云计算数据中心网络技术

1 前言

题目并不吸引人，主要是作者犯懒，罗列了一下关键词而已，当然好处是一看就知道文章要说啥。

简单说下结构，首先讲讲云计算，其次是数据中心，再然后是网络，重点还是技术。内容是循序渐进的，可以理解前面每个词都是后面词的定语。

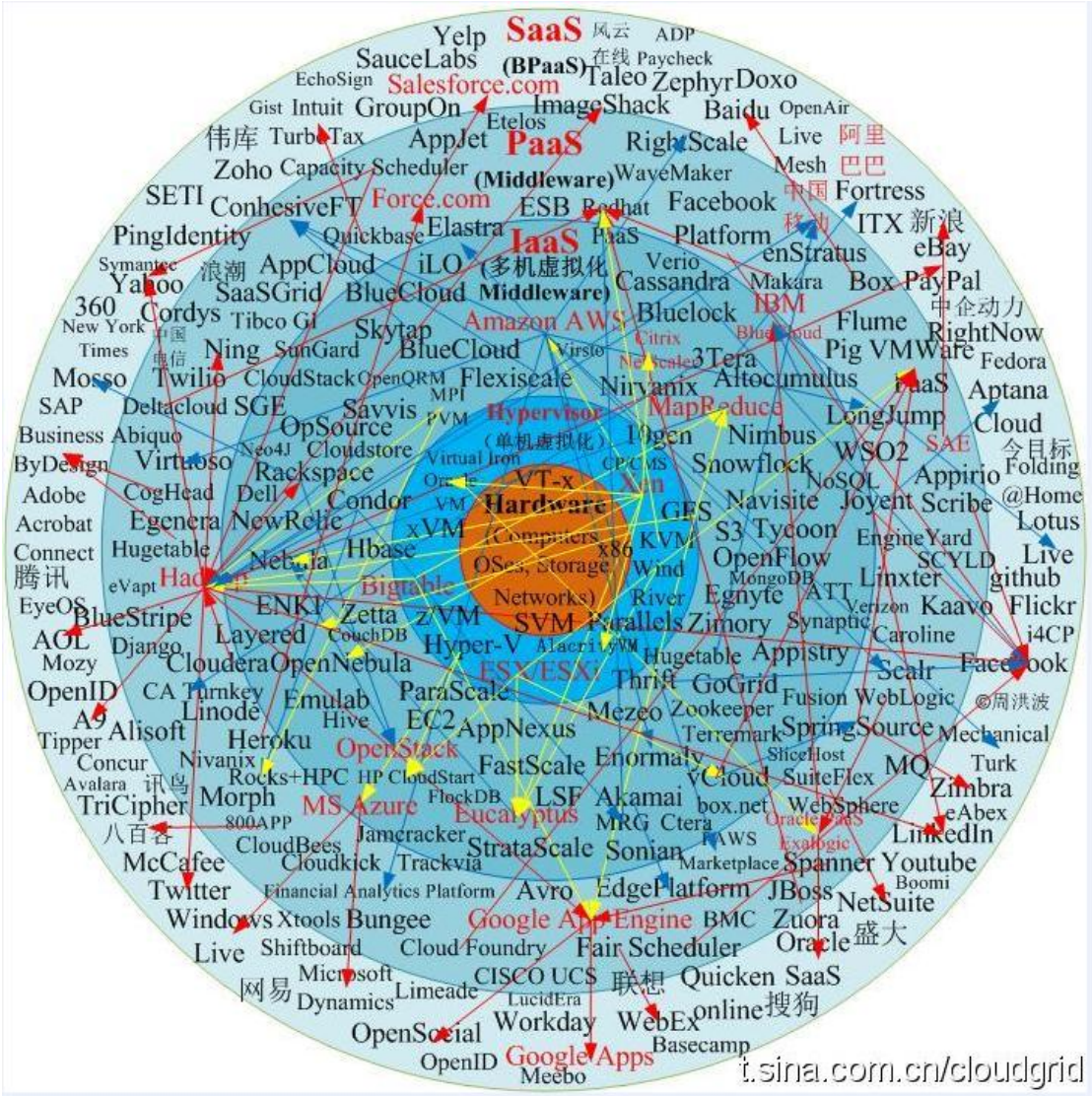
本文希望能够帮读者对云计算的数据中心的网络的技术建立起全面的结构性认识，因此除了总体思路的描述外，在介绍过程中也会力争用三言两语对前面部分中涉及的每个技术点都有所说明，至少让人明白这个东东怎么来的，要干啥和怎么干。但由于受篇幅所限，无法做到很详细，大家如果对某个技术点真感兴趣时，还是去网上找些更细节的资料来理解，本文是打算没有写成一本书的。

力争做到让文档读起来不感到枯燥吧，对作者来说那是相当有挑战的。

2 云计算

最早接触这个词好像是06年了，当时也是刚刚开始接触数据中心不久，这几年眼睁睁看着它被炒作得一塌糊涂，现在已经成为非常给力的一个概念。和别人谈数据中心要是不提云计算，你还真不好意思张这个嘴。

服务器厂商在喊云计算，网络、操作系统、应用软件甚至存储厂商都在喊。大家各喊各的，让我们感觉听上去都有那么点儿味道，但下来仔细一琢磨大都还在云里雾里。看看这张网上截取的云计算产业全景图，估计没有几个能够不头晕的。



云计算的各方面定义很多，基于用户的视角来看，目的就是让使用者在不需了解资源的具体情况下做到按需分配，将计算资源虚拟化为一片云。站在高处看，当前的主流云计算更贴切于云服务，个人认为可理解为早先运营商提供数据中心服务器租用服务的延伸。以前用户租用的是一台台物理服务器，现在租用的是虚拟机，是软件平台甚至是应用程序。公认的三个云计算服务层次是IaaS（Infrastructure as a Service）、PaaS（Platform as a Service）和SaaS（Software as a Service），分别对应硬件资源、平台资源和应用资源。对于用户来说：

1、当提供商给你的是一套a 个核CPU、b G大小内存的主机、c M带宽网络以及d G大小存储空间，需要你自己去装系统和搞定应用程序，那么这就是IaaS，举例如Amazon EC2；

2、当提供的是包含基本数据库和中间件程序的一套完整系统，但你还需要根据接口编写自己的应用程序时，那么就是PaaS，举例如Google AppEngine、Microsoft Azure和Amazon

SimpleDB, SQS；

3、最傻瓜的方式自然是连应用程序都写好了，例如你只需要告诉服务提供商想要的是个500人的薪酬管理系统，返回的服务就是个HTTPS的地址，设定好帐号密码就可以访问过去直接使用，这就是SaaS了，如SalesForce、Yahoo Hadoop和Cisco Webex: Collaboration SaaS等。

服务 属性	Amazon EC2	Google App Engine	Microsoft Azure	Yahoo Hadoop
架构	IaaS/PaaS	PaaS	PaaS	SaaS
服务形态	Compute/ Storage	Web application	Web and non- web	Software
管理技术	OS on Xen hypervisor	Application container	OS through Fabric controller	Map / Reduce Architecture
使用者界面	EC2 Command-line tools	Web-based Administratio n console	Windows Azure portal	Command line and web
APIs	yes	yes	yes	yes
收费	yes	yes	yes	no
编程语言	AMI (Amazon Machine Image)	Python	.NET framework	Java,

为啥举例都是国外的呢，因为国内目前的云服务状况是，能提供的都处于IaaS阶段，有喊着要做PaaS的，但还没听说有SaaS的。

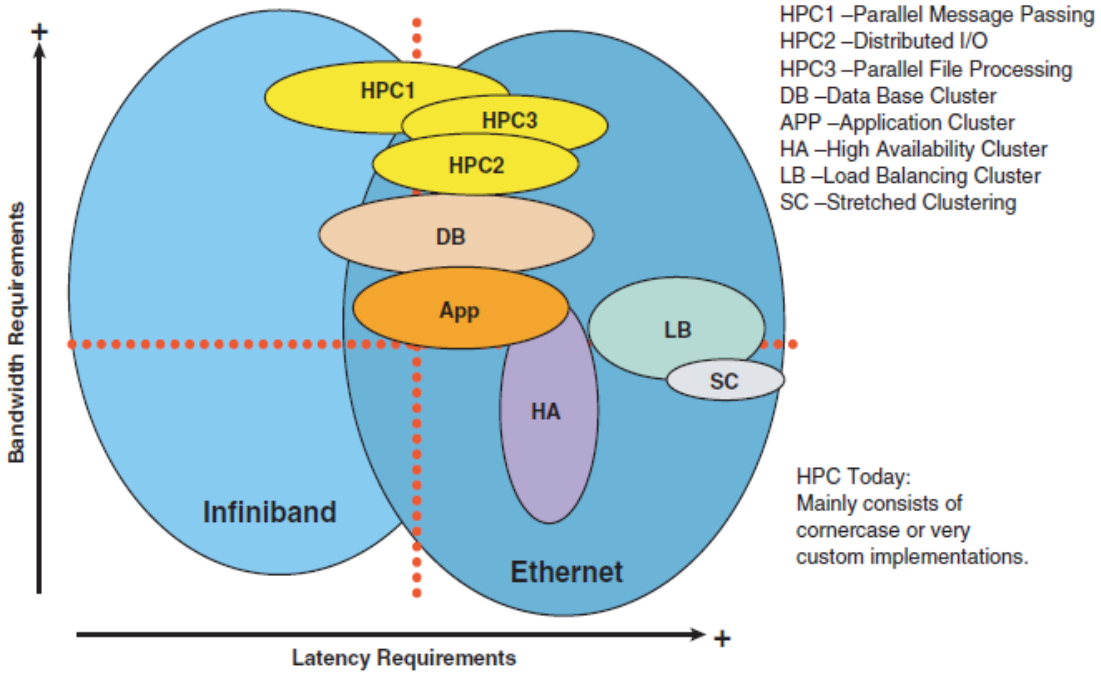
说完公共的，该讲些私货了。

个人理解云计算的核心首先是计算，什么网络、存储、安全等等都是外延，从技术上讲云计算就是计算虚拟化。最早的云计算来自于网格计算，通过一堆性能较差的服务器完成一台超级计算机才能完成的计算任务，简单的说就是计算多虚一。但是现如今一虚多（VM/XEN等）也被一些厂商扯着大旗给忽悠进来，并且成为主流。但是单从技术角度来看，这两者是南辕北辙的。因此云计算技术在下面被作者主观的分为集中云与分散云两个概念来阐述。

2.1 集中云

首先是集中云，根正苗红的多虚一，最早期的也是目前最大的一个典型实际用户就是

Google了 (注意这里说的不是现在Google云服务)。搜索引擎是超级消耗资源的典型应用，从你在网页上一个关键词的搜索点击，到搜索结果的产生，后台是经过了几百上千台服务器的统一计算。至于搜索引擎的工作模型本文就不多说了，网上很多资料的。随着互联网的发展，现在的开心、淘宝、新浪微博等等（好孩子不翻墙），虽然使用者看到的只是在简单的页面进行点击输入，但是后台的工作量已经远远不是少量几台大型服务器能够胜任的了，即使天河一号也不见得能搞定。集中云的应用主力就是这些大型的互联网内容提供商们，当然还有一些传统应用如地震、气象和科研项目的计算也会存在此类需求。



了解了需求，下面简单谈下技术，上图是Cluster集群多虚一技术的简单分布，除了按照承载网络类型可分成Infiniband和Ethernet外，根据技术分，还可分为Active-Standby主备与LoadBalance负载均衡两类。

主备模式好理解，所有的Server里面只有一台干活，其他都是候着的，只有侦听到干活的歇菜了，才开始接管处理任务。主备模式大部分就二虚一提供服务，多了如三虚一什么的其实意义都不太大，无非是为了再多增加些可靠性。主备模式以各类HA集群技术为代表。

而负载均衡模式复杂一些，在所有的LB技术中都存在两个角色，协调者与执行者，协调者一般是一个或多个（需要主备冗余时），主要工作就是接活儿和分活儿（有点儿像包工头）；而执行者就只处理计算了，分到啥就完成啥，典型的苦力。从流量模型上来说，LB集群技术有来回路径一致和三角传输两种，来回路径一致指流量都是客户发起连接，请求协

调者进行处理，协调者分配任务给执行者进行计算，计算完成后结果会都返回到协调者，再由协调者应答客户。这种结构简单，计算者不需要了解外界情况，由协调者统一作为内外接口，安全性最高。此模型主要应用于搜索和地震气象科研计算等业务处理中。三角传输模型指计算者完成计算后直接将结果反馈给客户，此时由于计算者会和客户直接通信，造成安全性降低，但返回流量减少了协调者这个处理节点，性能得到很大提升。此模型主要应用于腾讯新浪的新闻页面和阿里淘宝的电子商务等WEB访问业务。

集中云在云服务中属于富人俱乐部的范围，不是给中小企业和个人玩的，实际上都是各大互联网服务提供商自行搭建集中云以提供自己的业务给用户，不会说哪天雅虎去租用个Google的云来向用户提供自己的新闻页面访问。集中云服务可能的租用对象是那些高度科研项目，因而也导致当前集中云建设上升到国家宏观战略层面的地位。你能想象哪天百度的云服务提供给总装研究院去计算个导弹轨迹，核裂变什么嘛，完全不可能的事。

最后是多虚一对网络的需求。在集中云计算中，服务器之间的交互流量多了，而外部访问的流量相对减少，数据中心网络内部通信的压力增大，对带宽和延迟有了更高的要求，自然而然就催生出后面会讲到的一些新技术（L2MP/TRILL/SPB等）。

题外话，当前的多虚一技术个人认为不够给力，现在把10台4核CPU的服务器虚拟合一后，虚拟的服务器远远达不到一个40核CPU的计算能力。准确的说现在的多虚一只能基于物理服务器的粒度进行合并，理想的情况应该是能够精细到CPU核以及每台设备的内存缓存等等物理构件虚拟合一。这块应该就涉及到超算了，不熟不深谈。总的来说认为技术进步空间巨大，有些搞头。

2.2 分散云

再讲分散云，这块是目前的主流，也是前面提到的云服务的关键底层技术。由于有VMware和Citrix等厂家在大力推广，而且应用内容较集中云更加平民化，随便找台PC或服务器，装几个虚拟机大家都能玩一玩，想干点儿啥都成，也就使其的认知度更加广泛。

一虚多的最主要目的是为了提高效率，力争让所有的CPU都跑到100%，力争让所有的内存和带宽都占满。以前10台Server干的事，我整两台Server每台跑5个虚拟机VM（Virtual Machine）就搞定了，省电省空间省制冷省网线，总之省钱是第一位的（用高级词儿就是绿色环保）。技术方面从实现方案来看，目前大致可分为三类：

