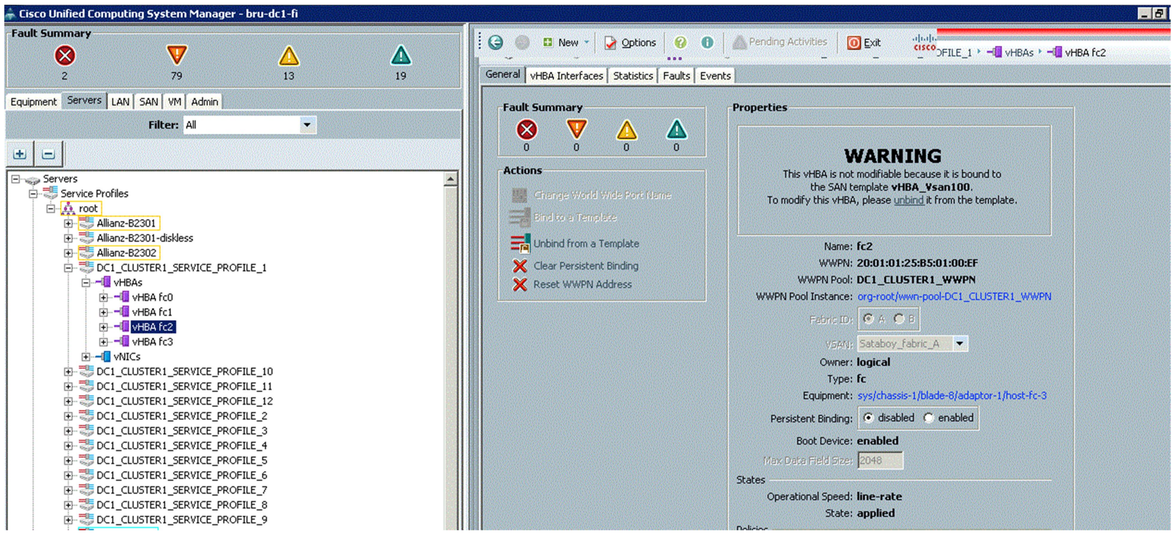
Logging:

```
logging level aaa 5
logging level cdp 6
logging level glbp 6
logging level hsrp 6
logging level interface-vlan 5
logging level lldp 5
logging level monitor 6
logging level otm 6
logging level port-channel 6
logging level radius 5
logging level snmpd 7
logging level spanning-tree 6
logging level tacacs 5
logging logfile messages 6
logging server 10.48.56.53 5 use-vrf management
logging timestamp milliseconds
```

Control plane configuration has been shown in the “General Best Practices” chapter.

Storage connectivity

Similarly as the Ethernet connectivity, the servers hosted in the UCS chassis are configured to have four vHBA each: two connected to fabric A (vsan 100, FCoE vlan 1100) and two connected to fabric B (vsan 200, FCoE vlan 1200).


Cisco Unified Computing System Manager - bru-dc1-fi
Fault Summary
2
79
13
19
Equipment
Servers
LAN
SAN
VM
Admin
Filter: All
Servers
Service Profiles
root
Allianz-B2301
Allianz-B2301-diskless
Allianz-B2302
DC1_CLUSTER1_SERVICE_PROFILE_1
vHBAs
vHBA fc0
vHBA fc1
vHBA fc2
vHBA fc3
vNICs
DC1_CLUSTER1_SERVICE_PROFILE_10
DC1_CLUSTER1_SERVICE_PROFILE_11
DC1_CLUSTER1_SERVICE_PROFILE_12
DC1_CLUSTER1_SERVICE_PROFILE_2
DC1_CLUSTER1_SERVICE_PROFILE_3
DC1_CLUSTER1_SERVICE_PROFILE_4
DC1_CLUSTER1_SERVICE_PROFILE_5
DC1_CLUSTER1_SERVICE_PROFILE_6
DC1_CLUSTER1_SERVICE_PROFILE_7
DC1_CLUSTER1_SERVICE_PROFILE_8
DC1_CLUSTER1_SERVICE_PROFILE_9
New
Options
Pending Activities
Exit
OFILE_1 vHBAs vHBA fc2
General
vHBA Interfaces
Statistics
Faults
Events
Fault Summary
0
0
0
0
Actions
Change World Wide Port Name
Bind to a Template
Unbind from a Template
Clear Persistent Binding
Reset WWPN Address
Properties
States
Policies

WARNING
This vHBA is not modifiable because it is bound to the SAN template **vHBA_Vsan100**.
To modify this vHBA, please unbind it from the template.

Name: **fc2**
WWPN: **20:01:01:25:B5:01:00:EF**
WWPN Pool: **DC1_CLUSTER1_WWPN**
WWPN Pool Instance: org-root/wwn-pool-DC1_CLUSTER1_WWPN
Fabric ID: [✓ A] B
VSAN: Sataboy_fabric_A
Owner: **logical**
Type: **fc**
Equipment: sys/chassis-1/blade-8/adaptor-1/host-fc-3
Persistent Binding: [✓ disabled] enabled
Boot Device: **enabled**
Max Data Field Size: 2048
Operational Speed: **line-rate**
State: **applied**

Figure 56: vHBA example

The interfaces are called fc0, fc1, fc2, fc3.

WWPN and WWNN are chosen from a PWWN pool defined in the UCSM. The WWN contains the DC number, other information may be added in the future.

The current WWPN pools are:

from 20:(DC number):01:25:BB:01:00:00 to 20:(DC number):01:25:BB:01:00:FE:FF

Each server currently has 4 vHBA: 2 are used to connect to a SATAboy (storage array) dual fabric and another 2 to the NetApp (decommissioned at the time of writing).

The principal switch for fabric A is bru-dc1-9148-1 and the principal switch for fabric 2 is bru-dc2-9148-1.

Zoning is done on the principal switches by using enhanced device aliases and zoning a single port of an array with a single port of a server. Example:

```
device-alias mode enhanced
device-alias database
  device-alias name bru-dc1-storage-1-1 pwwn 50:00:40:20:01:f4:40:0d
  device-alias name bru-dc1-cluster1-server1-fc2 pwwn 20:01:01:25:b5:01:00:ef
  <omit>

zone name bru-dc1-cluster1-server1-fc2_bru-dc1-storage-1-1 vsan 100
    member device-alias bru-dc1-cluster1-server1-fc2
    member device-alias bru-dc1-storage-1-1

<omit>
```

```
zoneset name ucs_storage vsan 100
    member bru-dc1-cluster1-server1-fc2_bru-dc1-storage-1-1
   <omit>

zoneset activate name ucs_storage vsan 100
```

Zoning is distributed via CFS to the rest of the fabric.

On the SATAboy we did LUN masking in order to allow certain servers' PWWN to login and have read write privileged for certain LUN and hide some more important LUNs (such as the ESXi boot LUNs) from being shared among multiple servers. Only the data LUNs where the VMs are stored have to be shared because shared storage is a prerequisite for vMotion.

The FI are configured in NPV mode and have an uplink (FI 1 in Fabric A and FI 2 in Fabric B) to the 9148 that acts as NPIV switches and DR for the fabrics.

Traffic flows in FCoE from the servers to the FI where it gets decapsulated and put into native FC towards the MDS 9148 [36] where the storage arrays connect.

A standalone rack server is also connected to the Nexus 5548 via FCoE. The Nexus 5548 is decapsulating the traffic from FCoE to FC and forwarding it to the MDS 9148 to reach the storage.

This is the configuration that is needed for a server to connect via FCoE to a Nexus 5548:

```
feature fcoe
vlan 1100
  fcoe vsan 100
vsan database
  vsan 100
interface vfc29
  bind interface Ethernet1/29
  switchport trunk allowed vsan 100
interface Ethernet1/29
  switchport mode trunk
  switchport trunk allowed vlan 1,56,101-106,1100
  spanning-tree port type edge trunk
```

Note as the port 1/29 carries both the FCoE vlan and the data vlans (pure Ethernet).

For sake of demonstration, an FCoE trunk is configured between the Nexus 5548 and the Nexus 7000 on a storage type VDC. Again we split the trunks between bru-dc1-n5548-1 and bru-dc1-n7k-1 in Fabric A and between bru-dc1-n5548-2 and bru-dc1-n7k-2 in Fabric B.

For maximum redundancy, the vFC interfaces are bound to a port channel rather than standard links.

To configure an FCoE trunk between a Nexus 7000 and a Nexus 5548, this is the configuration needed:

Nexus 5548:

```
feature fcoe
feature fport-channel-trunk
feature lacp
feature lldp
system qos
  service-policy type queuing input fcoe-default-in-policy
  service-policy type queuing output fcoe-default-out-policy
  service-policy type qos input fcoe-default-in-policy
  service-policy type network-qos fcoe-default-nq-policy
vlan 1100
  fcoe vsan 100
port-channel load-balance ethernet source-dest-port
vsan database
  vsan 100
interface port-channel100
  description Port channel for multihop FCoE to bru-dc1-n7k-1
  switchport mode trunk
  switchport trunk allowed vlan 1100
interface vfc100
  bind interface port-channel100
  switchport mode E
  switchport trunk allowed vsan 100
  no shutdown
vsan database
  vsan 100 interface fc2/1
  vsan 100 interface fc2/2
interface Ethernet1/31
  switchport mode trunk
  switchport trunk allowed vlan 1100
  channel-group 100 mode active

interface Ethernet1/32
  switchport mode trunk
  switchport trunk allowed vlan 1100
  channel-group 100 mode active
```

Nexus 7000 Admin VDC:

```
vdc fcoe id 3 type storage
  limit-resource module-type f1
  allow feature-set fcoe
  allocate interface Ethernet3/29-32
  boot-order 1
  limit-resource vlan minimum 16 maximum 4094
  limit-resource monitor-session minimum 0 maximum 2
  limit-resource monitor-session-erspan-dst minimum 0 maximum 23
  limit-resource vrf minimum 2 maximum 4096
  limit-resource port-channel minimum 0 maximum 768
  limit-resource u4route-mem minimum 8 maximum 8
  limit-resource u6route-mem minimum 4 maximum 4
  limit-resource m4route-mem minimum 8 maximum 8
  limit-resource m6route-mem minimum 5 maximum 5
  limit-resource monitor-session-inband-src minimum 0 maximum 1
feature lacp
feature lldp
```

Nexus 7000 Storage VDC:

```
feature-set fcoe
hostname fcoe
feature fport-channel-trunk
feature lacp
feature lldp
vlan 1,1100
vlan 1100
  fcoe vsan 100
vsan database
  vsan 100
device-alias mode enhanced
device-alias commit

interface port-channel100
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 1100


interface vfc-po100
  bind interface port-channel100
  switchport mode E
  switchport trunk allowed vsan 100

interface Ethernet3/31
  switchport mode trunk
  switchport trunk allowed vlan 1100
  channel-group 100 mode active
  no shutdown

interface Ethernet3/32
  switchport mode trunk
  switchport trunk allowed vlan 1100
  channel-group 100 mode active
  no shutdown
```

These configurations are not enough to allow a server to connect to a storage array. For data to pass, proper zoning must be configured in the DR for a fabric to allow the PWWN of the server port to login into the storage array port.