操作系统虚拟化OS-Level

在操作系统中模拟出一个个跑应用程序的容器，所有虚拟机共享内核空间，性能最好，耗费资源最少，一个CPU号称可最多模拟500个VPS(Virtual Private Server)或VE(Virtual Environment)。缺点是操作系统唯一，如底层操作系统跑的Windows，VPS/VE就都得跑Windows。代表是Parallels公司（以前叫SWsoft）的Virtuozzo（商用产品）和OpenVZ（开源项目）。Cisco的Nexus 7000猜测也是采用这种方案运行的VDC技术，但不太清楚为什么会有最多4个VDC的数量限制，也许是基于当前应用场景进行规格控制的一种商业手段。

主机虚拟化Hosted

先说下Hypervisor或叫做Virtual Machine Monitor（VMM），它是管理虚拟机VM的软件平台。在主机虚拟化中，Hypervisor就是跑在基础操作系统上的应用软件，与OS-Level中VE的主要区别在于：

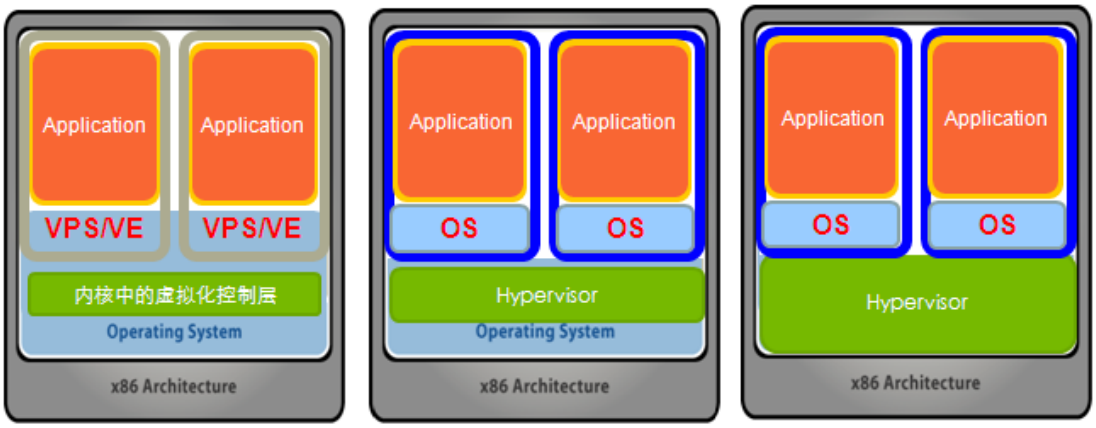
Hypervisor构建出一整套虚拟硬件平台（CPU/Memory/Storage/Adapter），上面需要你再去安装新的操作系统和需要的应用软件，这样底层和上层的OS就可以完全无关化，诸如Windows上跑Linux一点儿问题没有；

VE则可以理解为盗用了底层基础操作系统的资源去欺骗装在VE上的应用程序，每新创建出一个VE，其操作系统都是已经安装好了的，和底层操作系统完全一样，所以VE比较VM（包括主机虚拟化和后面的裸金属虚拟化）运行在更高的层次上，相对消耗资源也少很多。

主机虚拟化中VM的应用程序调用硬件资源时需要经过:VM内核->Hypervisor->主机内核，导致性能是三种虚拟化技术中最差的。主机虚拟化技术代表是VMware Server（GSX）、Workstation和Microsoft Virtual PC、Virtual Server等。

裸金属虚拟化Bare-metal

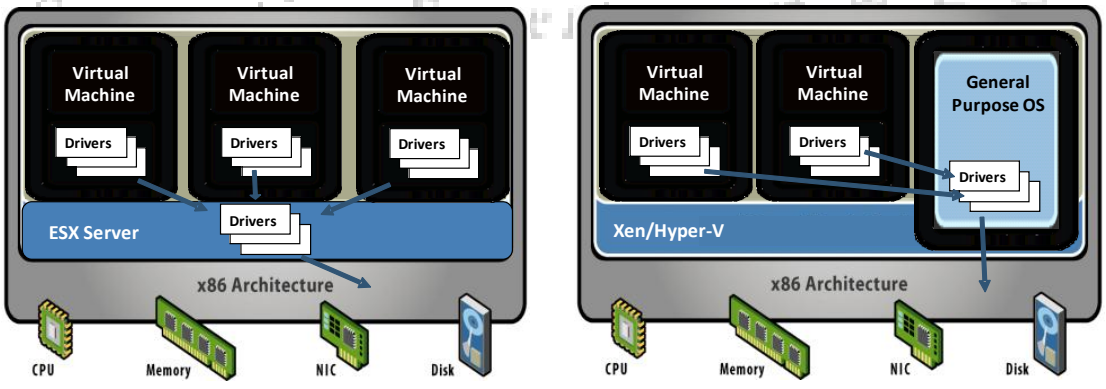
裸金属虚拟化中Hypervisor直接管理调用硬件资源，不需要底层操作系统，也可以理解为Hypervisor被做成了一个很薄的操作系统。这种方案的性能处于主机虚拟化与操作系统虚拟化之间。代表是VMware ESX Server、Citrix XenServer和Microsoft Hyper-V。



OS-Level　　Hosted　　Bare-Metal

上图描述了三种虚拟化方案的形态区别。当前分散云数据中心服务器虚拟化使用的主要是Bare-Metal方案。分散云给数据中心网络带来了新的挑战，虚拟机之间的数据通信管理需求促使了一系列网络新技术的发展。在OS-Level与Hosted方案中，虚拟机都是架设于操作系统之上的，因此VM/VE之间的通信主要由同样运行于基础操作系统之上的网络交换应用程序来完成。而在最主流的Bare-Metal结构中，由于Hypervisor薄操作系统的引入，性能、管理、安全和可靠性等多维度的考虑，造成VM间网络通信管理发展出不同的技术道路（EVB与BPE），后文会对这些技术方向加以详述。

VMware ESX与Xen/Hyper-V的Bare-Metal方案实现结构有所不同，简单如下图所示。



VMware ESX ***Full-Virtualization***　　**XEN /Hyper-V** ***Para-Virtualization***

分散云除了给网络带来上述的VM通信问题，同样由于其对服务器硬件能力的极端榨取，造成网络中的流量压力增大，与集中云一样存在着带宽扩展的需求。原本一台服务器一个操作系统跑一个应用只需要10M流量带宽就够了，现在装了10个VM跑10个应用，带宽可能就需要100M了。

大型机与小型机的一虚多技术早在30年前IBM就做出来了，现在RISC平台上已经相当

完善了，相比较而言X86架构的虚拟化才处于起步阶段，但X86架构由于性价比更高成为了分散云计算的首选。

X86架构最早期是纯软件层面的Hypervisor提供虚拟化服务，缺陷很多，性能也不够，直到2006年Intel推出了实现硬件辅助虚拟化的VT技术CPU产品后才开始迅猛发展（AMD也跟着出了VM技术）。硬件辅助虚拟化技术主要包括CPU/Chipset/Network Adapter等几个方面，和网络技术紧密相关的就是网卡虚拟化了，后文会对如SR-IOV等网卡虚拟化技术应用进行更具体分析。随着2007年Intel VT FlexMigration技术的推出，虚拟机迁移成为可能，2009年Intel支持异构CPU间动态迁移再次向前迈进。

vMotion

这里再多唠叨几句vMotion技术。vMotion是VMware公司提出的虚拟机动态迁移技术名称（XEN也有相应的XENMotion技术），由于此名称被喊得最早，范围最广，认知度最高，因此下文提到虚拟机迁移技术时大都会使用vMotion来代称。

先要明确vMotion是一项资源管理技术，不是高可靠性技术，如果你的某台服务器或VM突然宕机了，vMotion是无助于应用访问进行故障切换和快速恢复的。vMotion是将一个正常的处于服务提供中的VM从一台物理服务器搬家到另一台物理服务器的技术，vMotion的目的是尽可能方便的为服务管理人员提供资源调度转移手段，当物理服务器需要更换配件关机重启啦，当数据中心需要扩容重新安排资源啦，这种时候vMotion就会有用武之地了。

设想一下没有vMotion上述迁移工作是怎么完成的，首先需要将原始物理服务器上的VM关机，再将VM文件拷贝到新的物理服务器上，最后将VM启动，整个过程VM对外提供的服务中断会达到几分钟甚至几小时的级别。而且需要来回操作两台物理服务器上的VM，对管理人员来说也很忙叨。

使用vMotion后，两台物理服务器使用共享存储来保存VM文件，这样就节省了上述步骤2中的时间， vMotion只需在两台物理服务器间传递当前的服务状态信息，包括内存和TCP等上层连接表项，状态同步的拷贝时间相对较短，而且同步时原始VM还可以提供服务使其不会中断。同步时间跟VM当前负载情况及迁移网络带宽有关，负载大了或带宽较低使同步时间较长时，有可能会导致vMotion出现概率性失败。当状态同步完成后，原始物理服务器上的VM会关闭，而新服务器上的VM激活（系统已经在状态同步前启动完毕，始终处于等待状态），此时会有个较短的业务中断时间，可以达到秒级。再者vMotion是通过VMware

的vCenter管理平台一键化完成的，管理人员处理起来轻松了许多。

这里要注意vMotion也一定会出现业务中断，只是时间长短区别，不要轻易被一些宣传所忽悠。想想原理，不论怎么同步状态，只要始终有新建发生，在同步过程中原始服务器上新建立的客户连接，新服务器上都是没有的，切换后这部分连接势必被断开重建，零丢包只能是理想值。VMware也同样建议将vMotion动作安排在业务量最少的时候进行。

vMotion什么场景适用呢？首先肯定得是一虚多的VM应用场景，然后是对外业务中断恢复的可靠性要求极高，一般都是7*24小时不间断应用服务才用得上，最后是计算节点规模始终在不断增长，资源调度频繁，管理维护工作量大的数据中心。

另外共享存储这个强制要求会给数据中心带来了整体部署上的限制，尤其是下面提到的跨数据中心站点vMotion时，跨站点共享存储的问题解决起来是很麻烦的，由于这部分内容和网络关系不大，属于存储厂商的地盘，对跨站点共享存储技术有兴趣的读者可以参考EMC/IBM等厂商的资料看看，本文就不过多介绍了。

vMotion的出现推动了数据中心站点间大二层互联和多站点动态选路的网络需求，从而导致OTV和LISP等一系列新网络技术的出现。

2.3 云计算小结

通过前面的描述，希望大家能对云计算有个较为清晰的概念。云计算还有一大块内容是平台管理资源调度方面（目前很多厂家吆喝的云计算都是云平台）。这部分主要针对客户如何更便捷的创建与获取虚拟化服务资源，实际过程就是用户向平台管理软件提出服务请求，管理平台通过应用程序接口API（Application Program Interface）将请求转化为指令配置下发给服务器、网络、存储和操作系统、数据库等，自动生成服务资源。需要网络做的就是设备能够识别管理平台下发的配置，从技术创新的角度讲，没有啥新鲜东西，就不多说了。当前的云平台多以IaaS/PaaS为主，能做到提供SaaS的极少。但在今后看来，SaaS将会成为云服务租用主流，中小企业和个人可以节省出来IT建设和维护的费用，更专注于自身的业务发展。

总结一下云计算给数据中心网络带来的主要变化：

1、 更高的带宽和更低的延迟

2、 服务器节点（VM）规模的增加

3、 VM间通信管理

4、 跨数据中心站点间的二层互联以承载vMotion

题外再多说两句，计算虚拟化中一虚多与多虚一结合使用才是王道。但目前云计算服务提供商能够提供的只是先将物理服务器一虚多成多台VM，再通过LB/集群计算等技术将这些VM对外多虚一成一个可用的资源提供服务。个人感觉，如果能做到先将一堆物理服务器虚拟成一台几万个核Super Computer，用户再根据自己的需要几个几十个核的自取资源，这样才更有云计算的样子， Super Computer就是那朵云。当然计算虚拟化的时候不光是核的调配，还要包括IO/Memory等一起进行调度，这里只是简单举例。

3 数据中心

数据中心的产生有多早？从人类开始将信息记录到介质上传递开始就有了数据中心，那个记载信息的介质（石头或树皮）就是数据中心，不过那时的网络是靠手手相传而已。如果更甚一些，可以理解人类产生语言开始，知识最多的人（酋长/祭祀）就是数据中心，口口相传就相当于现如今的网络传输。有人该说，夸张了哈，写作手法而已，只是想突出一下数据中心的重要性。

当计算机网络连接到Server的那一刻起，整个世界的网络就从网状变成了树状，一个个数据中心就是网络世界的根。

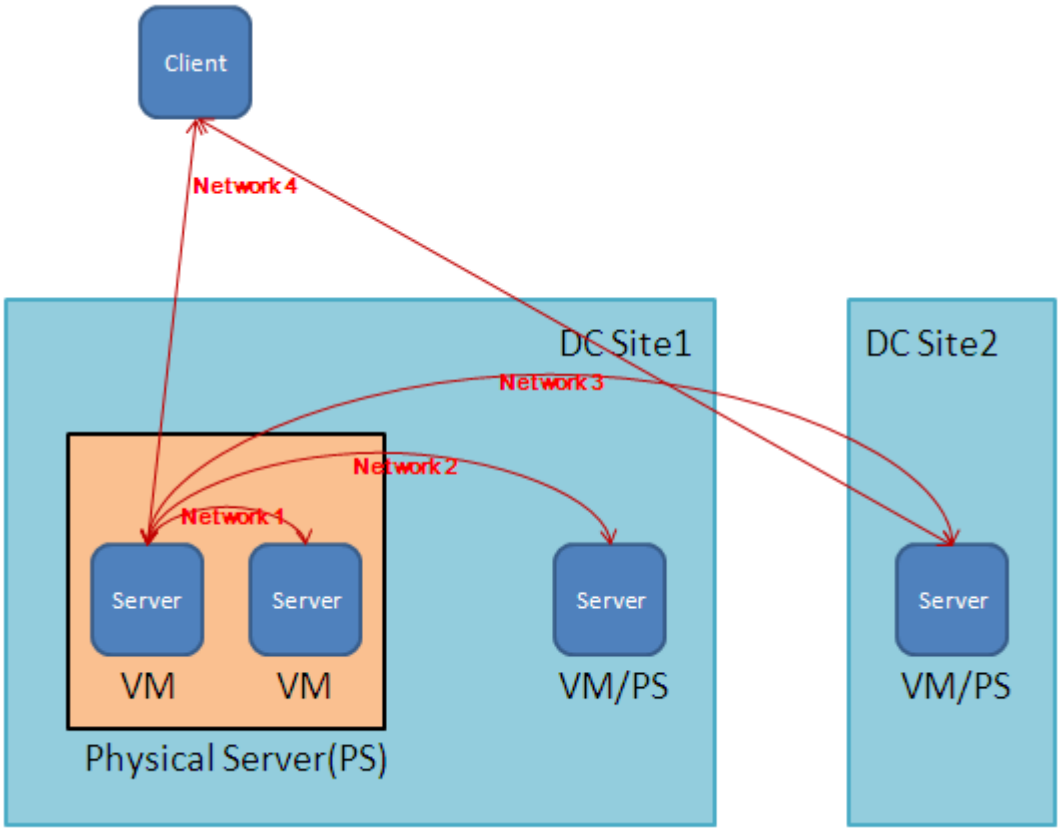
3.1 Client 与 Server

在所有的数据通信会话中，只有两个永恒的角色，Client与Server。为了下文叙述简便，作者把数据中心内部的终端统一称之为Server，数据中心外部的为Client。这样网络间的流量通信就只剩下Client-Server（CS）与Server-Server（SS）两种了。其实更准确说还是只有CS一种，SS通信也是有个发起方和响应方的。QQ/MSN等即时通信软件的流量模型实际可理解为CSC；唯有P2P对CS结构有所颠覆，但不管怎么处理也必定会存在Server角色进行最初的调度。