Due to enhanced device alias and distributed zoning, the configuration can be done in the DR and it will be distributed via CFS to all the switches in that fabric.

The devices are registered into the device database (also distributed via CFS):

```
device-alias mode enhanced
device-alias database
  device-alias name bru-dc1-c200-2-fc0 pwwn 20:00:00:22:bd:d6:67:cc
  device-alias name bru-dc2-c460-1-fc0 pwwn 20:00:c4:64:13:5b:56:7c
  device-alias name bru-dc1-storage-1-1 pwwn 50:00:40:20:01:f4:40:0d
  device-alias name bru-dc1-storage-2-1 pwwn 50:00:40:20:02:f4:23:41
  device-alias name bru-dc2-storage-1-1 pwwn 50:00:40:20:01:fc:6c:7c
  device-alias name bru-dc1-cluster1-server1-fc2 pwwn 20:01:01:25:b5:01:00:ef
  device-alias name bru-dc1-cluster1-server2-fc2 pwwn 20:01:01:25:b5:01:00:cf
  device-alias name bru-dc1-cluster1-server3-fc2 pwwn 20:01:01:25:b5:01:00:ee
  device-alias name bru-dc1-cluster1-server4-fc2 pwwn 20:01:01:25:b5:01:00:ed
  device-alias name bru-dc1-cluster1-server5-fc2 pwwn 20:01:01:25:b5:01:00:8d
  device-alias name bru-dc1-cluster1-server6-fc2 pwwn 20:01:01:25:b5:01:00:0c
  device-alias name bru-dc1-cluster1-server7-fc2 pwwn 20:01:01:25:b5:01:00:4c
  device-alias name bru-dc1-cluster1-server8-fc2 pwwn 20:01:01:25:b5:01:00:8c
  device-alias name bru-dc1-cluster1-server9-fc0 pwwn 20:01:01:25:b5:01:00:8b
  device-alias name bru-dc2-cluster1-server1-fc0 pwwn 20:02:01:25:b5:01:00:9f
  device-alias name bru-dc2-cluster1-server2-fc0 pwwn 20:02:01:25:b5:01:00:7f
  device-alias name bru-dc2-cluster1-server3-fc0 pwwn 20:02:01:25:b5:01:00:5f
  device-alias name bru-dc2-cluster1-server4-fc0 pwwn 20:02:01:25:b5:01:00:3f
  device-alias name bru-dc1-cluster1-server10-fc0 pwwn 20:01:01:25:b5:01:00:6b

device-alias commit
```

And then device aliases are used for zoning this because a change in the device alias applied immediately to the zoning configuration and it allows for easier management:

```
zone mode enhanced vsan 100
!Active Zone Database Section for vsan 100
zone name bru-dc1-cluster1-server1-fc2_bru-dc1-storage-1-1 vsan 100
    member pwwn 20:01:01:25:b5:01:00:ef
!               [bru-dc1-cluster1-server1-fc2]
    member pwwn 50:00:40:20:01:f4:40:0d
!               [bru-dc1-storage-1-1]

zone name bru-dc1-cluster1-server2-fc2_bru-dc1-storage-1-1 vsan 100
    member pwwn 20:01:01:25:b5:01:00:cf
!               [bru-dc1-cluster1-server2-fc2]
    member pwwn 50:00:40:20:01:f4:40:0d
!               [bru-dc1-storage-1-1]

zone name bru-dc1-c200-2-fc0_bru-dc1-storage-2-1 vsan 100
    member pwwn 20:00:00:22:bd:d6:67:cc
!               [bru-dc1-c200-2-fc0]
    member pwwn 50:00:40:20:02:f4:23:41
!               [bru-dc1-storage-2-1]

zone name bru-dc1-cluster1-server1-fc2_bru-dc2-storage-1-1 vsan 100
    member pwwn 20:01:01:25:b5:01:00:ef
!               [bru-dc1-cluster1-server1-fc2]
    member pwwn 50:00:40:20:01:fc:6c:7c
!               [bru-dc2-storage-1-1]

zone name bru-dc1-cluster1-server2-fc2_bru-dc2-storage-1-1 vsan 100
    member pwwn 20:01:01:25:b5:01:00:cf
!               [bru-dc1-cluster1-server2-fc2]
    member pwwn 50:00:40:20:01:fc:6c:7c
!               [bru-dc2-storage-1-1]
```

```
zone name bru-dc1-cluster1-server1-fc2_bru-dc1-storage-2-1 vsan 100
    member pwwn 20:01:01:25:b5:01:00:ef
!               [bru-dc1-cluster1-server1-fc2]
    member pwwn 50:00:40:20:02:f4:23:41
!               [bru-dc1-storage-2-1]

zone name bru-dc1-cluster1-server2-fc2_bru-dc1-storage-2-1 vsan 100
    member pwwn 20:01:01:25:b5:01:00:cf
!               [bru-dc1-cluster1-server2-fc2]
    member pwwn 50:00:40:20:02:f4:23:41
!               [bru-dc1-storage-2-1]

zoneset name ucs_storage vsan 100
    member bru-dc1-cluster1-server1-fc2_bru-dc1-storage-1-1
    member bru-dc1-cluster1-server2-fc2_bru-dc1-storage-1-1
    member bru-dc1-c200-2-fc0_bru-dc1-storage-2-1
    member bru-dc1-cluster1-server1-fc2_bru-dc2-storage-1-1
    member bru-dc1-cluster1-server2-fc2_bru-dc2-storage-1-1
    member bru-dc1-cluster1-server1-fc2_bru-dc1-storage-2-1
    member bru-dc1-cluster1-server2-fc2_bru-dc1-storage-2-1

zoneset activate name ucs_storage vsan 100
```

UCS Server

Now we have reached the bottom of the datacenter and we can see how the servers are configured to attach to the rest of the topology.

As described before, the servers are not distinct physical machines, but are blade servers connected via the UCS Fabric Interconnect (FI) to the network. The Fabric Interconnect appears as a host to the network and takes care of managing the servers and their connectivity [37].

Because the UCS and the Fabric Interconnect configuration is done via a GUI, instead of configuration, there will be screenshots of the application.

The FI are connected together using 2 special interfaces called L1 and L2 that are used as a redundant channel to carry a special clustering protocol that allows a single point of management, replication the configuration from one unit to the other and keepalive to see if any failure occurs in one of the two switches.

The FI are also connected to the network via a management interface each that allows SSH, telnet and HTTP in for management and access.

Once the FI have been connected, they are configured via console to perform the initial setup where a cluster name and cluster IP address is chosen. The two switches also each have a management IP address, but the pair is accessed using the shared cluster IP.

Once this has been configured, the UCS system can be accessed via the GUI by downloading and executing a Java application that is found when connecting to the cluster management IP.

This is what the GUI looks like:

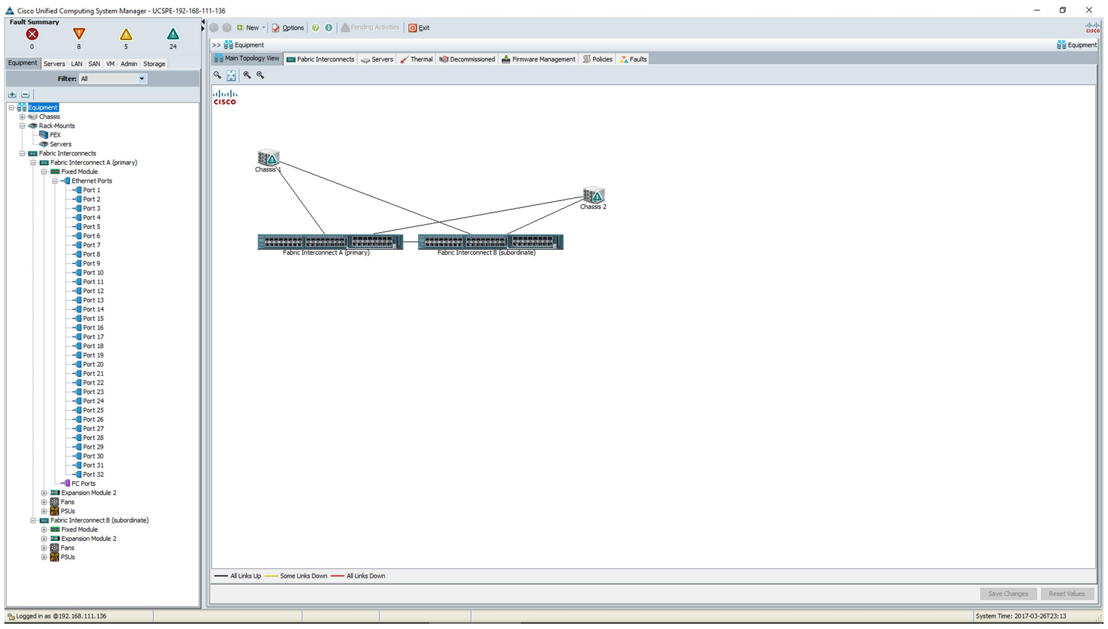

Cisco Unified Computing System Manager - UCSPE-192-168-111-136
Fault Summary
Equipment
Servers
LAN
SAN
VM
Admin
Storage
Main Topology View
Fabric Interconnects
Chassis 1
Chassis 2
Fabric Interconnect A (primary)
Fabric Interconnect B (subordinate)
All Links Up
Some Links Down
All Links Down
Logged in as @192.168.111.136
System Time: 2017-03-26T23:13

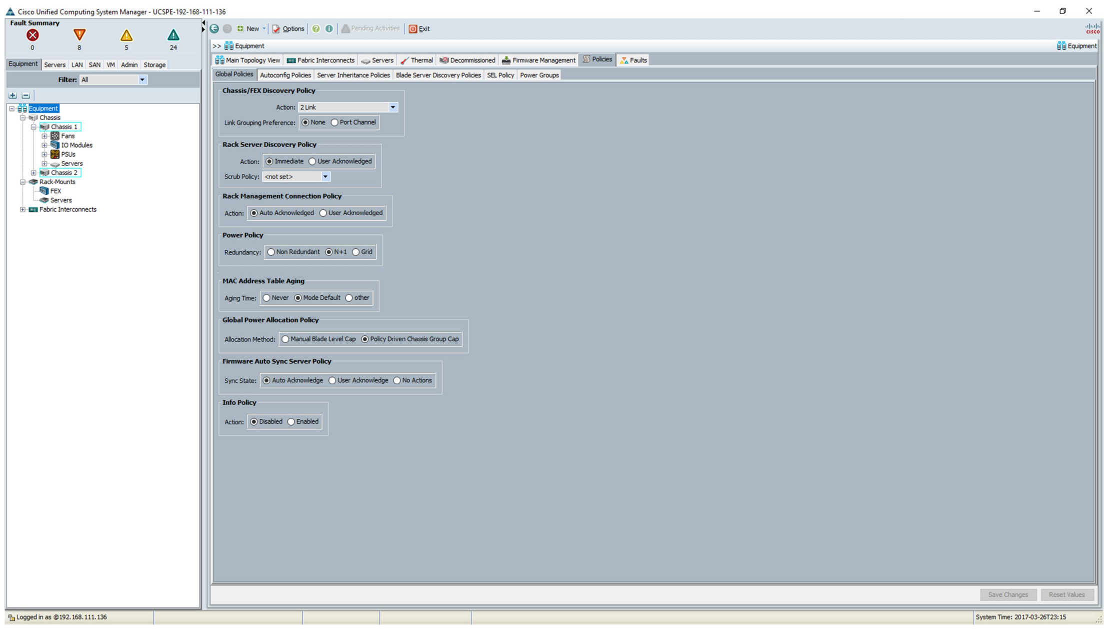
Figure 57: Initial UCS configuration page.

From here we can access all the parts of the configuration, for both the FI uplinks and the servers.

While there are no particular configuration orders required, different components may have dependencies on others like in a tree, so creating these "leafs" components first is necessary.

One of the first things is configuring the connectivity between the UCS and the chassis containing the servers.

This is done through the chassis discovery page:



Chassis/FEX Discovery Policy

Action: 2 Link

Link Grouping Preference: [✓ None] Port Channel

Rack Server Discovery Policy

Action: [✓ Immediate] User Acknowledged

Scrub Policy: <not set>

Rack Management Connection Policy

Action: [✓ Auto Acknowledged] User Acknowledged

Power Policy

Redundancy: Non Redundant [✓ N+1] Grid

MAC Address Table Aging

Aging Time: Never [✓ Mode Default] other

Global Power Allocation Policy

Allocation Method: Manual Blade Level Cap [✓ Policy Driven Chassis Group Cap]

Firmware Auto Sync Server Policy

Sync State: [✓ Auto Acknowledge] User Acknowledge No Actions

Info Policy

Action: [✓ Disabled] Enabled

Figure 58: chassis discovery policy.

A UCS can connect to the chassis with 1, 2 or 4 links each. 3 links are not supported and links can be grouped in bundles or left individually.

All chassis must have the same policy, so it is not possible to have “more important” chassis that have better connectivity. This will of course be influenced by your requirements in capacity and reliability.

To trigger the chassis discovery, some of the FI ports need to be assigned to servers. This can be done by clicking on the ports in the FI view and choosing the role of the interface:

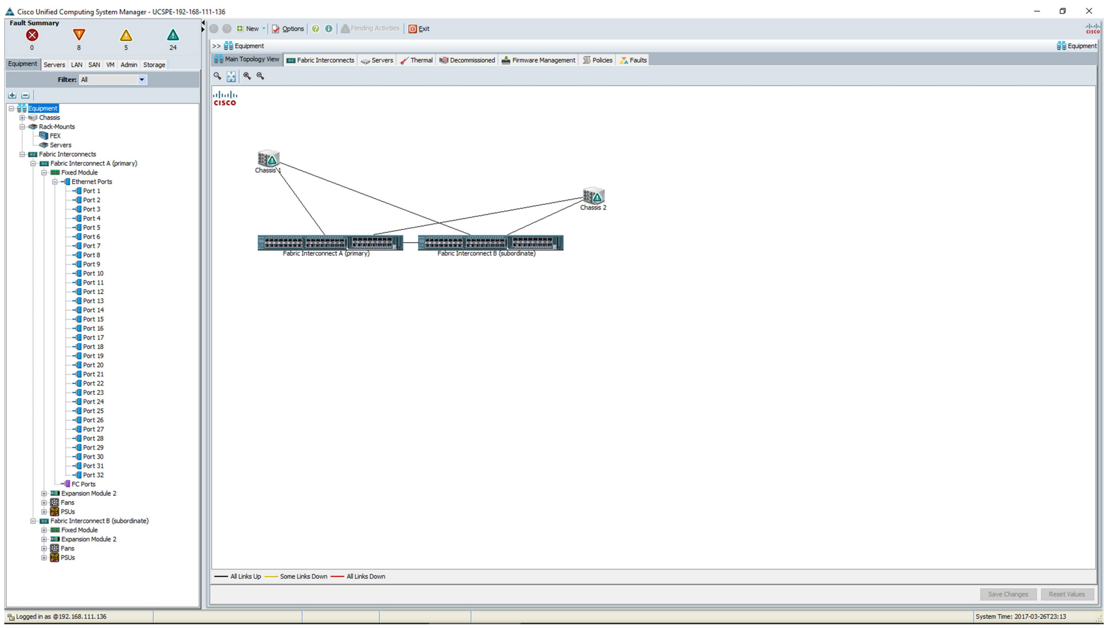


Figure 59: choosing role for FI interfaces.

Once this has been done, the chassis should be discovered after a while:

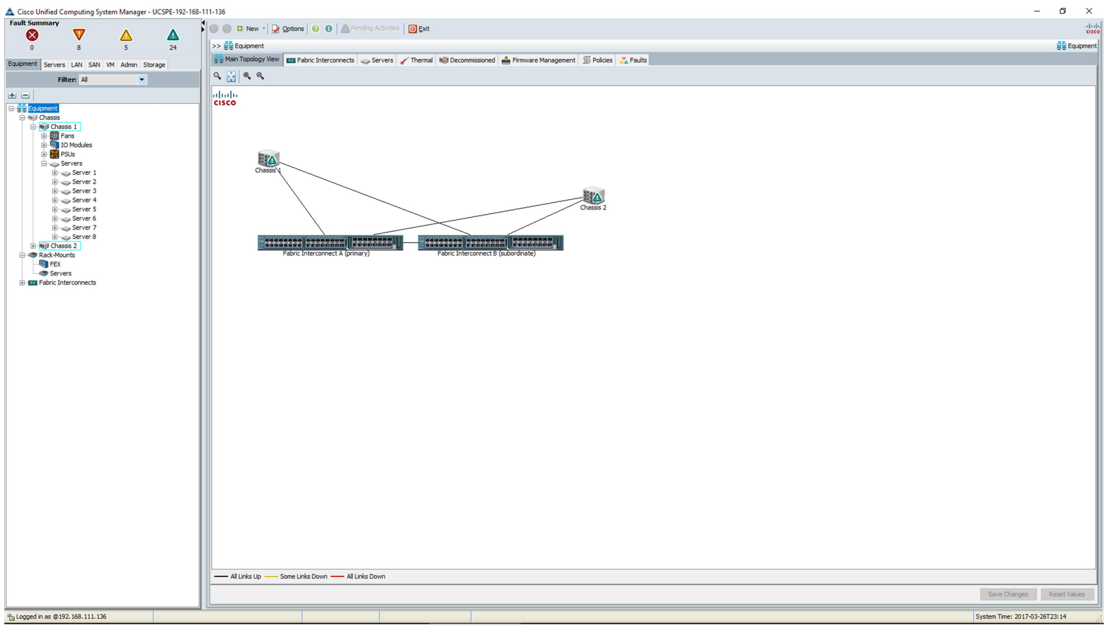
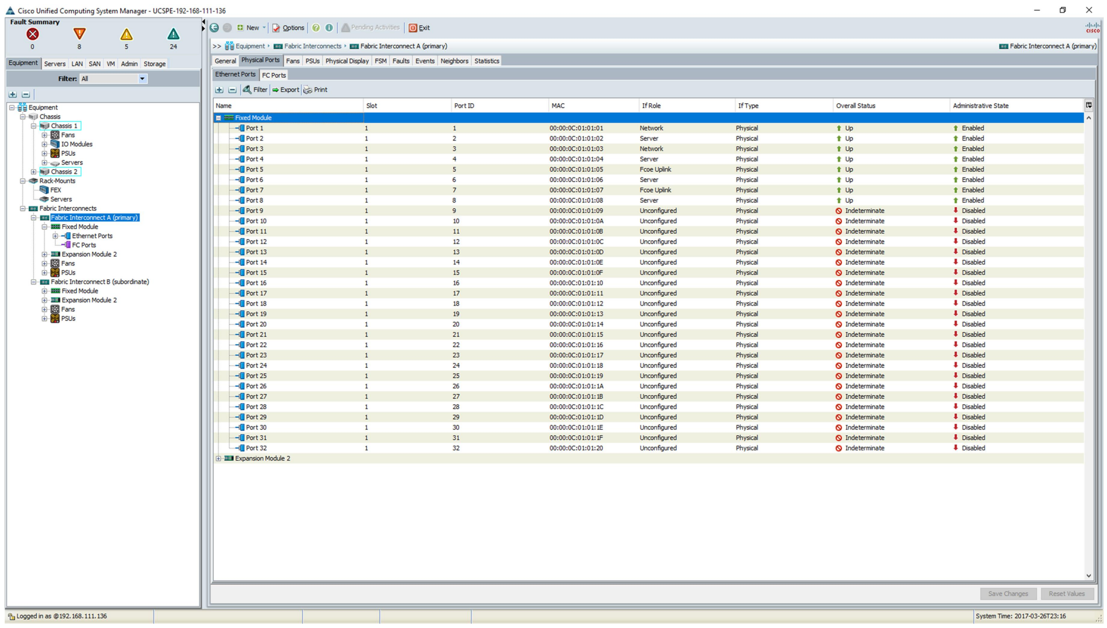


Figure 60: 2 discovered chassis with 8 servers each.

At this point, it is usually suggested to configure the uplink connectivity for the FI to the network, as the management interface cannot carry data because the management interfaces are not available for configuration of vlans or non-management IP.