所有数据中心需要处理的业务就是CS和SS两种，CS肯定是基于IP进行L3转发的了，SS则分为基于IP的L3和基于MAC的L2两种转发方式。基于IP的SS通信主要是不同业务间的数据调用，如WEB/APP服务器去调用DB服务器上的数据，再如有个员工离职，职工管理系统会同步通知薪酬管理、考勤管理、绩效管理等一系列系统进行删除信息的相关操作。基于MAC的SS通信则是同一类服务器间的数据同步计算，比如使用WEB集群分流用户访问时，

需要对修改或增删的数据进行集群同步；再比如多虚一中集群一起计算任务时协调者和执行者之间的大量通信进行任务调度。

可以看出云计算数据中心给网络带来的挑战主要是基于MAC的二层（OSI模型）SS通信。在一虚多技术影响下，Server的概念已经扩展到以单台VM为基础单元，因此可以引出下面这个图，看看新网络技术是如何划分的。



Network1：VM到VM之间的SS二层互联网络

Network2：DC站点内部SS二层互联网络

Network3：跨DC站点间的SS二层互联网络

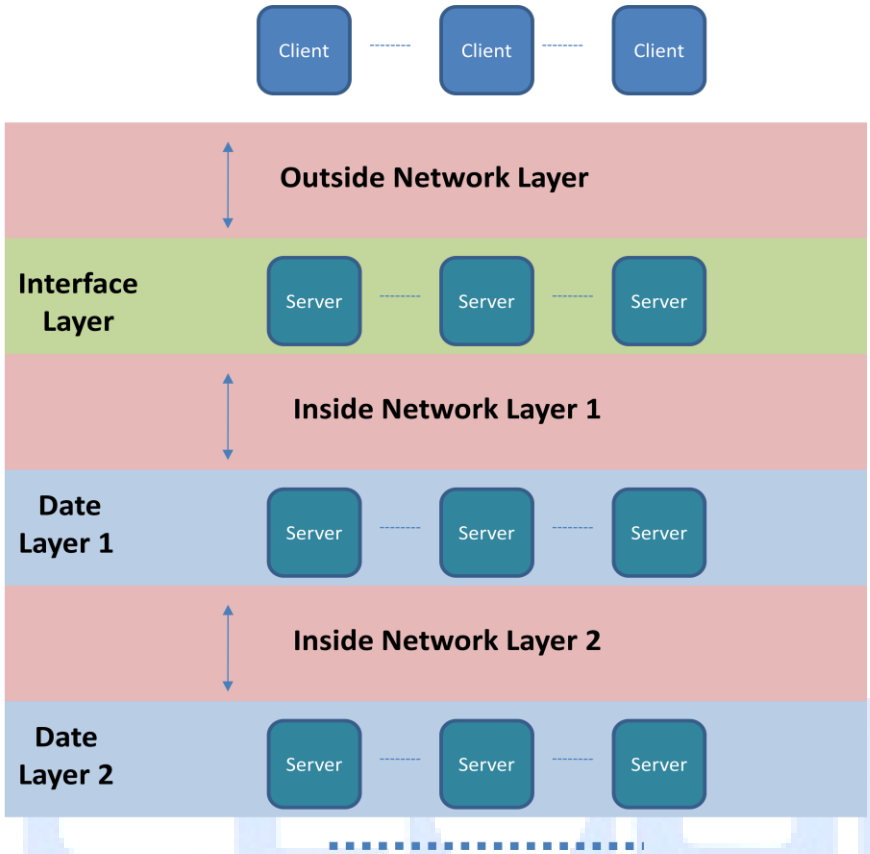
Network4：DC到Client之间的CS三层互联网络

后文的技术章节就会针对这些部分进行展开，详细说下都有哪些技术分别对应在这四段网络中，这些技术的特点是什么。

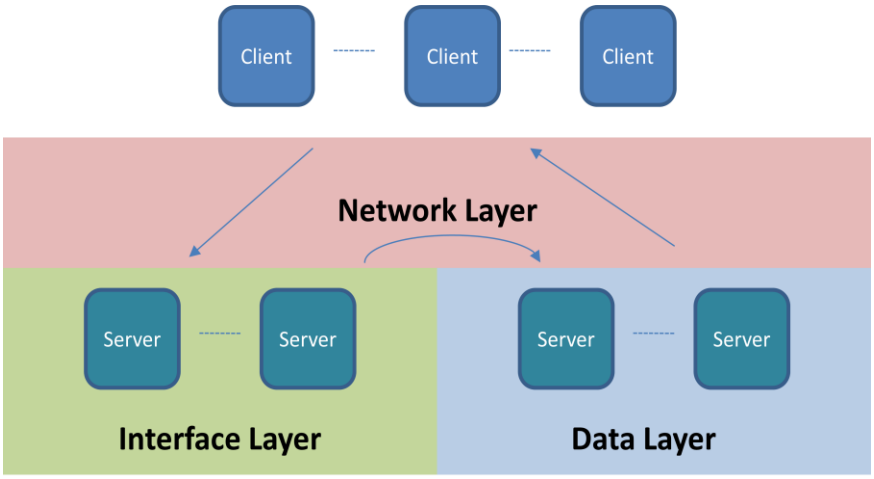
3.2 层次化与扁平化

数据中心的网络结构取决于应用计算模型，计算模型主要分为层次化与扁平化两种结构。层次化结构如下图所示，典型的应用如WEB-APP-DB、搜索引擎或高性能计算（地震、科

研）等。特点是客户请求计算结果必须逐层访问，返回数据也要逐层原路返回。

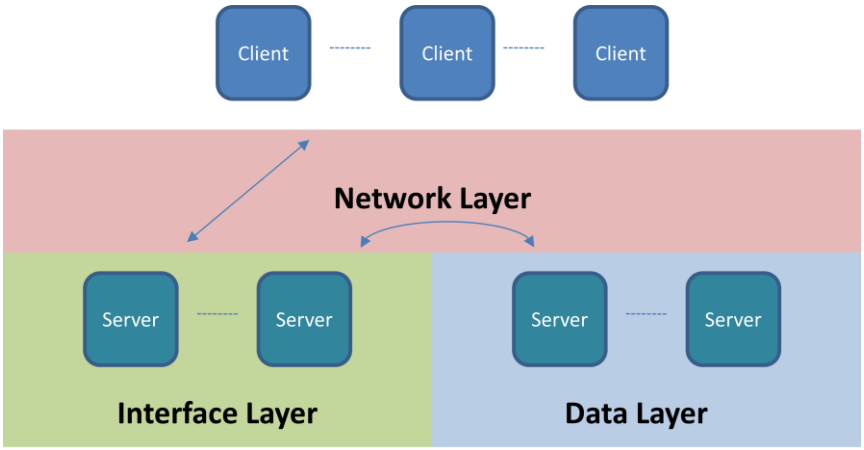


计算模型扁平化结构如下图所示，特点是数据层服务器会将结果直接返回给客户，不需要再由接口层服务器进行处理，也有管这种模型叫做三角传输的。典型的应用如一些Internet网站服务采用的LB结构，LB服务器就是只做调度，WEB服务器会直接将请求结果返回给用户。



注意，上面说的是计算模型，和网络模型并不是一一对应，采用层次化结构计算模型一

样可以进行扁平化组网，如下图所示。

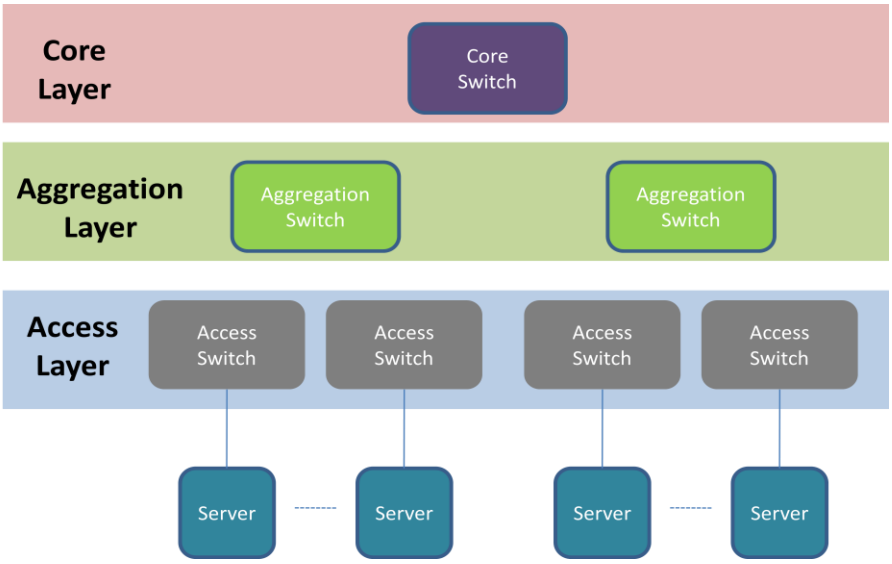


从网络角度讲，扁平化相比较层次化结构最大的好处是可以减少服务器的网卡接口数量（省钱），然而缺点是没有清晰的层次，部署维护的复杂度就会相应提升。总体来说，当前数据中心实际组网建设中，这两种方式谁都没占据到绝对优势，上哪种结构完全看规划者的考量重点是在哪个方面。

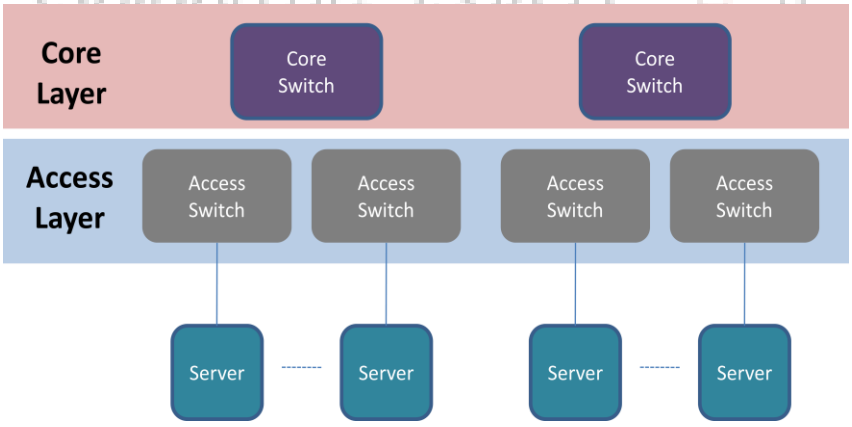
前面说过，云计算主要分为多虚一与一虚多两种虚拟化结构。一虚多对传统计算模型没有太大影响，只是将其服务器从物理机到虚拟机数量规模扩大了N倍，网络规模也随之进行扩大。而多虚一中，协调者角色对应了接口层服务器，执行者角色则对应数据层服务器，由于此时大量的通信交互是在不同执行者之间或执行者与协调者之间，需要重点关注的大规模网络就由原来的接口层服务器之前，转移到了接口层服务器与数据层服务器之间。

3.3 三层结构与两层结构

在以往的数据中心网络建设时，关注的重点都是指接口层服务器前的网络，传统的三层网络结构如下图所示。其中的汇聚层作为服务器网关，可以增加防火墙、负载均衡和应用加速等应用优化设备。

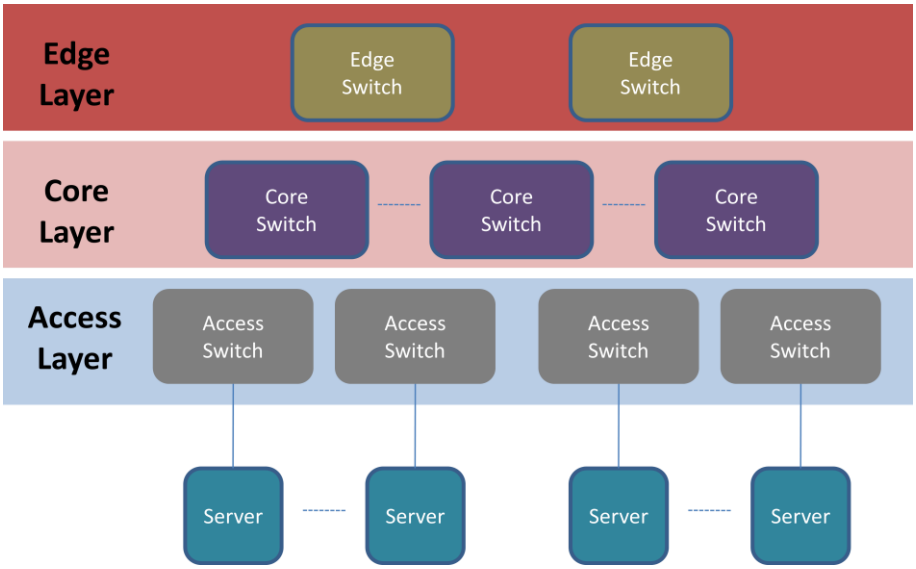


但在云计算数据中心里面Ethernet网络规模扩大，流量带宽需求增加，因此不会在网络中间位置再插入安全和优化设备了，转发性能太低，上去就是瓶颈，汇聚层的位置也就可有可无了。再加上带宽收敛比的问题，短期内大型云计算数据中心网络里面不会出现汇聚层的概念。以前是百兆接入、千兆汇聚、万兆核心，现在服务器接入已经普及千兆向着万兆迈进了，除非在框式交换机上40G/100G接口真的开始大规模部署，还有可能在云计算数据中心里面再见到超过两层的级联结构网络。现如今的云计算数据中心流行的都是如下图所示的千兆接入，万兆核心的两层网络结构。

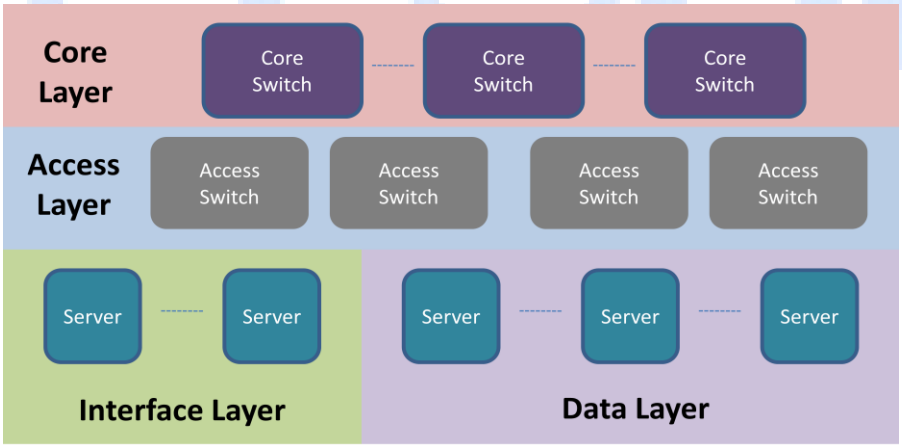


此两层网络结构部署在接口层服务器之前，则一般会将服务器网关部署在Core Switch上，但前提是网络规模不会太大，Core不会太多（2个就差不多了），否则VRRP/HSRP等多网关冗余协议只能走到一个活动网关，会导致网络效率很低。还有一种方式是将服务器网关部署在Access Switch上，Access SW与Core SW之间通过OSPF等动态路由协议达到全互联，使用等价路由达到多Core SW的负载均担。但此方式的缺点是L3路由交互转发效率较低，部

署复杂且占用大量IP地址。在未来的TRILL/SPB等二层Ethernet技术结构中，可能会出现专门作为网关与外部进行IP层面通信用的边缘交换机（由于出口规模有限，2-4台足够处理），内部的Core SW只做二层转发，可以大规模部署以满足内部服务器交互的需求，如下图所示。



当遇到多虚一高性能计算的模型，则二层网络结构会被部署在接口服务器与数据服务器之间，为二者构建纯二层的大规模交互网络，结构如下图所示。



3.4 Server 与 Storage

前面说的CS/SS网络可以统称为数据中心前端网络，目前和以后基本上都是IP+Ethernet一统天下（IB Infiniband只能吃到高性能计算的一小口）。有前端当然就有后端，在数据中心里面，服务器与存储设备连接的网络部分统称为数据中心后端网络。就目前和短期的未来来看，这块儿都是FC的天下。

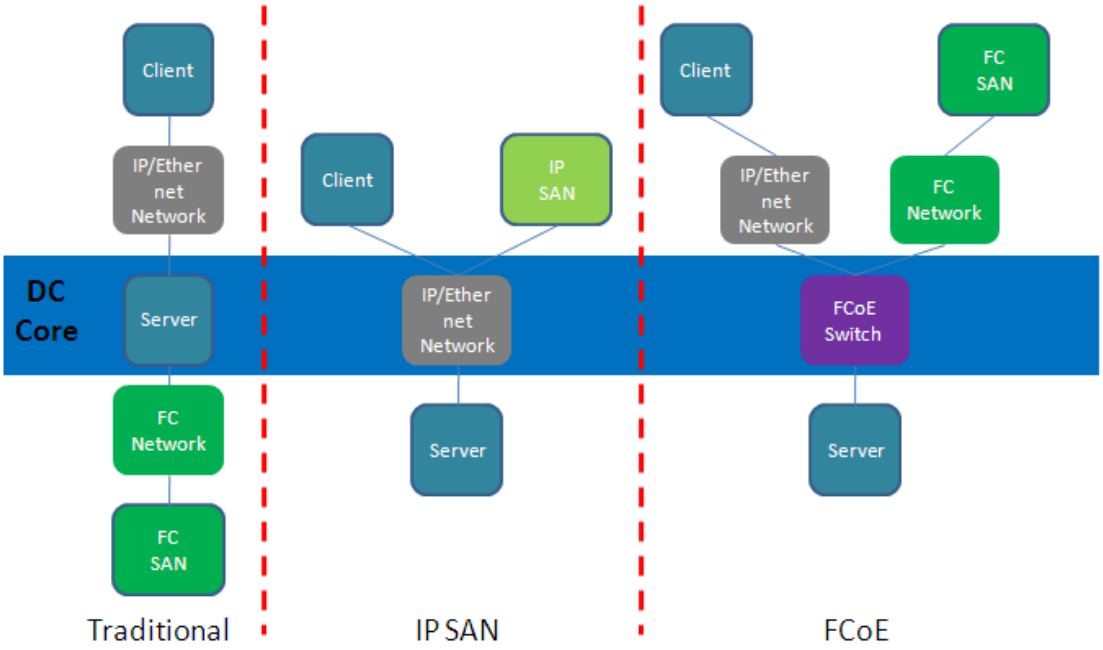
简单说两句存储，DAS（Direct Attached Storage）直连存储就是服务器里面直接挂磁盘，

NAS（Network Attached Storage）则是网络中的共享文件服务器，此二者大多与数据中心级别存储没什么关系。只有SAN（Storage Area Network）才是数据中心存储领域的霸主，磁盘阵列会通过FC或TCP/IP网络注册到服务器上模拟成直连的磁盘空间。而目前FC SAN是主流中的主流，基于TCP/IP的IP SAN等都是配太子读书的角色。

在服务器到存储的后端网络中，涉及到IO同步问题，高速、低延迟与无丢包是对网络的基本需求，而Ethernet技术拥有冲突丢包的天然缺陷，FC的无丢包设计使其领先一步，加上早期Ethernet还挣扎在100M带宽时，FC已经可以轻松达到2G，所以在后端网络中从开始到现在都是FC独占鳌头。但是从发展的眼光看，Ethernet目前已经向着40G/100G迈进，而FC的演进并不理想，无论是BASE10的10/20/40G路线（主要用在FC交换机之间，目前基本已经被淘汰）还是BASE2的2/4/8/16/32G路线（当前主流FC应用）都已经落后，加上各种以太网零丢包技术（CEE/DCE/DCB）的出现，以后鹿死谁手还真不好说。

在目前阶段，为了兼容数据中心已有的主流FC网络和存储设备，在基于iSCSI技术的IP SAN技术没能开花结果的情况下，众多Ethernet网络厂商又推出了FCoE来蚕食服务器到存储这块蛋糕。下文技术章节会专门介绍FCoE的内容。

先简单说下，FCoE没有惦着像IP SAN那样一下子完全取代FC去承载后端网络，而是走前后端网络融合，逐步蚕食的路线，是网络厂商们将数据中心的核心由服务器向网络设备转移的重要武器。如下图，就是看谁做太阳，谁做星星。相比较IP SAN的壮烈牺牲，FCoE采用了一条更为迂回的兼容道路，目前已经出现了支持FCoE的存储设备，也许Ethernet完全替代FC的时代真的能够到来。



如果FCoE能成功，虽然短期内交换机、服务器和存储的价格对比不会有太大的变化，但是占据了核心位置，对未来的技术发展就有了更大的话语权，附加值会很高。又如当前的EVB（Edge Virtual Bridging）和BPE（Bridging Port Extend）技术结构之争也同样是话语权之争。

顺便一提，当一项完全不能向前兼容的全新技术出现时，除非是有相当于一个国家的力量去推动普及，而且原理简单到8-80岁都一听就明白，否则注定会夭折，与技术本身优劣无太大关系。老话说得好，一口吃不成胖子。

3.5 数据中心多站点

这是个有钱人的话题，且符合2-8原则，能够建得起多个数据中心站点的在所有数据中心项目中数量也许只能占到20%，但他们占的市场份额肯定能达到80%。

建多个数据中心站点主要有两个目的，一是扩容，二是灾备。

扩容

首先说扩容，一个数据中心的服务器容量不是无限的，建设数据中心时需要考虑的主要因素是空间、电力、制冷和互联。数据中心购买设备场地建设只是占总体耗费的一部分，使用过程中的耗能维护开销同样巨大，以前就闹过建得起用不起的笑话。当然现在建设时要规范得多，考虑也会更多，往往做预算时都要考虑到10年甚至以上的应用损耗。

再讲个故事，以前曾有某大型ISP打算找个雪山峡谷啥的建数据中心，荒郊野外空间本

来就大，融雪制冷，水力发电，听上去一切都很美，但是就忘了一件事，互联。光纤从哪里拉过去，那么远的距离中间怎么维护，至少从目前阶段来说这个问题无解。也许等到高速通信发展到可以使用类似铱星的无线技术搞定时，数据中心就真的都会建到渺无人烟的地方吧，现在还只能在城市周边徘徊。貌似听说过国外有建得比较偏远的大型数据中心，但个人觉得应该还是人家通信行业发达，光纤资源丰富，四处都能接入。但至少目前国内的运营商们不见得会支持，大城市周边搞搞就算了，远了没人会陪你玩。

有些扯远，回到正题。现在国内已经有超过10k台物理服务器在一个数据中心站点的项目了，再多我还没有听说过。只有几百上千的物理服务器就敢喊搞云计算是需要一定勇气的，既然是云，规模就应永无止境。所以建多个数据中心站点来扩容就成了必然之举。这时就可能遇到Cluster集群计算任务被分配在多个站点的物理服务器或虚拟机来完成的情况，从而提出了跨多个数据中心站点的Ethernet大二层互联需求。

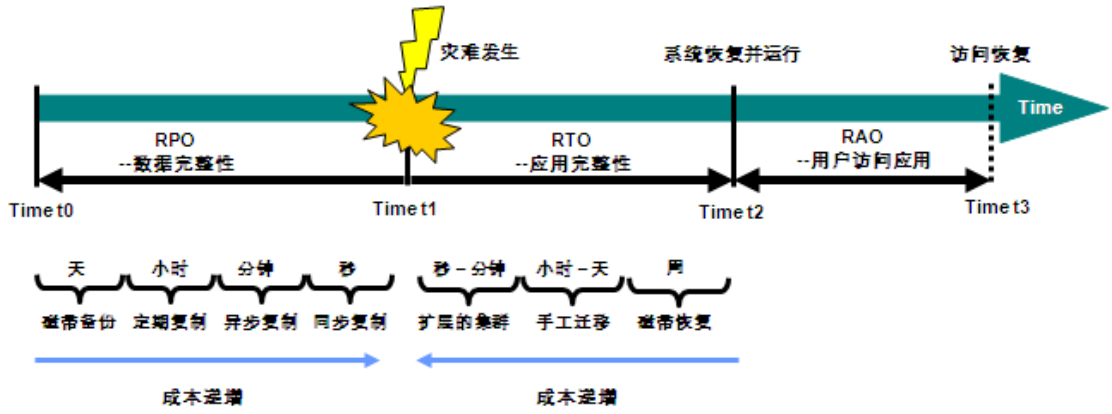
在扩容时，就可以充分利用vMotion等虚拟机迁移技术来进行新数据中心站点的建设部署，同样需要站点间的大二层互通。支持IP层的vMotion目前虽然已经出现，但由于技术不够成熟，限制很多，实用性不强，还是以Ethernet二层迁移技术为主。

灾备

再说说灾备，最近几年天灾人祸着实不少，数据中心容灾就越来越受到重视。扩容和灾备的主要区别就是：扩容的多个站点针对同一应用都要提供服务；而灾备则只有主站点提供服务，备份站点当主站点挂掉的时候才对外服务，平时都处于不运行或者空运行的状态。

参考国标《信息系统灾难恢复规范》GB/T 20988－2007，灾备级别大致可划分为数据级别（存储备份），应用级别（服务器备份），网络级别（网络备份），和最高的业务级别（包括电话、人员等所有与业务相关资源）。

国内外统一的容灾衡量标准是RPO（Recovery Point Objective）、RTO（Recovery Time Objective）和RAO（Recovery Access Objective）了，通过下图形象一些来体现他们的关系。



简单来说RPO衡量存储数据恢复，RTO衡量服务器应用恢复，RAO衡量网络访问恢复。一般来说RPO设计都应小于RTO。国外按照RTO/RPO的时间长短对灾难恢复分级参考由高到低为：

Class 1/A　　RTO=0-4 hrs; RPO=0-4 hrs

Class 2/B　　RTO=8-24 hrs; RPO=4 hrs

Class 3/C　　RTO=3 day; RPO=1 day

Class 4/D　　RTO=5+ days; RPO=1 day

标准归标准，真正建设时候最重要的参考条件还是应用的需求，像银行可以直接去调研储户能容忍多长时间取不出来钱，腾讯去问问QQ用户能容忍多长时间上不了线，就都知道该怎么设计容灾恢复时间了。

真正在玩多中心灾备的行业，国内集中在金融系统（尤其是银行），政府和能源电力等公字头产业，国外的不太清楚，但我想以盈利为主要目的企业不会有太强烈意愿去建设这种纯备份的低效益站点，更多的是在不同站点内建设一些应用服务级别的备份，所有站点都会对外提供服务。

小结

在云计算规模的数据中心中，对于LB类型的多虚一集群技术，执行者（概念参见文档前面集中云部分）少上几个不会影响全局任务处理的，只要在扩容时做到数据中心间大二层互通，所有站点内都有计算任务的执行者，并且配合HA技术将协调者在不同站点做几个备份，就已经达到了应用容灾的效果。针对一虚多的VM备份，VMware/XEN等都提出了虚拟机集群HA技术，此时同样需要在主中心站点与备份中心站点的服务器间提供二层通道以完成HA监控管理流量互通，可以达到基于应用层面的备份。

云计算数据中心多站点主要涉及的还是扩容，会部署部分针对VM做HA的后备服务器，但是不会搞纯灾备站点。针对多站点间网络互联的主要需求就是能够做而二层互联，当站点数量超过两个时所有站点需要二层可达，并部署相关技术提供冗余避免环路。

3.6 多站点选择

数据中心建设多站点后，由于同一应用服务可以跑在多个站点内部，对Client来说就面临着选择的问题。

首先要记住的是一个Client去往一个应用服务的流量必须被指向一台物理或虚拟的Server。你可以想象一个TCP请求的SYN到ServerA，而ACK到了ServerB时，ServerA和B为了同步会话信息都会疯掉。想办法维持一对Client-Server通信时的持续专一是必须的。

Client到Server的访问过程一般分为如下两步：

1、 Client访问域名服务器得到Server IP地址（很少人会去背IP地址，都是靠域名查找）

2、 Client访问Server IP，建立会话，传递数据。

当前的站点选择技术也可以对应上面两个步骤分为两大类。

第一类是在域名解析时做文章，原理简单来说就是域名服务器去探测多个站点内IP地址不同的服务器状态，再根据探测结果将同一域名对应不同IP返回给不同的Client。这样一是可以在多个Client访问同一应用时，对不同站点的服务器进行负载均担，二是可以当域名服务器探测到主站点服务器故障时，解析其他站点的服务器IP地址给Client达到故障冗余目的。这时要求不同站点的服务地址必须在不同的三层网段，否则核心网没法提供路由。缺点很明显，对域名解析服务器的计算压力太大，需要经常去跟踪所有服务器状态并Hash分配Client请求的地址。此类解决方案的代表是F5/Radware/Cisco等厂商的3DNS/GSLB/GSS等技术。

第二类就是把多个站点的服务IP地址配置成一样，而各个站点向外发布路由时聚合成不同位数的掩码（如主中心发布/25位路由，备中心发布/24位路由），或通过相同路由部署不同路由协议Cost值以达到主备路由目的。使用掩码的问题是太细则核心网转发设备上的路由数量压力大，太粗则地址使用不好规划很浪费。使用Cost则需要全网IP路由协议统一，节点规模受到很大限制。另外这种方式只能将所有Client访问同一服务IP的流量指向同一个站点，负载分担只能针对不同的服务。好处则是这种站点选择技术谁都能用，不需要专门设备支持，部署成本低成为其存活的根据。