It is important to remember that from that view, you can also choose which ports are FC uplink ports and which ones are Ethernet uplinks. Because the UCS uses FCoE to the servers, there are no FC ports towards the servers; the only FC ports connected to some hosts are used to connect to any FC storage array that may be connected directly to the FI.

This is what a FI looks like after port configuration:



Name	Slot	Port ID	MAC	If Role	If Type	Overall Status	Administrative State
Fixed Module							
Port 1	1	1	00:00:0C:01:01:01	Network	Physical	Up	Enabled
Port 2	1	2	00:00:0C:01:01:02	Server	Physical	Up	Enabled
Port 3	1	3	00:00:0C:01:01:03	Network	Physical	Up	Enabled
Port 4	1	4	00:00:0C:01:01:04	Server	Physical	Up	Enabled
Port 5	1	5	00:00:0C:01:01:05	Fcoe Uplink	Physical	Up	Enabled
Port 6	1	6	00:00:0C:01:01:06	Server	Physical	Up	Enabled
Port 7	1	7	00:00:0C:01:01:07	Fcoe Uplink	Physical	Up	Enabled
Port 8	1	8	00:00:0C:01:01:08	Server	Physical	Up	Enabled
Port 9	1	9	00:00:0C:01:01:09	Unconfigured	Physical	Indeterminate	Disabled
Port 10	1	10	00:00:0C:01:01:0A	Unconfigured	Physical	Indeterminate	Disabled
Port 11	1	11	00:00:0C:01:01:0B	Unconfigured	Physical	Indeterminate	Disabled
Port 12	1	12	00:00:0C:01:01:0C	Unconfigured	Physical	Indeterminate	Disabled
Port 13	1	13	00:00:0C:01:01:0D	Unconfigured	Physical	Indeterminate	Disabled
Port 14	1	14	00:00:0C:01:01:0E	Unconfigured	Physical	Indeterminate	Disabled
Port 15	1	15	00:00:0C:01:01:0F	Unconfigured	Physical	Indeterminate	Disabled
Port 16	1	16	00:00:0C:01:01:10	Unconfigured	Physical	Indeterminate	Disabled
Port 17	1	17	00:00:0C:01:01:11	Unconfigured	Physical	Indeterminate	Disabled
Port 18	1	18	00:00:0C:01:01:12	Unconfigured	Physical	Indeterminate	Disabled
Port 19	1	19	00:00:0C:01:01:13	Unconfigured	Physical	Indeterminate	Disabled
Port 20	1	20	00:00:0C:01:01:14	Unconfigured	Physical	Indeterminate	Disabled
Port 21	1	21	00:00:0C:01:01:15	Unconfigured	Physical	Indeterminate	Disabled
Port 22	1	22	00:00:0C:01:01:16	Unconfigured	Physical	Indeterminate	Disabled
Port 23	1	23	00:00:0C:01:01:17	Unconfigured	Physical	Indeterminate	Disabled
Port 24	1	24	00:00:0C:01:01:18	Unconfigured	Physical	Indeterminate	Disabled
Port 25	1	25	00:00:0C:01:01:19	Unconfigured	Physical	Indeterminate	Disabled
Port 26	1	26	00:00:0C:01:01:1A	Unconfigured	Physical	Indeterminate	Disabled
Port 27	1	27	00:00:0C:01:01:1B	Unconfigured	Physical	Indeterminate	Disabled
Port 28	1	28	00:00:0C:01:01:1C	Unconfigured	Physical	Indeterminate	Disabled
Port 29	1	29	00:00:0C:01:01:1D	Unconfigured	Physical	Indeterminate	Disabled
Port 30	1	30	00:00:0C:01:01:1E	Unconfigured	Physical	Indeterminate	Disabled
Port 31	1	31	00:00:0C:01:01:1F	Unconfigured	Physical	Indeterminate	Disabled
Port 32	1	32	00:00:0C:01:01:20	Unconfigured	Physical	Indeterminate	Disabled
Expansion Module 2							

Figure 61: FI port configuration summary.

The network ports are Ethernet uplinks, the FCoE ports are FCoE uplinks that carry only the FC encapsulated traffic and the Server ports are ports connected to the chassis.

The servers are configured using Service Profiles. A Service Profile lists all the settings for each configuration part of the server, NIC, vlans, FC ports, vsans, boot policies and so on.

All policies (NIC templates, vsans….) are separately configured in the FI and are linked together in the Service Profile. A Service Profile can be static or updating. An updating Service Profile is updated if the linked policy changes, while a static Service Profile will not.

Before we can create the Service Profile, we need to create all the objects and policies that will have to be used. Some of these objects are mandatory, but don't have a great impact in the server configuration (like the UUID pool or the MAC pool), but others are very important (like the vsan configuration).

We will see now how to create a minimum configuration that will allow us to configure a server and boot it from a SAN boot target.

The first step is to create a list of UUID that will be used to uniquely identify any server. Note that for some of these objects, we need to create a pool of resources that are at least as big as the number of servers we need (as UUID), but for others, we need more, for example, if the servers have more than one NIC, we need multiple MAC addresses per server. Also, failing to create big enough pools will prevent us from installing new servers as we may need to create new profiles for them while it may be easier to use existing Service Profiles.

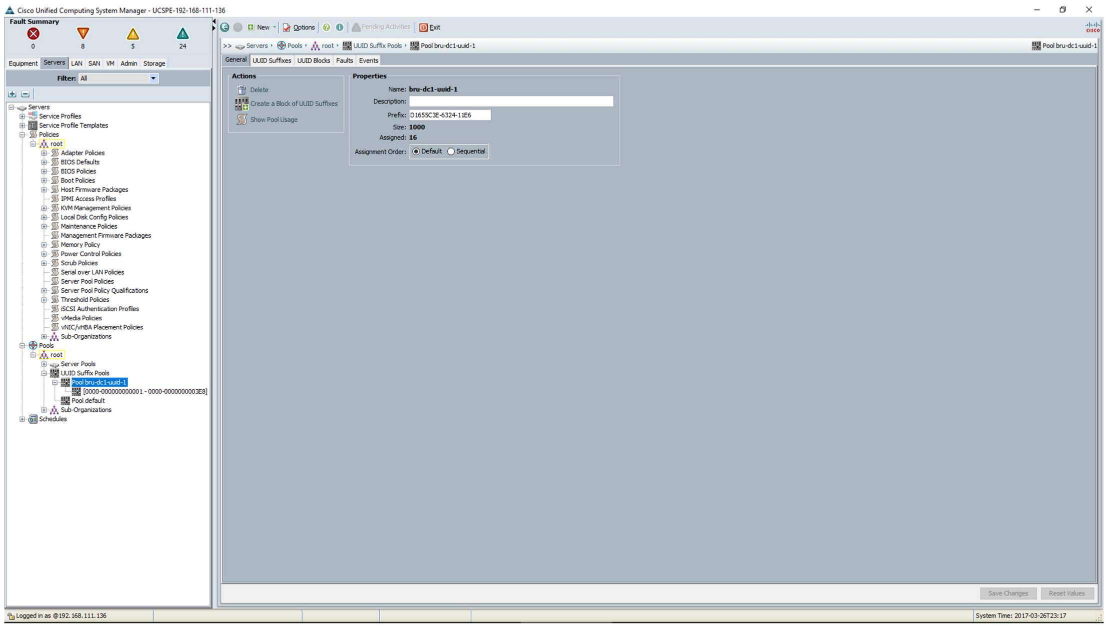
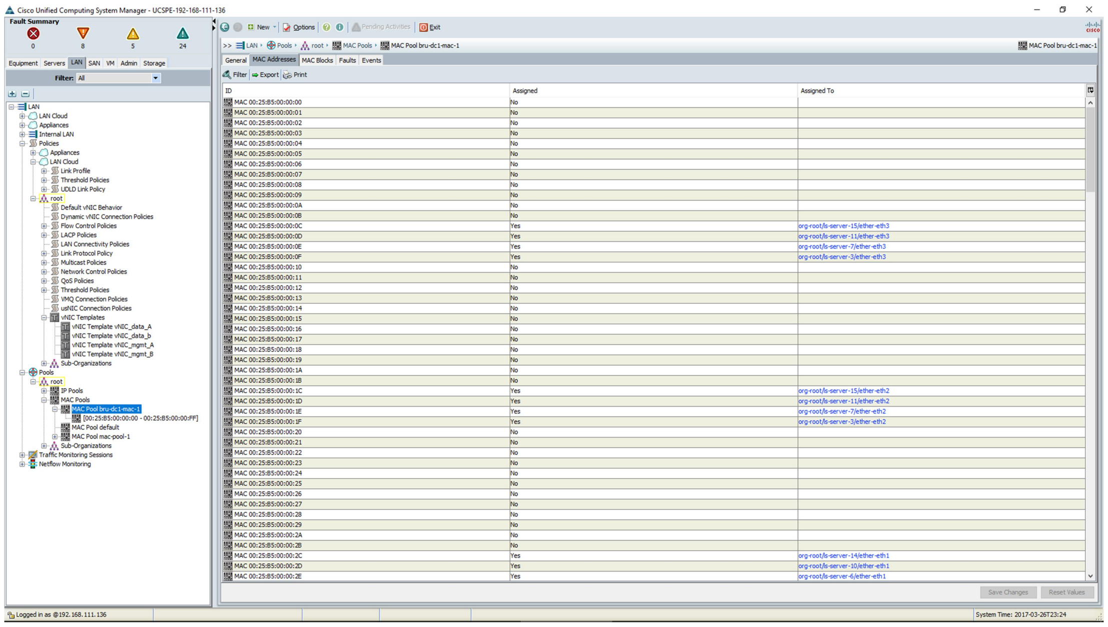

Name: bru-dc1-uuid-1
Description:
Prefix: D1655C3E-6324-11E6
Size: 1000
Assigned: 16
Assignment Order: [✓ Default] Sequential

Figure 62: UUID pool.

We are now creating a MAC pool that will be used to populate the physical MAC address for each server NIC.