在云计算大二层数据中心部署下，各个站点提供同一服务的Server都处于一个二层网络

内，且不能地址冲突，与前面描述的两种站点选择技术对服务器IP设置要求都不匹配，因此需要配合SLB设备一起使用。可以理解其为一种基于IP粒度的多虚一技术，使用专门LB硬件设备作为协调者，基于IP地址来分配任务给服务组中不同的Server执行成员。LB设备通常将多个Server对应到一个NAT组中，外部访问到一个NAT Server虚拟IP地址，由LB设备按照一定算法分担给各个成员。LB设备同时会探测维护所有Server成员状态。当各个站点内LB设备将同一服务对外映射为不同的虚拟IP地址时，可以配合域名解析方式提供Client选路；而配置为相同时则可以配合路由发布方式使用。

现有的站点选择技术都不尽如人意，即使是下文介绍的Cisco新技术LISP也只是部分的解决了路由发布技术中，发布服务器地址掩码粒度过细时，给核心网带来较大压力的问题，目前还不算是一套完整的站点选择解决方案。个人感觉，最好的路还是得想法改造DNS的处理流程，目前的DNS机制并不完备，在攻击面前脆弱不堪，后面的安全附加章节中会对此再深入讨论。

3.7 数据中心小结

又到了小结部分，云计算数据中心相比较传统数据中心对网络的要求有以下变化：

1、 Server-Server流量成为主流，而且要求二层流量为主。

2、 站点内部物理服务器和虚拟机数量增大，导致二层拓扑变大。

3、 扩容、灾备和VM迁移要求数据中心多站点间大二层互通。

4、 数据中心多站点的选路问题受大二层互通影响更加复杂。

题内话，FCoE并不是云计算的需求，而是数据中心以网络为核心演进的需求，至于云计算里面是不是一定要实现以网络为核心，就看你是站在哪个设备商的角度来看了。

4 网络

说到网络，这里关注的重点是前文提到的数据中心内部服务器前后端网络，对于广泛意义上的数据中心，如园区网、广域网和接入网等内容，不做过多扩散。

4.1 路由与交换

网络世界永远的主题，至少目前看来还没有出现能取代这二者技术的影子，扩展开足够写好几本书的了。

数据中心的网络以交换以太网为主，只有传统意义的汇聚层往上才是IP的天下。参考前文的需求可以看出，数据中心的以太网络会逐步扩大，IP转发的层次也会被越推越高。

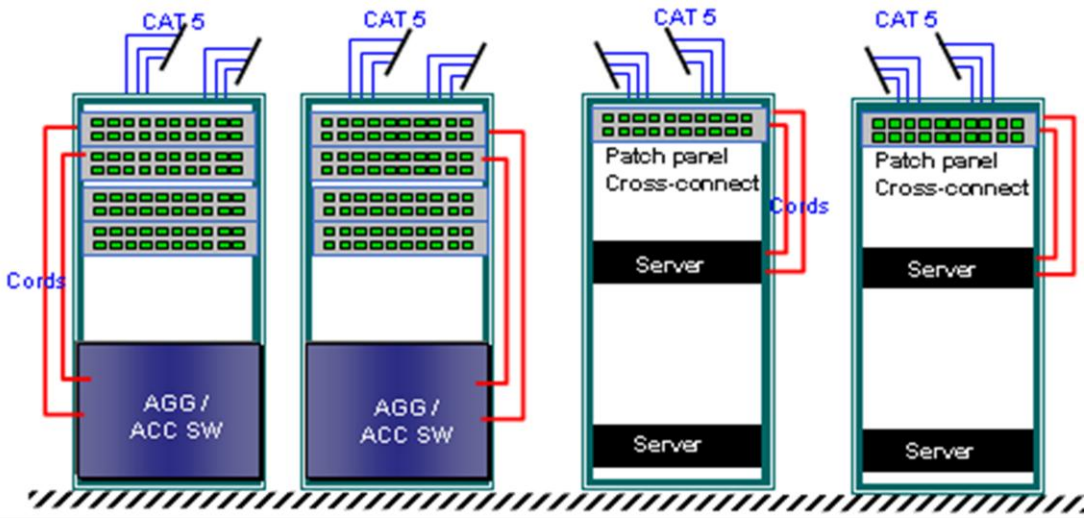
数据中心网络从设计伊始，主要着眼点就是转发性能，因此基于CPU/NP转发的路由器自然会被基于ASIC转发的三层交换机所淘汰。传统的Ethernet交换技术中，只有MAC一张表要维护，地址学习靠广播，没有什么太复杂的网络变化需要关注，所以速率可以很快。而在IP路由转发时，路由表、FIB表、ARP表一个都不能少，效率自然也低了很多。

云计算数据中心对转发带宽的需求更是永无止境，因此会以部署核心-接入二层网络结构为主。层次越多，故障点越多、延迟越高、转发瓶颈也会越多。目前在一些ISP（Internet Service Provider）的二层结构大型数据中心里，由于传统的STP需要阻塞链路浪费带宽，而新的二层多路径L2MP技术还不够成熟，因此会采用全三层IP转发来暂时作为过渡技术，如前面提到过的接入层与核心层之间跑OSPF动态路由协议的方式。这样做的缺点显而易见，组网复杂，路由计算繁多，以后势必会被Ethernet L2MP技术所取代。

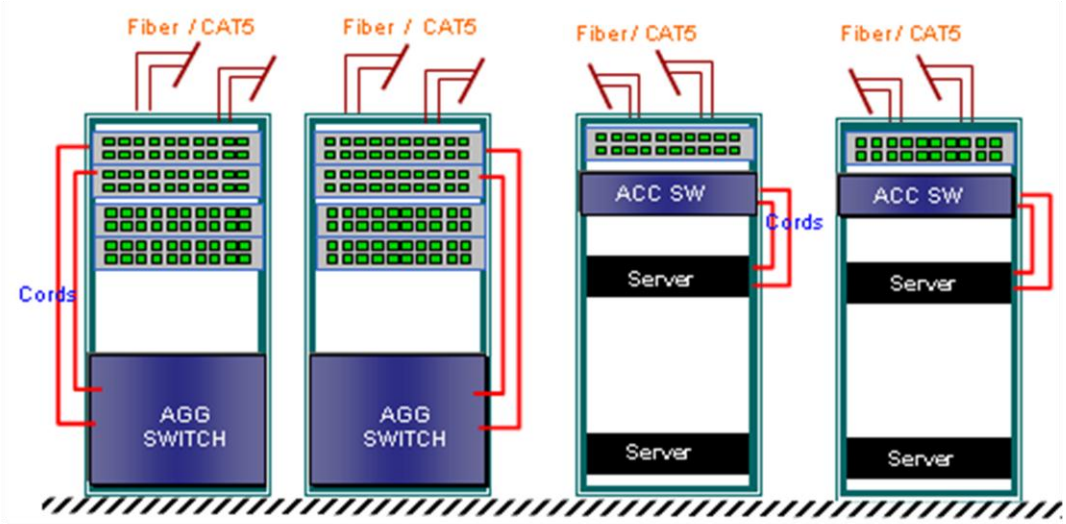
新的二层多路径技术会在下文做更详细的介绍，不管是TRILL还是SPB都引入了二层ISIS控制平面协议来作为转发路径计算依据，这样虽然可以避免当前以太网单路径转发和广播环路的问题，但毕竟是增加了控制平面拓扑选路计算的工作，能否使其依然如以往Ethernet般高效还有待观察。MPLS就是一个尴尬的前车之鉴，本想着帮IP提高转发效率而生根发芽，没想到却在VPN路由隔离方面开花结果了，世事难料啊。

4.2 EOR 与 TOR

前面说了，数据中心网络设备就是交换机，而交换机就分为框式与盒式两种。当前云计算以大量X86架构服务器替代小型机和大型机，导致单独机架Rack上的服务器数量增多。受工程布线的困扰，在大型数据中心内EOR（End Of Row）结构已经逐步被TOR（Top Of Rack）结构所取代。盒式交换机作为数据中心服务器第一接入设备的地位变得愈发不可动摇。而为了确保大量盒式设备的接入，汇聚和核心层次的设备首要解决的问题就是高密度接口数量，由此框式交换机也就占据了数据中心转发核心的位置。



End Of Row



Top Of Rack

4.3 控制平面与转发平面

对交换机来说，数据报文转发都是通过ASIC（Application Specific Integrated Circuit）芯片完成，而协议报文会上送到CPU处理，因此可以将其分为控制平面与转发平面两大部分。

控制平面主体是CPU，处理目的MAC/IP为设备自身地址和设备自身发给其他节点的报文，同时下发表项给转发ASIC芯片，安排数据报文的转发路径。控制平面在三层交换机中尤为重要，需要依靠其学习路由转发表项并下发到ASIC芯片进行基于IP的转发处理。而二层交换机中数据报文的转发路径都是靠MAC地址广播来直接学习，和控制平面CPU关系不大。

转发平面则是以ASIC芯片为核心，对过路报文进行查表转发处理，对交换机来说，ASIC转发芯片是其核心，一款交换机的能力多少和性能大小完全视其转发芯片而定。而控制平面CPU虽然也是不可或缺的部分，但不是本文介绍的重点，下文将以分析各类型交换机的转发处理为主。

4.4 Box 与集中式转发

经常看到设备商们今天推出个“高性能”，明天推出个“无阻塞”，后天又搞“新一代”的网络交换产品，各种概念层出不穷，你方唱罢我登台，搞得大家跟着学都学不过来，总有一种是不是被忽悠了的感觉。其实很多时候真的是在被忽悠。

先说说Box盒式设备。盒式交换机从产生到现在，以转发芯片为核心的集中式转发结构就没有过大的变化。集中式转发盒子的所有接口间流量都是走转发芯片来传输，转发芯片就是盒子的心脏。

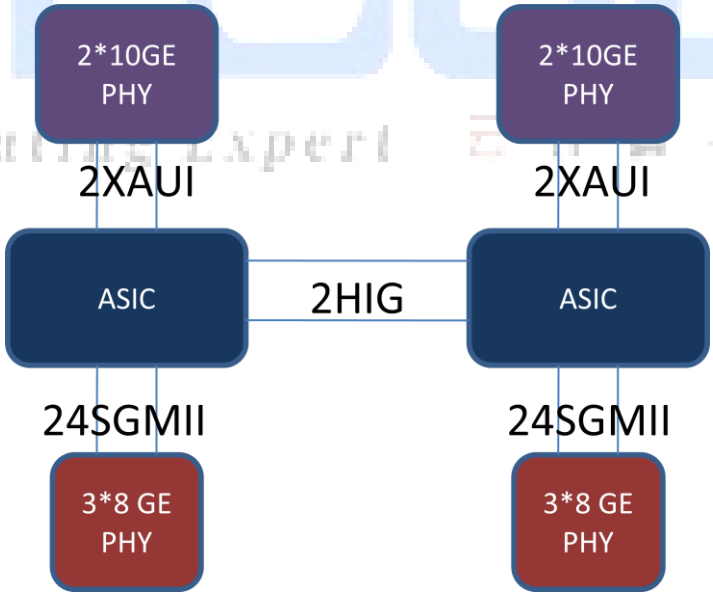
而这个心脏的叫法多种多样，神马Port ASIC、Switch Chip、Fabric ASIC、Unified Port Controller等等都是各个厂家自行其说罢了，关键就看各个物理接口的PHY（将0/1信号与数据互相转换用的器件）连接到哪里，哪里就是核心转发芯片。一般的中小型交换机设备厂商（H3C/中兴/锐捷/ Foundry/Force10等，Juniper目前的数据中心Switch不提也罢，下文会简单说说未来的QFabric）都会直接采购Broadcom和Marvell等芯片生产厂商的产品，只有Cisco和Alcatel等寥寥几家大厂商有能力自己生产转发芯片。但说实话，从转发能力来看这些自产的还真不见得能赶上公用的，人家专业啊。自产的最大好处其实在于方便搞些私有协议封包解包啥的，我的芯片我做主。

下面来说说集中式转发能力的计算，假设一个盒子喊自己的转发能力是x Gbps/y Mpps，x是依靠所有外部接口带宽总和算出来的，如48GE+2*10GE的盒子，转发能力就是单向68GE，双向136GE，一般x都会取双向的值；而y则是整机的包转发能力，y=x*1000/2/8/(64+20)，1000是G到M的转换，2是双向，8是每字节8比特，64是报文最小载荷，20是IP头长。要注意下面的机框式转发就不是这么算的了。大部分盒子的包转发能力还是能够很接近这个理论值的，毕竟能选的转发芯片就那么多，设备厂商在这里自己搞不出太多猫腻来。唯一有可能用来混淆客户的就是用芯片转发能力替代设备接口转发能力作为x值来宣传，绝大部分交换机使用的芯片转发能力是大于所有接口带宽总和的。这时x与y都会比实际的要大一些，但是很明显，芯片再强，没有接口引出来也没用的。所以这里的防忽悠技巧就是数接口数自己加一下。

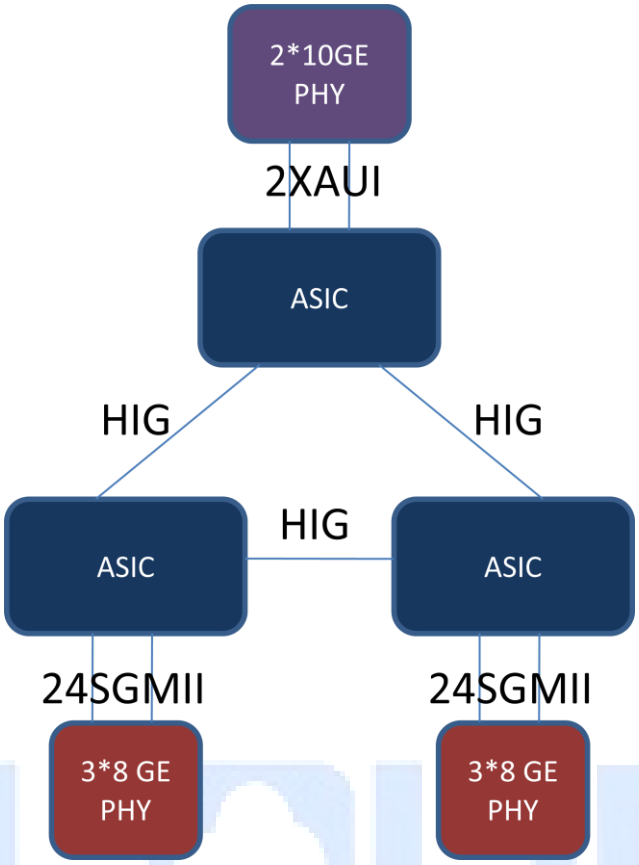
再说结构，决定一款盒式交换机的接口转发容量的是转发芯片，反之你看一款盒子的接口排布情况大概能反推出其使用的芯片能力。转发芯片的接口多种多样（如SGMII、XAUI、HIG、Senders等），但通常每个芯片只连接24个GE接口（8个口一个PHY，3个PHY为一组），因此遇到24GE口交换机，通常都是单芯片的，而48GE或更多就肯定是多芯片的了。而10GE接口的多少要看芯片的能力，个人了解Broadcom有支持24个10GE的转发新片，应该还有能力更强的。现在作者知道的10GE接口密度最高的盒子是Arista的7148SX和Juniper的QFX3500，都支持48个10GE接口，具体布局有待拆机检查。

多芯片交换机还是很考验设备厂商架构设计人员的，既要保证芯片间足够带宽互联无阻塞，又要考虑出接口不能浪费，需拿捏好平衡。所以现在的多芯片盒式交换机设备大多是2-3个转发芯片的，再多的就极少了，芯片间互联设计起来太麻烦了。举两个例子，大家可以看看下面这两种结构有没有问题。

首先是我们能不能用两块6个HIG接口级别转发能力的ASIC（HIG接口带宽12.5GE），设计一款48GE+4*10GE的交换机呢？答案是可以做，但存在结构性拥塞，芯片间至少需要4条HIG才能满足完全无阻塞。

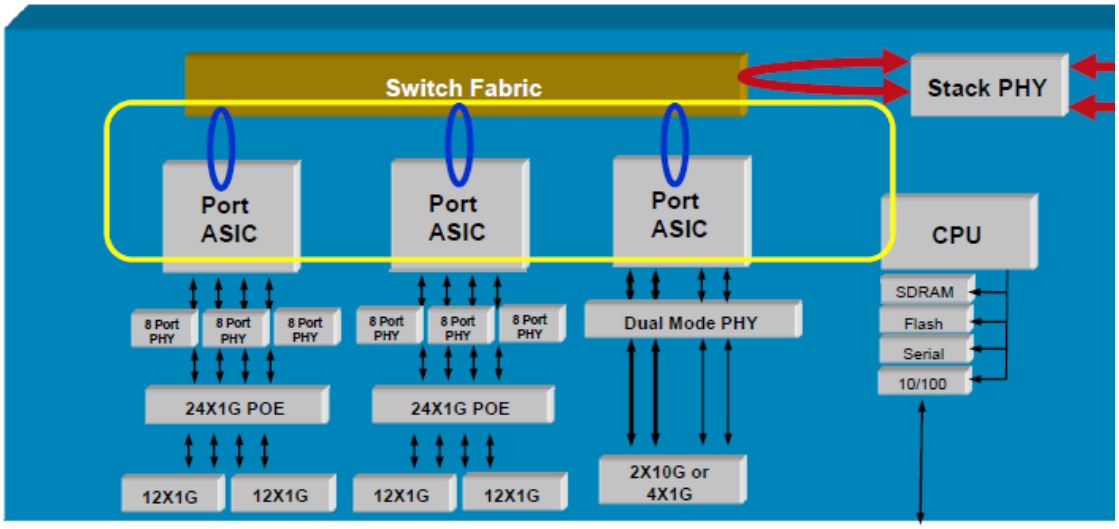


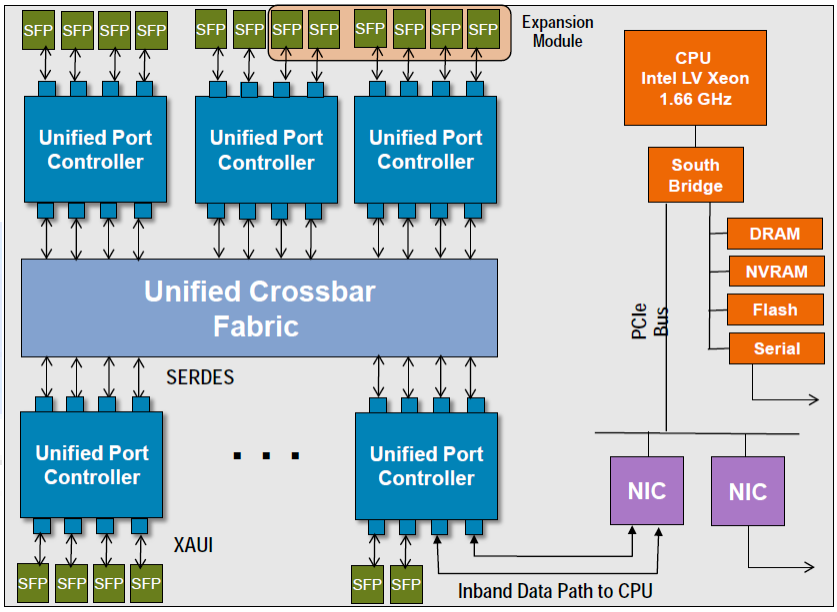
再来看一个，能不能用3块4个HIG接口级别转发能力的ASIC搭建出一款48GE+2*10GE的交换机呢？没有问题，如下图所示内部结构是完全无阻塞的，缺点是部分流量会多绕经1个ASIC转发。



看完了前面这部分，相信大家对盒式交换机都能有个大致了解了，这里只讲讲结构，更详细的转发功能流程就需要有兴趣的童鞋自行去查看下各种芯片手册。另外上述两个例子只为讲解，请勿将当前市场产品对号入座。

刚刚说了，当盒子里面芯片较多的时候连接起来很麻烦，于是出现了新的转发单元Switch Fabric（Cisco N5000上新的名词叫做Unified Crossbar Fabric）。其实这个东东在框式交换机里面很常见，下面会有更详细的介绍。而在盒式交换机里面，目前看到的发布资料使用此种架构的就是Cisco的3750X和N5000了，连接方式如下图所示，这已经接近分布式转发的范围了。





作者将这个Fabric单元叫做交换芯片，便于和前面的ASIC转发芯片区分，二者的主要区别是，交换芯片只处理报文在设备内部的转发，类似Cut-Through，为不同转发芯片间搭建路径，不做过滤和修改。而转发芯片要对报文进行各种查表、过滤和修改等动作，包括缓存都在其中调用，大多是基于Store-Forward方式进行报文处理，是交换机处理数据报文的核心部件。

3750X目前还没有看到进一步的发展需要，而N5000其实是为了Cisco的网络虚拟化架构而服务，不再单单属于传统意义上的Ethernet交换机了。Juniper为QFabric设计的QFX3500接入盒子（48*10GE+4*40GE）估计也是类似于N5000这种带交换芯片的分布式架构。另外怀疑Arista的7148SX也是分布式架构的，应该是6个8*10G的转发芯片通过交换芯片连接，和它的机框式交换机中48*10G接口板布局相同。

总的来说盒子里面搞分布式的最主要原因就是希望提高接口密度，尤其是万兆接口密度，后面相信还会有其他厂商陆续跟进，但是其接口数量需求是与部署位置息息相关的，盲目的扩充接口数并不一定符合数据中心的需要。

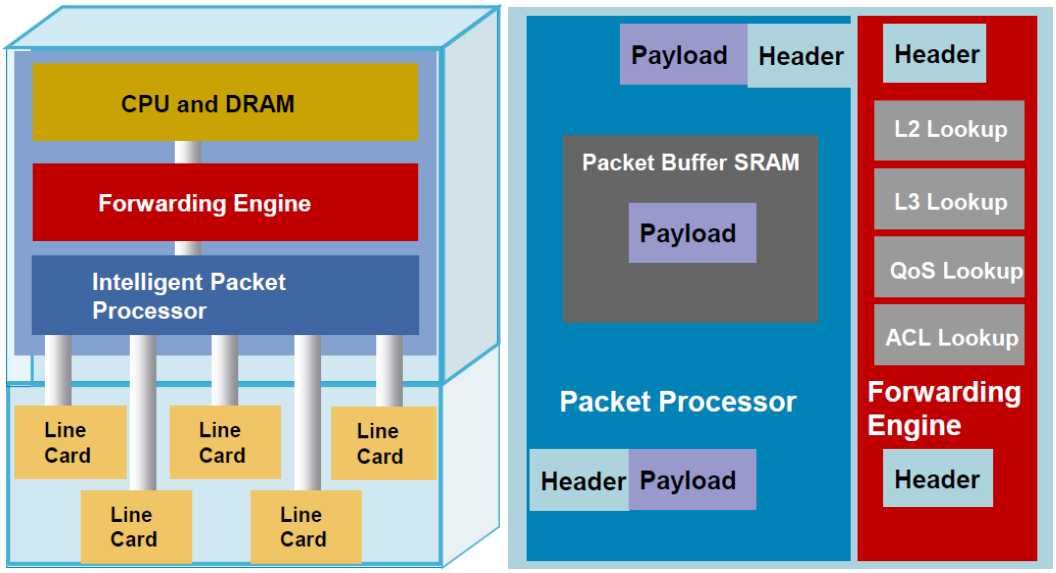
再唠叨几句数据中心Box交换机的选型，前面说了Top Of Rack是Box的主要归宿，一个标准Rack目前最高的42U，考虑冗余怎么也得搞2台Box，剩下最多装40台1U的Server，那么上48GE+4*10GE的Box就是最适合的。依此类推，接口数量多的box不见得真有太大作用，位置会很尴尬。考虑选择Box的最大转发容量时，直接根据服务器接口数来计算接口即可。目前随着FCoE的推进，服务器提供10GE CNA接口上行到接入交换机越来越常见，那么对Box的要求也随之提升到10GE接入40G/100G上行的趋势，像Juniper的QFX3500（48*10GE+4*40GE）明显就是上下行带宽1：3收敛的交换机，估计下一代Top Of Rack的数据中心交换机怎么也得要40*10GE+4*100GE的接口才能彻底搞定42U机架，如果全部署2U的服务器，则最少也需要16*10GE+4*40GE接口的Box才靠谱一些。

4.5 Chassis 与分布式转发

本章节涉及转发能力的举例计算量较大，对数字不感兴趣的同学可以直接略过相关内容。

盒子说完了讲讲框，盒式设备发展到一定程度，接口密度就成了天花板，必须要搞成机框式才能继续扩展了。可以把机框里面的板卡理解为一个个独立的盒子，然后通过交换网络将其连接起来形成整体。

罗马不是一天建成的，机框式交换机最初也是按照集中式转发架构来进行设计。例如Cisco4500系列（又是Cisco，没办法，就他家产品最全，开放出来的资料最多，而且确实是数通领域的无冕之王，下文很多技术也都跟其相关），其接口板（LineCard）上面都没有转发芯片的（XGStub ASIC只做接口缓存和报文排队的动作），所有的数据报文都需要通过背板通道（Fabric），上送到主控板（Supervisor）的转发芯片（Forwarding Engine）上进行处理。结构如下图所示，其中PP（Packet Processor）是做封包解包的，FE（Forwarding Engine）是做查表处理的。

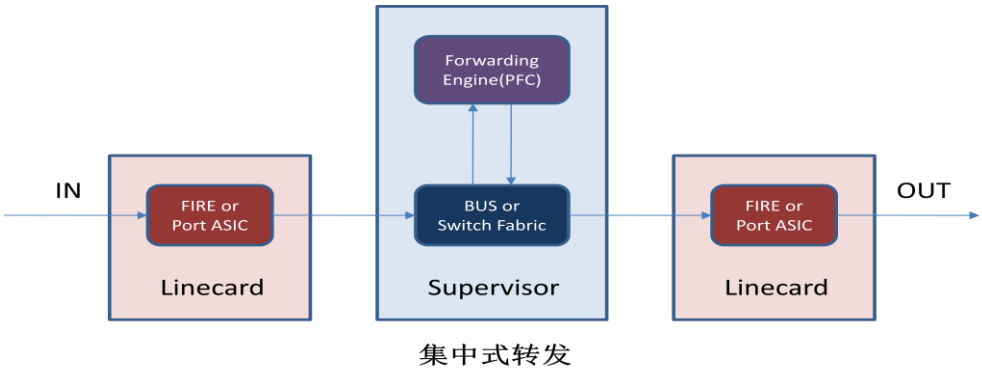


在早期Cisco6500系列交换机设备上同样是基于总线（BUS）的集中式转发结构。如Classic类型接口板（Module）就只有Port ASIC做缓存和排队，所有的报文同样要走到主控板（Supervisor32或720）上的转发芯片（PFC3）来处理。普通的CEF256和CEF720系列接口板虽然以Switch Fabric替代BUS总线通道来处理接口板到主控板的流量转发，但仍然是靠主控板上的PFC3对流量进行集中处理，因此还是集中式转发。直到CEF256和CEF720的DFC（Distributed Forwarding Card）扣板出来，才能在板卡上进行转发，称得上是真正的分布式架构。而最新的第四代接口板dCEF720 Linecards已经直接将DFC变成了一个非可选组件直接集成在接口板上。

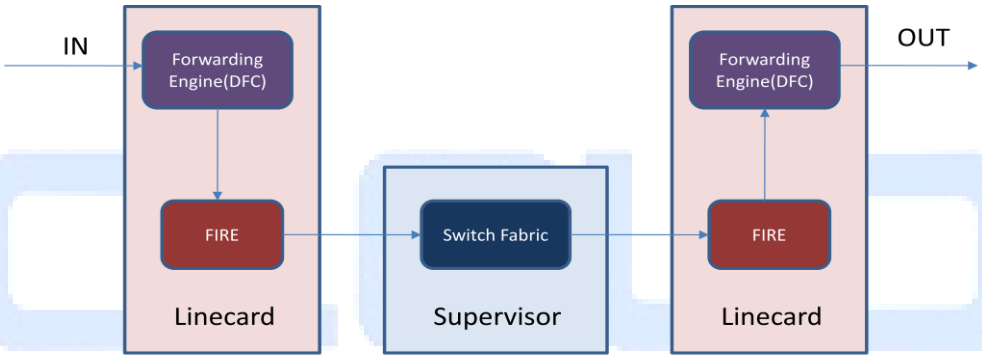
分布式架构指所有的接口板都有自己的转发芯片，并能独立完成查表转发和对报文的L2/L3等处理动作，接口板间通过交换芯片进行报文传递，机框的主控板只通过CPU提供协议计算等整机控制平面功能。分布式架构接口板上都会专门增加一个Fabric连接芯片（Fabric Interface或Fabric Adapter Process等），用以处理报文在框内接口板间转发时的内部报头封装解封装动作。当报文从入接口板向交换芯片转发时，连接芯片为报文封装一个内部交换报头，主要内容字段就是目的出接口板的Slot ID和出接口Port ID，交换芯片收到报文后根据Slot ID查找接口转发，出接口板的连接芯片收到后根据Slot ID确认，并将此内部交换报头去掉，根据Port ID将报文从对应出接口转出交换机。很显然分布式对比集中式的区别主要是芯片更多，成本更高，转发能力也更高。目前各厂商最新一代的主流数据中心交换机都已经是完全的分布式转发架构（如Cisco的N7000，H3C的12500等）。