ID	Assigned	Assigned To
MAC 00:25:B5:00:00:00	No	
MAC 00:25:B5:00:00:01	No	
MAC 00:25:B5:00:00:02	No	
MAC 00:25:B5:00:00:03	No	
MAC 00:25:B5:00:00:04	No	
MAC 00:25:B5:00:00:05	No	
MAC 00:25:B5:00:00:06	No	
MAC 00:25:B5:00:00:07	No	
MAC 00:25:B5:00:00:08	No	
MAC 00:25:B5:00:00:09	No	
MAC 00:25:B5:00:00:0A	No	
MAC 00:25:B5:00:00:0B	No	
MAC 00:25:B5:00:00:0C	Yes	org-root/ls-server-15/ether-eth3
MAC 00:25:B5:00:00:0D	Yes	org-root/ls-server-11/ether-eth3
MAC 00:25:B5:00:00:0E	Yes	org-root/ls-server-7/ether-eth3
MAC 00:25:B5:00:00:0F	Yes	org-root/ls-server-3/ether-eth3
MAC 00:25:B5:00:00:10	No	
MAC 00:25:B5:00:00:11	No	
MAC 00:25:B5:00:00:12	No	
MAC 00:25:B5:00:00:13	No	
MAC 00:25:B5:00:00:14	No	
MAC 00:25:B5:00:00:15	No	
MAC 00:25:B5:00:00:16	No	
MAC 00:25:B5:00:00:17	No	
MAC 00:25:B5:00:00:18	No	
MAC 00:25:B5:00:00:19	No	
MAC 00:25:B5:00:00:1A	No	
MAC 00:25:B5:00:00:1B	No	
MAC 00:25:B5:00:00:1C	Yes	org-root/ls-server-15/ether-eth2
MAC 00:25:B5:00:00:1D	Yes	org-root/ls-server-11/ether-eth2
MAC 00:25:B5:00:00:1E	Yes	org-root/ls-server-7/ether-eth2
MAC 00:25:B5:00:00:1F	Yes	org-root/ls-server-3/ether-eth2
MAC 00:25:B5:00:00:20	No	
MAC 00:25:B5:00:00:21	No	
MAC 00:25:B5:00:00:22	No	
MAC 00:25:B5:00:00:23	No	
MAC 00:25:B5:00:00:24	No	
MAC 00:25:B5:00:00:25	No	
MAC 00:25:B5:00:00:26	No	
MAC 00:25:B5:00:00:27	No	
MAC 00:25:B5:00:00:28	No	
MAC 00:25:B5:00:00:29	No	
MAC 00:25:B5:00:00:2A	No	
MAC 00:25:B5:00:00:2B	No	
MAC 00:25:B5:00:00:2C	Yes	org-root/ls-server-14/ether-eth1
MAC 00:25:B5:00:00:2D	Yes	org-root/ls-server-10/ether-eth1
MAC 00:25:B5:00:00:2E	Yes	org-root/ls-server-6/ether-eth1

Figure 63: MAC address pool.

In the case of this MAC address pool, the association with servers can already be seen. While creating MAC address pools before Service Profiles creation, the server association column will be empty.

This screen shows the creation of vlans and the FI Ethernet uplink configuration:

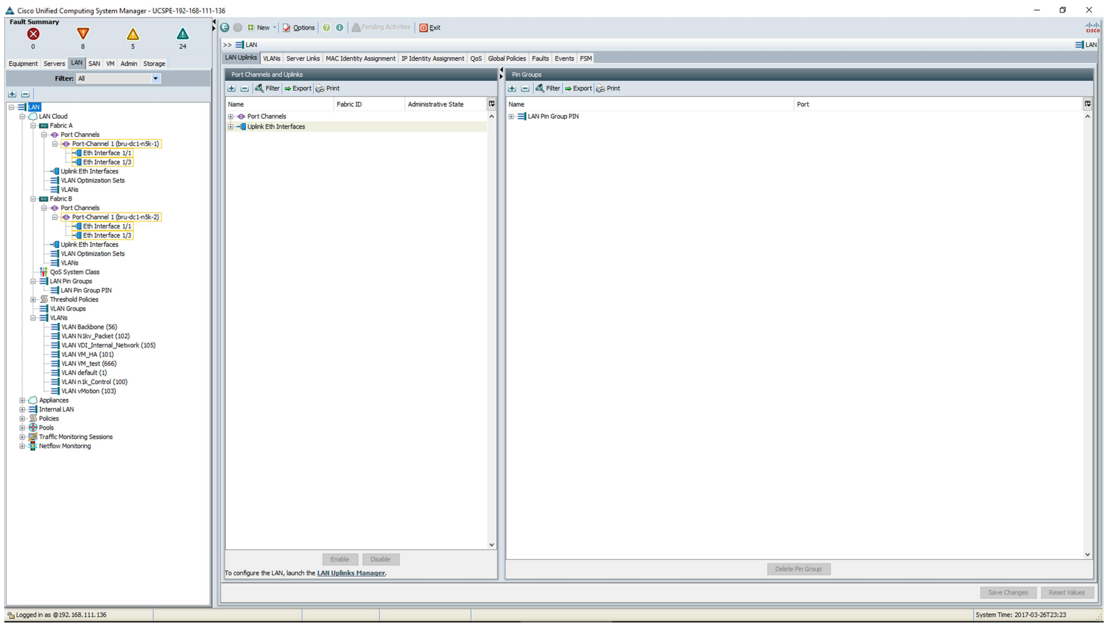


Figure 64: VLANs and port channels uplink.

Uplinks are configured by choosing among the network ports after configuring the FI ports. With default settings, all vlans are trunked everywhere as the FI doesn't run spanning trees, but appears as a host. One uplink is chosen for each server (pinning) and by default all packets from that server are sent over that uplink.

After creating the uplinks, we can create the vlans that the servers will use for data and for management.

All vlans are the same independently of their use.

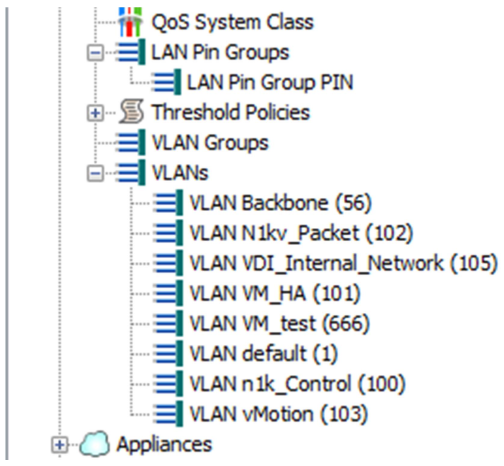
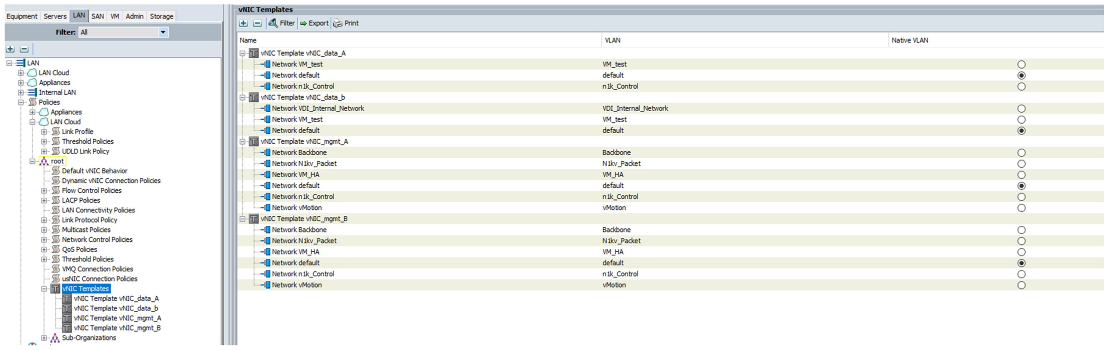


Figure 65: vlans.

Once vlans are created, we can create a template for the vNIC that will be used to create the interfaces when the servers are booted.

In this example, each server will have 2 NIC: one for management carrying the vlans for the backbone management connectivity, one vlan for vMotion, one for VMWare High Availability and other 2 vlans used for the Nexus 1kv. The second NIC will carry the data vlans.

Each NIC can have a native vlans. Untagged frames received on that NIC will be considered as belonging to that native vlan. Tagged frames for vlans that are not allowed on the NIC are discarded.



Name	VLAN	Native VLAN
vNIC Template vNIC_data_A		
Network VM_test	VM_test	○
Network default	default	●
Network n1k_Control	n1k_Control	○
vNIC Template vNIC_data_b		
Network VDI_Internal_Network	VDI_Internal_Network	○
Network VM_test	VM_test	○
Network default	default	●
vNIC Template vNIC_mgmt_A		
Network Backbone	Backbone	○
Network N1kv_Packet	N1kv_Packet	○
Network VM_HA	VM_HA	○
Network default	default	●
Network n1k_Control	n1k_Control	○
Network vMotion	vMotion	○
vNIC Template vNIC_mgmt_B		
Network Backbone	Backbone	○
Network N1kv_Packet	N1kv_Packet	○
Network VM_HA	VM_HA	○
Network default	default	●
Network n1k_Control	n1k_Control	○
Network vMotion	vMotion	○

Figure 66: vNIC templates.

As you can see, because each server has 2 connections to the backplane of the switch, we will have a template for the Fabric A and the Fabric B.

Now we need to configure the vHBA for storage connectivity.

To configure the HBA, we need first to configure some pools of resources that will be used to populate the vHBA profiles.