下面说下Chassis的转发能力，这个可比盒子要复杂多了，各个厂家多如繁星的机框、

主控和接口板种类足以使用户眼花缭乱。还是以Cisco6500系列交换机举例，一法通万法通，搞明白这个其他的也不过尔尔了。选择Cisco6500还有一个主要原因就是其结构从集中式跨越到分布式，从BUS总线通道跨越到Crossbar转发，堪称传统机框交换机百科全书。



集中式转发



分布式转发

FIRE（Fabric Interface & Replication Engine）为Cisco的接口板连接芯片，除了作为连接Switch Fabric的接口对报文进行内部报头的封包解包动作外，还能提供本地镜像和组播复制功能。图中举例了报文在65机框式交换机中跨接口板转发的主要节点。集中式转发时板内接口间流量转发同样适用此图，而分布式转发时板内转发流量不需要走到Switch Fabric。

另外报文走到出方向接口板时是否经过转发芯片处理各个厂家的设备实现并不一致，最简单的一个方法就是看交换机接口板支持不支持出方向的报文ACL（Access Control List）过滤，就知道其有没有上出口板转发芯片处理了。

从上图可以看出接口板的转发能力都受限于板卡连接BUS或Switch Fabric的接口带宽，而衡量整机转发能力时，集中式转发受限于转发芯片FE的转发能力，分布式转发受限于交换芯片Switch Fabric的转发能力。先说接口板转发能力，大家以前可能经常会听到接口板存在非线速和收敛比的概念，看到这里就很好明白了，例如CEF256类型接口板的Switch Fabric

接口带宽是8G，那最多就支持8个GE口和其他接口板进行流量转发，其WS-X6516-GBIC接口板的面板上有16个GE口，明显就是一块2：1的收敛比的非线速板。再如CEF720类型接口板的Switch Fabric接口是2*20G（单板上有两个FIRE），那48GE口的单板也明显不可能是线速的了。即使是号称第四代的dCEF720接口板，其Switch Fabric接口和CEF720一样都是2*20G接口，那么X6708-10G接口板（提供8*10GE接口）和X6716-10G接口板（提供16*10GE接口）只能是2：1和4：1收敛的非线速板了。背板通道预留不足，Switch Fabric交换能力不够，6500系列的这些架构缺陷促使Cisco狠下心来为数据中心重新搞出一套Nexus7000，而其他交换机厂商也都几乎同时期推出了新架构的机框式交换机，都是被逼的啊，谁让1000M接入这么快就替代了100M接入呢，核心更得开始拼万兆了。

再说说整机转发能力。在集中式转发时，Cisco6500不论使用Supervisor32还是Supervisor720主控，FE转发芯片都是走BUS的，带宽都是16G（双向32G），因此只要用的接口板没有DFC，整机最大也就双向32G了。而其中Supervisor32不支持Switch Fabric，也就支持不了DFC的分布式转发，名称里的32就代表了其双向32G的最大整机转发能力。Supervisor720主控支持18*20的Switch Fabric交换，名称中的720是指整个Switch Fabric的双向交换能力18*20*2=720G。但其中1个通道用于连接FE转发芯片，1个通道暂留未用，只有16个通道留给了接口板，意味着整机实际最大能够支持的双向转发能力是16*20*2=640G。Supervisor720-10GE支持20*20的Switch Fabric，多出来的2个10G通道给了Supervisor上的2个10GE接口，实际提供给接口板的交换通道仍然是16*20G。刚刚说了，目前最新的CEF720系列接口板每块有2*20G的出口，简单做个除法，16/2=8，主控板的交换芯片最多能够承载8块CEF720接口板，熟悉Cisco6500产品的同学这时候就会想到6513机框怎么办呢。6513除去7-8的主控槽位外，一共有11个接口板槽位，1-6槽位背板只提供1个Switch Fabric通道，9-13才能提供2个通道，正好是6+2*5=16个通道满足主控板的Switch Fabric交换能力。而6513E虽在1-6槽位背板提供了2个通道，但实际上1-6槽位也同样只能支持1个Switch Fabric通道，否则Supervisor720的Switch Fabric也搞不定的。如果想6513E的接口板通道全用起来，只能等Cisco6500出下一代引擎了，至少是Supervisor880才能搞定6513E的全线速转发，不过从交换芯片的发展来看，Supervisor960的可能性更大一些，1280就有些拗口了。由上看出即使将CEF720接口板插到6513/6513E的1-6槽，也只能跑20G的流量，这下连24GE接口板都无法线速了。

前面算了好多数，好在都是加减乘除，只要搞明白了，完全可以避免选型时再被设备厂商忽悠。题外话，很多厂商的机框千兆接口板（24或48个光/电口）都可以在其同时代盒式交换机中找到相似的影子，假如看到支持相同接口数量类型的接口板和盒子，相信里面的转发芯片十之八九也用的一样。万兆接口板不做成盒式是因为接口密度太低，价格上不去；而高密万兆的盒子做不成接口板则是因为框式交换机交换芯片和背板通道结构限制导致跨板转发能力上不去。

框式交换机架构从集中式发展到分布式后，整机的转发能力迎来了一次跳跃性发展，从Cisco6500的Supervisor32到Supervisor720就可见一斑。那么下一步路在何方呢，各个厂家都有着不同的看法。看回到前面分布式转发的结构图，可以想到要继续提升转发能力有两个主要方向，一个是将单芯片处理能力提升，交换芯片只处理一次查表转发，工作简单相对更容易提升，而转发芯片要干的事情太多就不是那么好换代的了。而另一条路就是增加芯片的数量，转发芯片由于要排布在接口板上，毕竟地方就那么大，发展有限，现在的工艺来说，一块单板放4个转发芯片基本上已经到极限了，6个的也只看到Arista的7548接口板上有，再多的还没有见过，因此转发芯片的发展还是要看芯片厂商的能力了。而像Cisco6500的Supervisor720一样将交换芯片布在主控板上的话，同样面临空间的限制，上面还得放些CPU/TCAM什么的，最多每块主控上面放2个交换芯片就顶天了，双主控能支撑4个，但是全用做转发的话就做不到冗余了。最新的思路是将交换芯片拿出来单独成板，这样只要新机框设计得足够大，交换芯片的数量就不再是限制。例如Cisco的N7000可以插5块交换网板，而H3C的12500能够插9块交换网板。当然转发能力并不是交换芯片的数量越多就越好，还要看具体其单体转发能力和整机背板通道布局。

以Cisco的N7000举例分析，其交换网板Fabric Modules上的CFA（Crossbar Fabric ASICs）宣称是支持每槽位（Slot）2*23G的通道交换，整机最大支持2*23*5=230G的每槽位单向转发能力。这样能看出来啥呢？

1、N7010上8个板卡槽位，2个主控槽位（主控槽位支持1条23G通路），一共是8*2+2=18条通道，可以看出7010的交换网板上就一块Crossbar Fabric ASIC，这个交换芯片和以前Cisco6500 Supervisor720上的18*20交换芯片除了每通道带宽从20G提升到了23G以外，通路数都是18条没有变化，应该属于同一代交换芯片产品。7018可以算出是16*2+2=34条通道，

那么其每块交换网板上应该是2个与7010相同的CFA交换芯片，而且还空了2条通道暂时没用上。

2、其接口板上的数据通道同样应该与交换网板通道相匹配，升级到23G的容量。看下48GE接口板的图，上面只有一块2通道的转发芯片Forwarding Engine，于是了解为啥其只能提供46G的全线速转发，而且使用一块交换网板就可以达到最大转发能力了。

3、再看其10GE接口板，8口万兆板上面有两块2通道的转发芯片，这样80G流量完全够处理，那么算算需要2块交换网板才能线速跨板转发，1块就只能转40G了。而32口万兆板上面就一块4通道的转发芯片，只能搞定80G流量转发，是收敛比4:1的非线速板，同样需要两块交换网板才能达到最大的跨板转发能力。

4、由上面3点可以看出，只使用目前Cisco N7000的接口板的话，交换网板2+1冗余就完全足够用的了。Cisco的下一步换代目标肯定是要想办法提升接口板转发芯片的能力了。首先应该搞定两块4通道转发芯片FE的工艺布局（VOQ和Replication Engine芯片的数量都要翻倍），这样能把16口线速万兆板先搞出来，然后是否研究20*10GE接口板就看其市场战略了。再下一步由于目前交换网板支持每接口板230G的总带宽限制，24/32口万兆线速板肯定是搞不定的。只能先想法将交换网板升级一下，至少得让交换能力再翻一翻才好拿出来搞定32/40口万兆板的线速转发，至于交换芯片是换代还是数量翻番就都有可能了。不过无论走哪条路都不是可以一蹴而就的事情，一两年内应该没戏。

再简单说说H3C的12500，由于其公布资料太少，说多了会有问题。还是从网站公布的宣传值来看。12508背板7.65T，交换容量3.06T/6.12T，包转发率960Mpps/2400Mpps；12518背板16.65T，交换容量6.66T/13.32T，包转发率2160Mpps/5400Mpps。12508与12518都是最大支持9块交换网板，当前主要接口板与N7000相似，含48GE和4万兆、8万兆的线速接口板，32万兆非线速接口板。

1、从背板算起，首先16.65-7.65=9T就是10个槽位的容量，考虑到厂商的宣传值都是双向，那么每接口板槽位应该是预留了9000/2/10=450G的最大出口带宽。根据12508推算，双主控每主控板槽位应该是预留(7650/2-450*8)/2=112.5G的最大出口带宽。由此背板预留通道数接口板与主控板为450:112.5=4:1的关系。基于省钱原则，主控板上肯定只有一条通道，那么接口板都是4条通道，12508背板槽位一共给接口和主控板留了4*8+2=34条通道。

2、12508交换网板总的交换容量3.06T，则每条通道的带宽应该是3060/34=90G，由此可

以推算出实际每块接口板的出口带宽为90*4=360G，同样由于3.06T肯定是个双向值，则每接口板最大交换即可偶带宽理论值为180G（比较Cisco N7000的230G理论值要低一些）。“交换容量3.06T/6.12T”的写法应该指新一代的交换网板芯片能力翻倍或者是数量翻倍，那时其接口板理论带宽就可以达到360G了，还是小于前面计算的背板预留450G的最大带宽，说明背板设计还是考虑不错的。

3、再来算算接口板，从8万兆接口板支持线速转发看来，首先4个通道应该对应到4块转发芯片，每转发芯片对应2个万兆接口，处理20G的流量。而32*10G非线速板应该是同样使用4块转发芯片，所以也是4:1的收敛比。而其48GE和4*10GE的接口板应该是只用了2块同样的转发芯片，转发芯片的接口应该是使用类似于前面盒式交换机中的12.5G带宽线路类型，每块转发芯片对应2组12GE接口或2个10GE接口。考虑其所有接口板采用完全相同转发芯片是因为大量采购时存在价格优势，不像Cisco自己做芯片。

4、返回来再说下12518，总的通道数应该是18*4+2=74，则总的交换容量应该为90*74=6.66T与其宣传值相同。有个小问题，这里的通道数计算是按照接口板与主控板来统计的，以交换板的角度来看时，12508每块交换板一个交换芯片要连8块接口板，每接口板最大4条通道，既需要8*4=32个出口（主控板通道不见得会连接到交换芯片上，也可能是连接到交换网板的CPU）；而12518每块交换板肯定是两个交换芯片，每芯片需要18*4/2=36个出口。这说明12500系列交换机网板上的交换芯片要不就都是32出口的，那么12518有2个槽位只有一半的转发能力；要不就都是36+出口的，12508存在部分出口空余用不上。

5、最后说下包转发率的计算，机框式交换机的包转发率应该是所有转发芯片转发能力的总和。如每个转发芯片20G处理带宽（单向），则转发率应该为20/8/(64+20) =29.76Mpps，取整为30Mpps。按每接口板最大4个转发芯片计算，则12508整机为30*4*8=960M，12518为30*4*18=2160M，符合其宣传值。至于其后面的2400M和5400M两个值，反向推算，每接口板转发能力为2400/8=5400/18=300Mpps，带宽则为300*8*(64+20)=201600约200G，难道是预示着其下一代接口板能够使用2个100G的转发芯片支持2个100G接口，拭目以待。

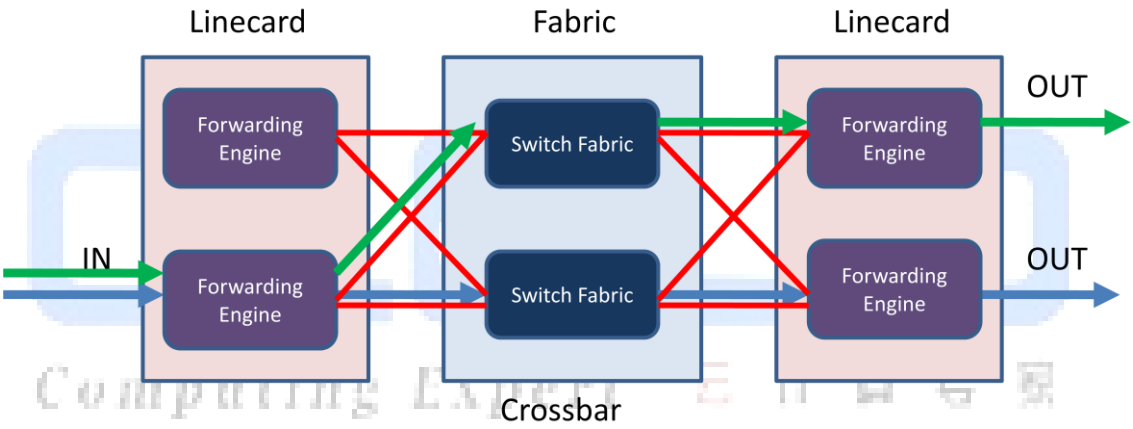
前面算了这么多，希望不会导致头晕吧。

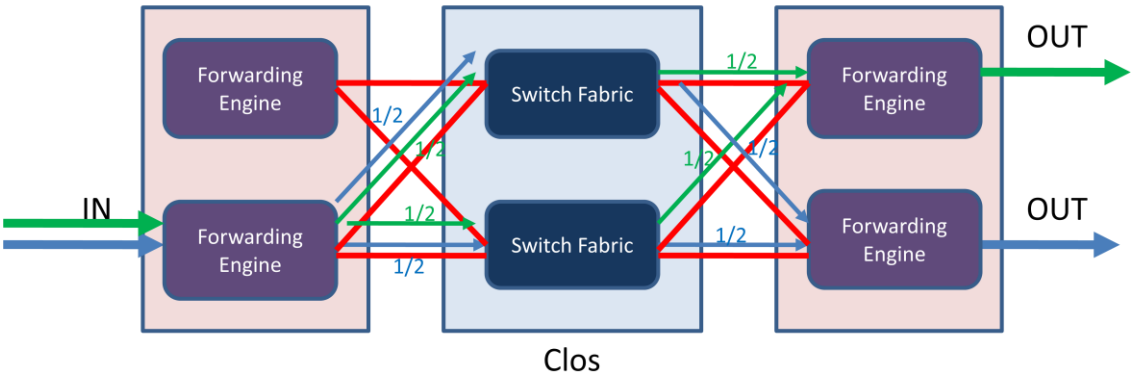
4.6 Clos 与 VOQ

在Crossbar里面，任何两个转发芯片之间的只会经过一块交换芯片，路径是根据背板固定死的。这就导致了两个结构性问题的产生：一是多交换芯片时同一对转发芯片之间的流量