The pools we need are the WWNN and WWPN:

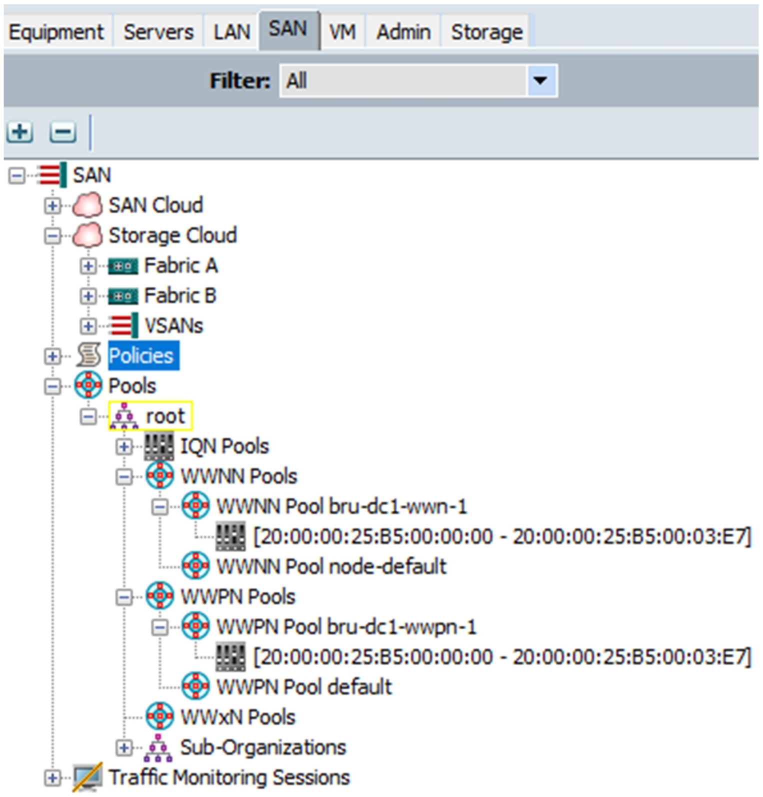
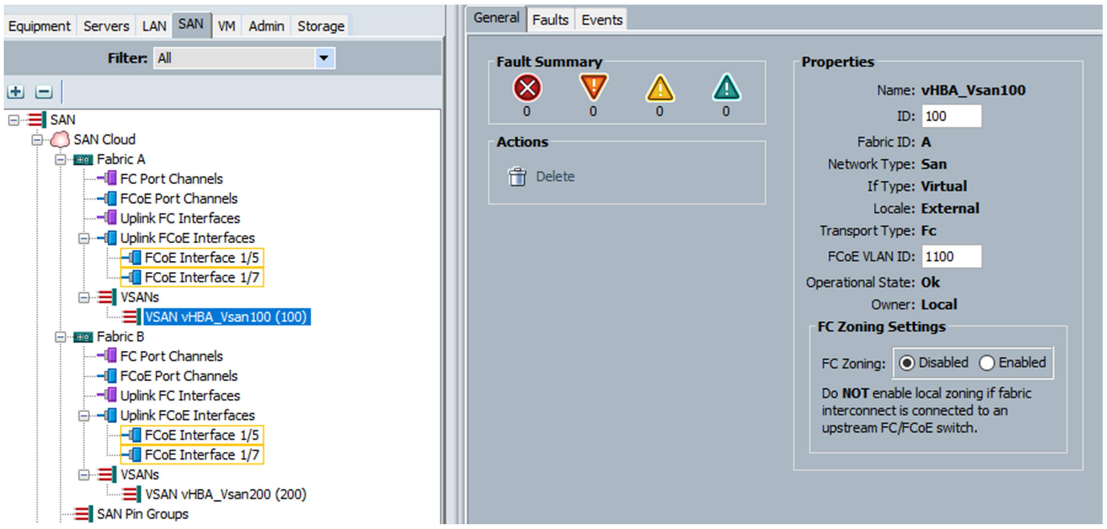


Figure 67: WWNN and WWPN pools.

Then we need to configure the VSANs as we did for the Vlans, but this time, the VSANs have to be pinned to a single fabric because each fabric has its own SAN uplinks to keep logical and physical separation.



Properties

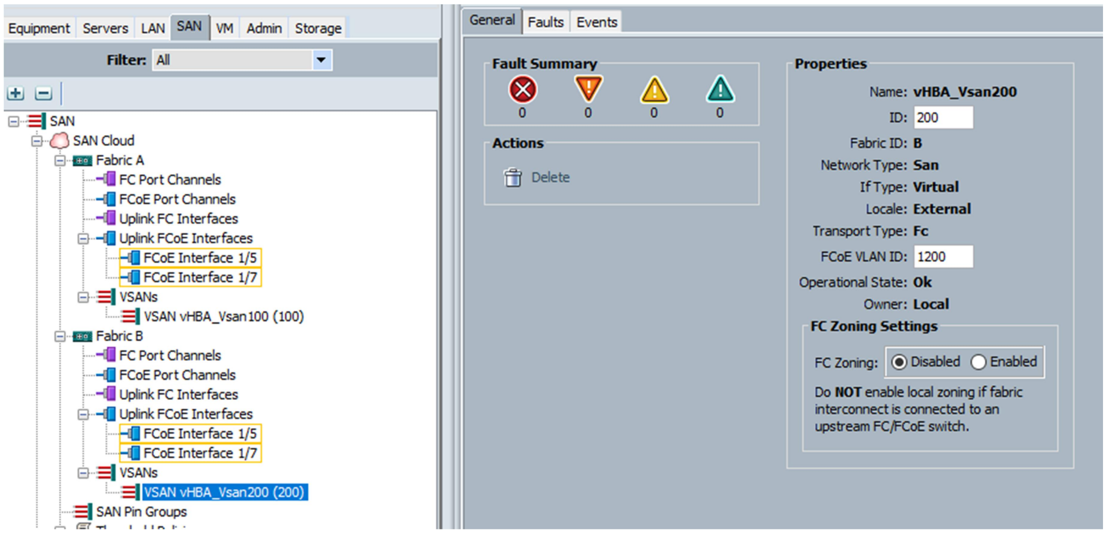
Name: vHBA_Vsan100
ID: 100
Fabric ID: A
Network Type: San
If Type: Virtual
Locale: External
Transport Type: Fc
FCoE VLAN ID: 1100
Operational State: Ok
Owner: Local

FC Zoning Settings

FC Zoning: [✓ Disabled] Enabled

Do NOT enable local zoning if fabric interconnect is connected to an upstream FC/FCoE switch.

Figure 68: VSAN 100 configuration.



Properties

Name:	vHBA_Vsan200
ID:	200
Fabric ID:	B
Network Type:	San
If Type:	Virtual
Locale:	External
Transport Type:	Fc
FCoE VLAN ID:	1200
Operational State:	Ok
Owner:	Local

FC Zoning Settings

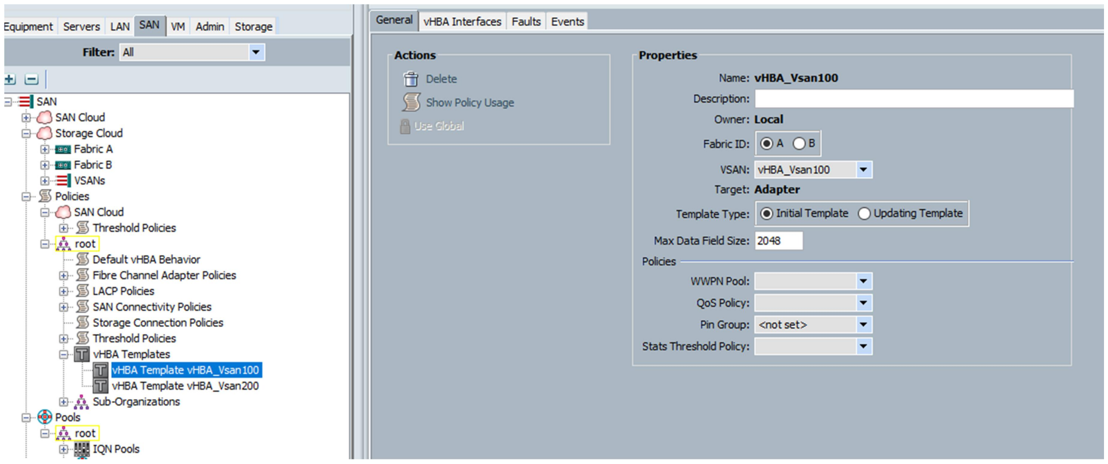
FC Zoning: [✓ Disabled] Enabled

Do **NOT** enable local zoning if fabric interconnect is connected to an upstream FC/FCoE switch.

Figure 69: VSAN 200 configuration.

Note as each VSAN is tied to an FCoE VLAN that is automatically configured. We leave zoning disabled as the FI is in host mode and zoning will be done by another switch.

Now we have what we need to configure the vHBA.



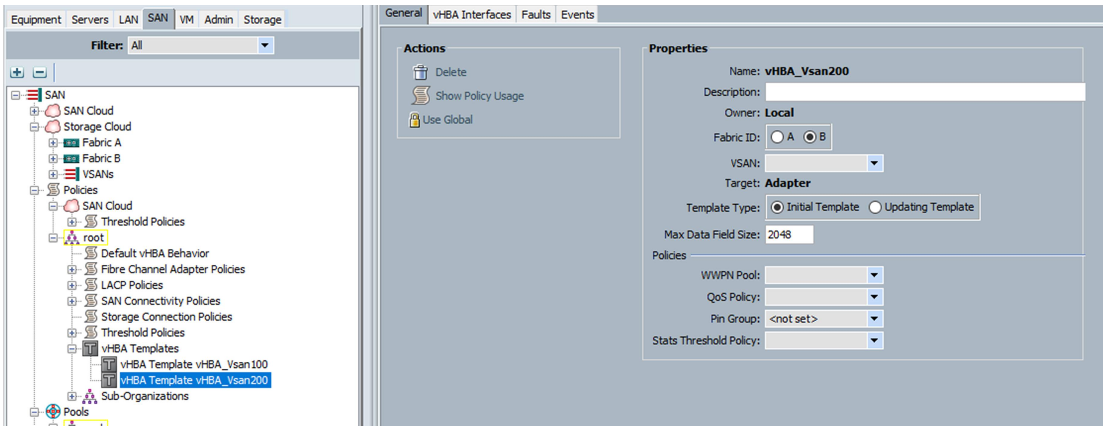
Properties

Name:	vHBA_Vsan100
Description:	
Owner:	Local
Fabric ID:	[✓ A] B
VSAN:	vHBA_Vsan100
Target:	Adapter
Template Type:	[✓ Initial Template] Updating Template
Max Data Field Size:	2048

Policies

WWPN Pool:	
QoS Policy:	
Pin Group:	<not set>
Stats Threshold Policy:	

Figure 70: vHBA configuration for Fabric A.



Properties

Name: **vHBA_Vsan200**
Description:
Owner: **Local**
Fabric ID: A [✓ B]
VSAN:
Target: **Adapter**
Template Type: [✓ Initial Template] Updating Template
Max Data Field Size: 2048

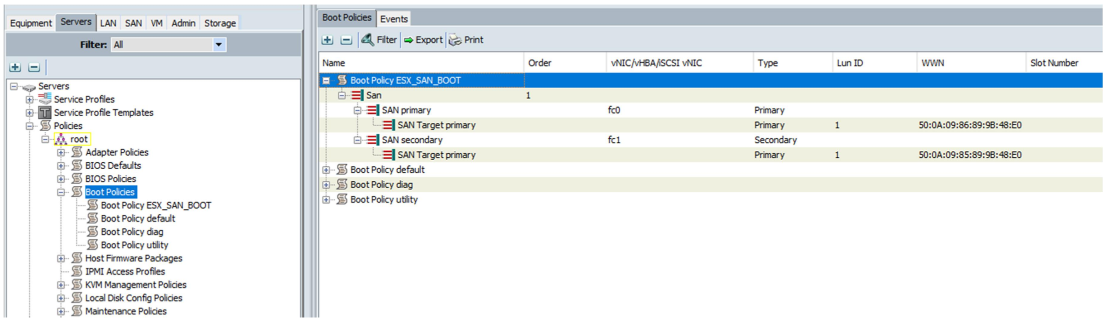
Policies

WWPN Pool:
QoS Policy:
Pin Group: <not set>
Stats Threshold Policy:

Figure 71: vHBA configuration for Fabric B.

Based on the WWNN and WWPN configured, the LUN masking and LUN mapping have to be configured on the storage appliances.

Based on the information of the LUN and the WWPN of the storage appliance we can now configure a boot policy. Server can boot from local HDD or iSCSI HDD, but for this example we will use a SAN boot with a remote disk.

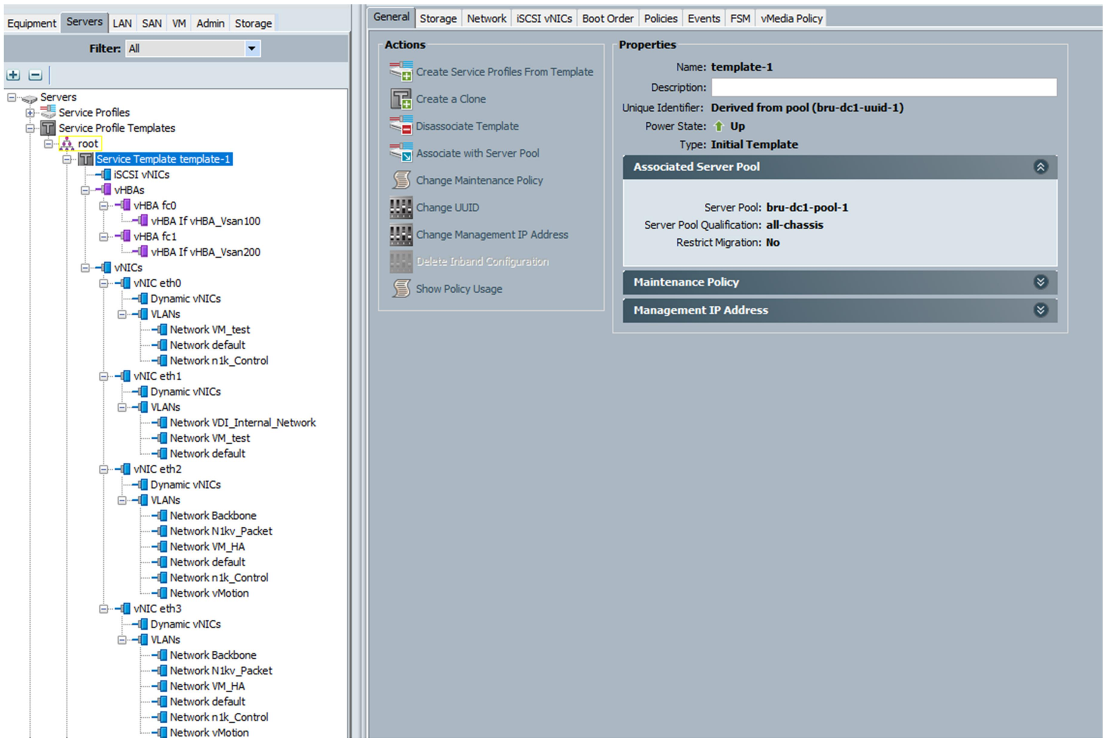


Name	Order	vNIC/vHBA/iSCSI vNIC	Type	Lun ID	WWN	Slot Number
Boot Policy ESX_SAN_BOOT						
San	1					
SAN primary		fc0	Primary			
SAN Target primary			Primary	1	50:0A:09:86:89:9B:48:E0	
SAN secondary		fc1	Secondary			
SAN Target primary			Primary	1	50:0A:09:85:89:9B:48:E0	
Boot Policy default						
Boot Policy diag						
Boot Policy utility						

Figure 72: boot policies configuration.

At this point we have all objects we need to assemble a service profile. A service profile groups together all the configuration options that will be used to populate a server physical attributes and its configuration.

During this phase we will choose how many vNIC and from which template the server will be equipped. The same applies to the vHBA. Later on, during the server programming, the physical values will be chosen from the pools of resources we created before and used in the templates.



Properties

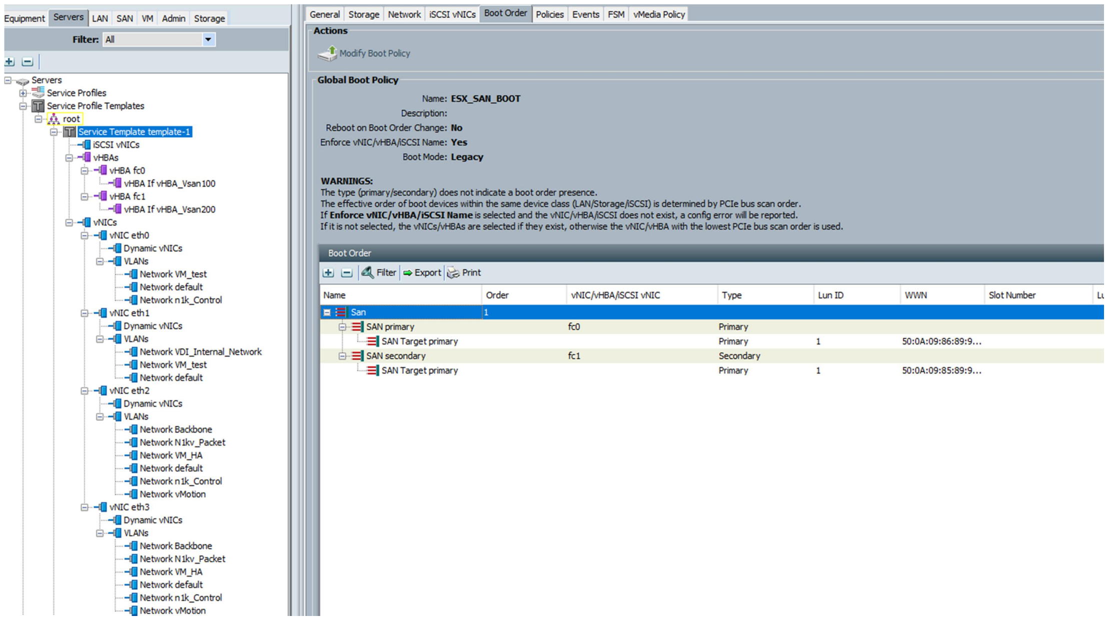
Name: **template-1**
Description:
Unique Identifier: **Derived from pool (bru-dc1-uuid-1)**
Power State: **Up**
Type: **Initial Template**

Associated Server Pool

Server Pool: **bru-dc1-pool-1**
Server Pool Qualification: **all-chassis**
Restrict Migration: **No**

Figure 73: first part of a service profile creation.

In the second part we can see the boot policy that we create to utilize the SAN storage.



Global Boot Policy

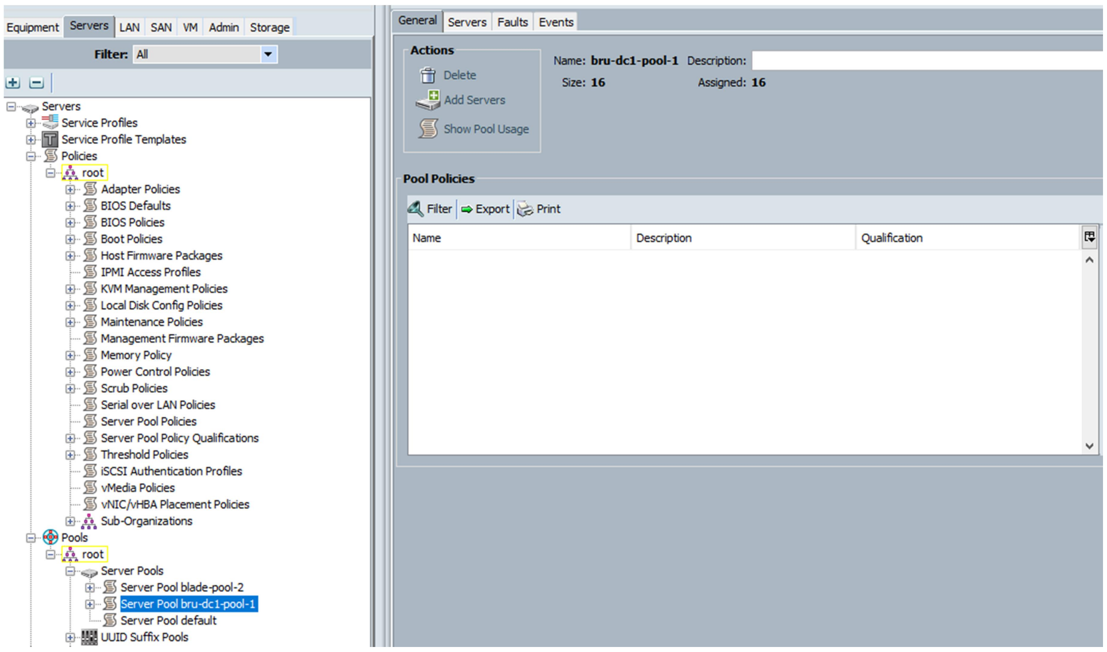
Name: **ESX_SAN_BOOT**
Description:
Reboot on Boot Order Change: **No**
Enforce vNIC/vHBA/iSCSI Name: **Yes**
Boot Mode: **Legacy**

WARNINGS:
The type (primary/secondary) does not indicate a boot order presence.
The effective order of boot devices within the same device class (LAN/Storage/iSCSI) is determined by PCIe bus scan order.
If **Enforce vNIC/vHBA/iSCSI Name** is selected and the vNIC/vHBA/iSCSI does not exist, a config error will be reported.
If it is not selected, the vNICs/vHBAs are selected if they exist, otherwise the vNIC/vHBA with the lowest PCIe bus scan order is used.

Name	Order	vNIC/vHBA/iSCSI vNIC	Type	Lun ID	WWN	Slot Number
San	1					
SAN primary		fc0	Primary			
SAN Target primary			Primary	1	50:0A:09:86:89:9...	
SAN secondary		fc1	Secondary			
SAN Target primary			Primary	1	50:0A:09:85:89:9...	

Figure 74: second part of a service profile creation.

At this point we can assign a service profile to a server. Rather than doing this manually, we can create a server pool to select a set of the available and unassociated servers and link them with a service profile. The server poll configuration requires some arguments, like the model of the servers, the number of CPU or quantity of RAM. Those arguments will be used to select the servers that will belong to the pool.



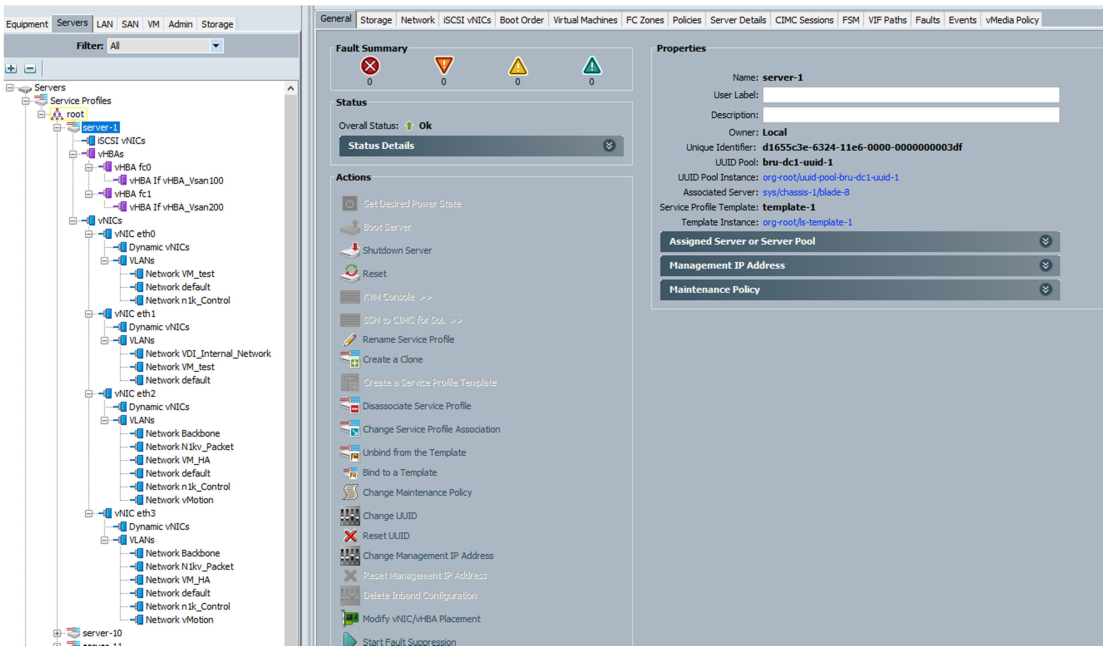
Name: bru-dc1-pool-1 Description:

Size: 16 Assigned: 16

Name	Description	Qualification

Figure 75: server pool configuration.

At this point, we can associate the service profile with the server pool. The servers in the chassis that belong to the server pool will be automatically associated with the service profile and if enough resources are available in the resource pools, the servers will be booted and configured with the physical address. In addition, the servers will also be configured with the boot policy and at the next reload, if the LUN masking, mapping and zoning has been done correctly, the servers will boot the OS.



```
Servers
  Service Profiles
    root
      server-1
        iSCSI vNICs
        vHBAs
          vHBA fc0
            vHBA If vHBA_Vsan100
          vHBA fc1
            vHBA If vHBA_Vsan200
        vNICs
          vNIC eth0
            Dynamic vNICs
            VLANs
              Network VM_test
              Network default
              Network n1k_Control
          vNIC eth1
            Dynamic vNICs
            VLANs
              Network VDI_Internal_Network
              Network VM_test
              Network default
          vNIC eth2
            Dynamic vNICs
            VLANs
              Network Backbone
              Network N1kv_Packet
              Network VM_HA
              Network default
              Network n1k_Control
              Network vMotion
          vNIC eth3
            Dynamic vNICs
            VLANs
              Network Backbone
              Network N1kv_Packet
              Network VM_HA
              Network default
              Network n1k_Control
              Network vMotion
      server-10
```

Fault Summary: 0 0 0 0

Status
Overall Status: Ok

Properties
Name: server-1
User Label:
Description:
Owner: Local
Unique Identifier: d1655c3e-6324-11e6-0000-0000000003df
UUID Pool: bru-dc1-uuid-1
UUID Pool Instance: org-root/uuid-pool-bru-dc1-uuid-1
Associated Server: sys/chassis-1/blade-8
Service Profile Template: template-1
Template Instance: org-root/ls-template-1

Figure 76: server configured.

Future work

With the growth of the internet, new needs and new services, the datacenters are always evolving.

The applications of tomorrow will give customized content targeted to each single user. This causes the traffic pattern to drastically change, from north-to-south, to east-to-west, so server to server within the datacenter.

This new traffic is required to perform operations like map reduce and data mining and puts a strain on the network that was not there before. Enormous amounts of data are passed between servers and a normal topology is not well suited anymore. The future datacenters will implement a statistically non-blocking Clos network organized in stages, where each stage has the same downstream and upstream bandwidth capacity.

This topology allows for almost infinite horizontal scalability and, when coupled with a tool to organize software workloads and move them around, better efficiency and performances.

A Clos network is usually made of point to point layer 3 links where the servers run a routing protocol used to announce the IPs belonging to the software services that are running on it.

If layer 2 extensions are still required, this is usually done by using an overlay network on top of the layer 3 Clos topology that simulates vlans. This is usually achieved by some sort of GRE encapsulation.

Another trend that is being seen is the one of the cloud natives: companies that have never had any equipment, but started by renting compute from a public cloud provider. Such companies use containers to deploy software. Containers are a way to virtualize an OS so that multiple workloads can run in the same OS. This new paradigm reduces the need for maintenance as there is only one OS per machine and, in the case of public cloud; the OS is maintained by the provider.

Such a distributed environment allows us to increase the availability of a service by distributing it in various datacenters and a way to provide statefulness. The infrastructure is not required anymore to have a really high availability and this allows for further cost saving.

Conclusion

This thesis presented a sample design of a horizontally scalable datacenter including a layer 2 WAN extension able to accommodate legacy applications and new ones.

Datacenters are currently moving towards a fully layer 3 CLOS style fabric, but the majority of the one found in the field don't have scalability requirements to justify this kind of design.

The majority of the enterprises are still using legacy applications that need layer 2 connectivity for exchanging information or providing failover functionalities. In the absence of a centralized software defined networking controller that can simulate layer 2 connectivity through an overlay network, the enterprises will still be using Ethernet protocol features like vlans and spanning tree to provide services to their end users.

These features however do not work well over long distances and links with high latency like WAN links, so technologies like OTV have been created to extend layer 2 domains over the internet minimizing the bandwidth cost of flooding traffic and keeping STP events local to a site.

We have mentioned technologies like Fabric Path that promise layer 3 functionalities at layer 2 level and allow better utilization of all available links.

Regarding the storage side, we have explored the Fiber Channel protocol and its differences with Ethernet when it comes to design. FCoE promises a convergence of the two worlds by merging the advanced requirements of FC with the cheap cost (and lower reliability) of Ethernet.

Designs similar to the one described in this document are used by the biggest enterprises in the world, allowing them to provide the services that we all use every day.

Bibliography

[1] https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data

[2] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#719e260560ba

[3] https://en.wikipedia.org/wiki/Data_center

[4] https://en.wikipedia.org/wiki/Fibre_Channel

[5] https://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-series-switches/white_paper_c11-605488.html

[6] https://en.wikipedia.org/wiki/Storage_area_network

[7] https://www.netapp.com/us/info/what-is-storage-area-network.aspx

[8] https://en.wikipedia.org/wiki/Spanning_Tree_Protocol

[9] https://en.wikipedia.org/wiki/Virtualization

[10] https://www.vmware.com/solutions/virtualization.html

[11] https://www.google.com/about/datacenters/efficiency/how/

[12] https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/view/vmware-stateless-virtual-desktops-ref-arch-white-paper.pdf

[13] https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/data-center-desktop-virtualization.html

[14] https://en.wikipedia.org/wiki/TRILL_(computing)

[15] https://tools.ietf.org/html/rfc5556

[16] https://tools.ietf.org/html/rfc6326

[17] https://www.cisco.com/c/dam/en/us/products/collateral/switches/nexus-7000-series-switches/white_paper_c07-728188.pdf

[18] https://www.cisco.com/c/dam/global/en_hk/assets/pdfs/marty_ma_data_center_3-0_technology_evolution.pdf

[19] https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/HA_campus_DG/hacampusdg.html

[20] https://www.cisco.com/c/en/us/products/switches/nexus-1000v-switch-vmware-vsphere/index.html

[21] http://hsi.web.cern.ch/HSI/fcs/spec/overview.htm

[22] https://www.sciencedirect.com/topics/computer-science/fibre-channel-protocol

[23] https://www.sanog.org/resources/sanog8/sanog8-san-functional-overview-asimkhan.pdf

[24] https://en.wikipedia.org/wiki/Fibre_Channel_over_Ethernet

[25] https://www.cisco.com/c/dam/en/us/products/collateral/storage-networking/mds-9700-series-multilayer-directors/guide-c07-733622.pdf

[26] https://www.cisco.com/c/dam/en/us/td/docs/switches/datacenter/sw/design/vpc_design/vpc_best_practices_design_guide.pdf

[27] https://www.cisco.com/c/dam/en/us/products/collateral/switches/nexus-7000-series-switches/guide_c07-728315.pdf

[28] https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DCI/whitepaper/DCI3_OTV_Intro/DCI_1.html

[29] https://en.wikipedia.org/wiki/IS-IS

[30] https://tools.ietf.org/html/rfc1142

[31] https://www.cisco.com/c/en/us/products/switches/nexus-7000-series-switches/index.html

[32] https://www.cisco.com/c/en/us/products/collateral/storage-networking/mds-9500-series-multilayer-directors/white_paper_C11-515630.pdf

[33] https://tools.cisco.com/security/center/resources/copp_best_practices

[34] https://docs.vmware.com/en/VMware-Validated-Design/4.3/com.vmware.vvd.sddc-design.doc/GUID-40CD3531-FDFD-4CC7-A608-94094329B27F.html

[35] https://www.vmware.com/pdf/vmware-validated-design-20-reference-architecture-guide.pdf

[36] https://www.cisco.com/c/en/us/products/collateral/interfaces-modules/storage-networking-modules/prod_white_paper0900aecd8044c807.html

[37] https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-manager/products-installation-and-configuration-guides-list.html