不能被负载分担，如Cisco N7000就是如此；二是当多块入方向接口板往一块出方向接口板打流量的时候，流量可能都走到一块交换芯片上，导致本来应该在出接口板发生的拥塞，提前发生到交换芯片上，产生结构性拥塞，影响其他经过此芯片转发的流量。而且交换芯片采用Cut-Through方式转发是没有缓存的，报文都会直接丢弃，对突发流量的处理不理想。

为解决这两个问题，H3C的12500、Force10的E系列和Foundry BigIron RX等设备都引入了Clos架构的概念（Cisco的CRS系列高端路由器也是Clos结构，但Nexus7000不是）。Clos架构是1953年贝尔实验室研究员Charles Clos设计的一种多级交换结构，最早应用在电话网络中。主要是两个特点，一是可以多级交换，二是每个交换单元都连接到下一级的所有交换单元上。上述厂商设备中基本都是入接口板-交换网板-出接口板的3级交换结构，而根据Clos设计，后续交换网可以扩展成多层结构。Crossbar与Clos的主要区别如下图所示。



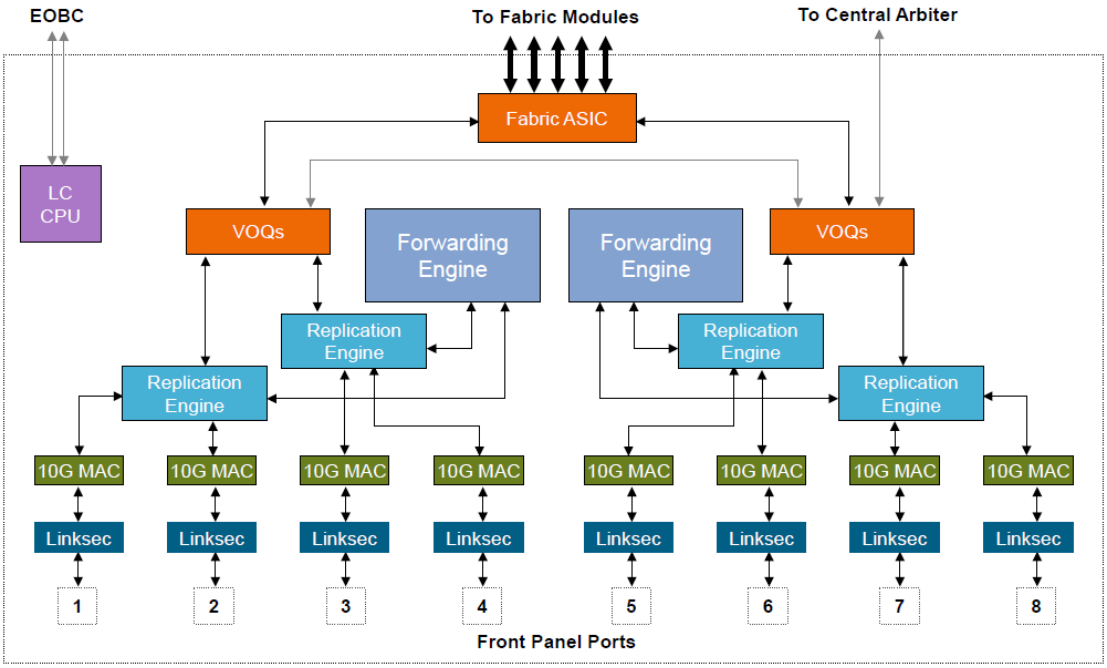


全连接的方式满足了对中间交换芯片的负载均衡需求，同时可以避免单交换芯片的架构性阻塞。不过话说回来，目前机框式交换机的转发能力提升瓶颈还是在转发芯片上，像前面举例的N7000和125都是结构转发能力远远大于实际接口板处理能力。所以暂时还不好说Clos就一定是趋势或代表啥下一代结构，就好像我现在一顿能吃2碗米饭，你给10碗还是20

碗对我没有啥区别，都得等我胃口先练起来再说，但当我胃口真练起来那天，说不定又改吃馒头了呢。

多说一句，H3C的12500在交换芯片转发流量时，报文是在入接口板先被切成等长信元再交给交换芯片的，到出接口板再组合，有些类似ATM转发，号称效率更高。而Force10的E系列则是按报文逐包转发，号称是为了避免乱序等问题。又是各有道理，管他呢，不出问题就什么都好。

目前新的分布式转发交换机另一项重要的技术就是VOQ（Virtual Output Queues）。刚才说的Crossbar第二个拥塞问题在Clos架构中，虽然流量不会在Switch Fabric拥塞，但是多打一的情况下仍然会在出接口板拥塞。VOQ就是在入接口板将报文发给Switch Fabric之前，先用VOQ缓存一下，然后通过中央裁决线路，发一个问询给出接口板，看看那边还有没有空间接收，有的话就发，没有先缓存一会儿，和FC网络中实现零丢包的Bufer to Bufer Credit机制很相似（BB Credit机制详见下文FCoE技术部分）。这样就使出接口板的缓存能力扩充到多块入接口板上，容量翻倍提升，可以有效的缓解突发拥塞导致的丢包问题。看下图Cisco N7000的8口万兆板结构图可以较好理解VOQ在接口板中的位置。



4.7 网络小结

数据中心网络看交换，交换机发展看芯片，分布式转发是必然，Clos架构有得盼。

本章内容是下文数据中心内部服务器通信网络发展技术的重要铺垫。充分了解机框式交换机，可以对后面提出的新一代数据中心网络虚拟化技术，（如Cisco的VN-Tag、Fabric Extend和Fabric Path/E-TRILL等）在理解时起到巨大的帮助。

题外话，目前很多企业规模大了以后，网络部门负责网络，业务部门负责应用和服务器，很多时候互不搭界，于是设计网络和应用的时候就各搞各的，等数据中心建起来之后发现这也是问题那也是问题，各个都变身救火队员，不是啥好现象。有一本书建议所有的网络规划设计人员翻看，《自顶向下的网络设计》，即使找不到或没时间看也请一定要记住这个书名，终身受益的。对应用业务设计人员，也请稍微了解下网络，最少也得能估算出业务上线后理论上的平均带宽和峰值带宽，好向网络设计人员提出需求，免得出事时焦头烂额互相推诿。

5 技术

终于到本文的根本了，前面balabala的说了那么多，都是本章的铺垫，就是希望大家明白下面这些技术是为何而来，要解决什么样的需求和问题。再次对前面的需求进行个汇总。

1、VM之间的互通

2、更多的接口，更多的带宽

3、二层网络规模扩大

4、数据中心站点间二层互联

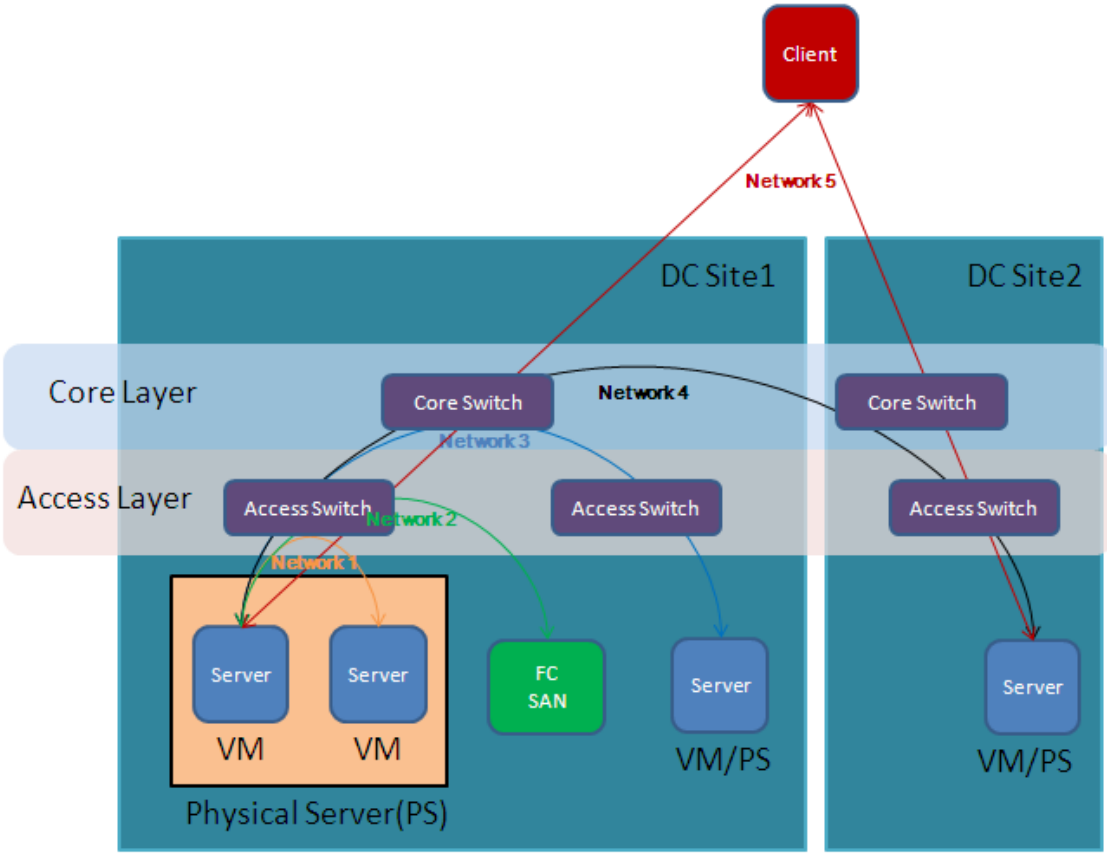
5、VM跨站点迁移与多站点选路

6、服务器前后端网络融合（这个属于厂家引导还是用户需求真不好说）

下面就来看看下面这些网络技术是如何解决上述需求问题的。

5.1 技术结构

前面说了，数据中心网络流量的根本出发点是Server，结合云计算最适合的核心-接入二层网络结构，可以把下面要介绍的各种技术分类如下图所示。此处只做结构上的介绍，具体技术细节将在下文展开。



Network1-VM本地互访网络，边界是Access Switch，包括物理服务器本机VM互访和跨Access Switch的不同物理服务器VM互访两个层面。原有技术以服务器内部安装软件虚拟交换机VSwitch为主，新技术则分为以服务器为主体的802.1Qbg EVB（VEPA/Multi-channel）和Cisco以网络交换机为主体的802.1Qbh BPE（Port Extend/VN-Tag/VN-Link）两大IEEE标准体系。

Network2-Ethernet与FC融合，就是FCoE，边界仍然是Access Switch。在服务器物理网卡到Access Switch这段，将FC数据承载在Ethernet的某个VLAN中传输。但实际上各个厂商当前实现都是做NPV交换机，并不是真正的FCoE，只有很少的产品如Cisco的Nexus5000系列和Brocade的8000系列等能够支持做FCF。

Network3-跨核心层服务器互访网络，边界是Access Switch与Core Switch。可理解为服务器互访流量从进入Access Switch，经过Core Switch，再从另一个Access Switch转出过程的网络处理技术。原有技术就是STP了，新技术分为设备控制平面虚拟化（VSS/vPC/IRF）和整网数据平面虚拟化（SPB/TRILL/Fabric Path）两大体系。这两个体系都是网络虚拟化中的多虚一方向，在一虚多方向除去传统的VLAN/VRF外，Cisco的N7000系列还依照X86架构虚

拟化整出了个VDC。

Network4-数据中心跨站点二层网络，边界是Core Switch。目标是跨越核心网为多个数据中心站点的Core Switch之间建立一条二层通道。根据站点间互联核心网的区别，分为以下三类技术：

- 光纤直连（SDH/DWDM等）对应Ethernet（RPR）
- MPLS核心网对以L2VPN（VLL/VPLS）
- IP核心网对应IP隧道技术（VLLoGRE/VPLSoGRE/L2TPv3/OTV）

Cisco的OTV虽然主要应用在IP核心网中，但实际前面两种方式下同样可以使用，只要多个数据中心站点的Core Switch设备间能够建立可达的IP路径即可部署。使用VLL/VPLS相关技术时必须要增加专门的PE设备为站点间的Core Switch建立二层隧道，而OTV可以直接部署在Core Switch上。

Network5-数据中心多站点选择，技术边界在数据中心与广域网相连的边缘。在云计算中，VM跨站点迁移后，业务服务器IP地址不变，网络指向需要随之变化。这块前面也提到现有技术就是DNS域名解析与ServerLB的NAT配合，以及主机IP路由发布等方式。新技术则是Cisco提出LISP以IPinIP技术结构绕开DNS，由网络设备单独处理Client在广域网中选择站点的情况。

5.2 网络虚拟化

云计算就是计算虚拟化，而存储虚拟化已经在SAN上实现得很好了，那么数据中心三大件也就剩下网络虚拟化了。那么为什么要搞网络虚拟化呢？还是被计算逼的。云计算多虚一时，所有的服务资源都成为了一个对外的虚拟资源，那么网络不管是从路径提供还是管理维护的角度来说，都得跟着把一堆的机框盒子进行多虚一统一规划。而云计算一虚多的时候，物理服务器都变成了一堆的VM，网络怎么也要想办法搞个一虚多对通路建立和管理更精细化一些不是。

5.2.1 网络多虚一技术

先说网络多虚一技术。最早的网络多虚一技术代表是交换机集群Cluster技术，多以盒式小交换机为主，较为古老，当前数据中心里面已经很少见了。而新的技术则主要分为两个方

向，控制平面虚拟化与数据平面虚拟化。

控制平面虚拟化

顾名思义，控制平面虚拟化是将所有设备的控制平面合而为一，只有一个主体去处理整个虚拟交换机的协议处理，表项同步等工作。从结构上来说，控制平面虚拟化又可以分为纵向与横向虚拟化两种方向。

纵向虚拟化指不同层次设备之间通过虚拟化合多为一，代表技术就是Cisco的Fabric Extender，相当于将下游交换机设备作为上游设备的接口扩展而存在，虚拟化后的交换机控制平面和转发平面都在上游设备上，下游设备只有一些简单的同步处理特性，报文转发也都需要上送到上游设备进行。可以理解为集中式转发的虚拟交换机

横向虚拟化多是将同一层次上的同类型交换机设备虚拟合一， Cisco的VSS/vPC和H3C的IRF都是比较成熟的技术代表，控制平面工作如纵向一般，都由一个主体去完成，但转发平面上所有的机框和盒子都可以对流量进行本地转发和处理，是典型分布式转发结构的虚拟交换机。Juniper的QFabric也属于此列，区别是单独弄了个Director盒子只作为控制平面存在，而所有的Node QFX3500交换机同样都有自己的转发平面可以处理报文进行本地转发。

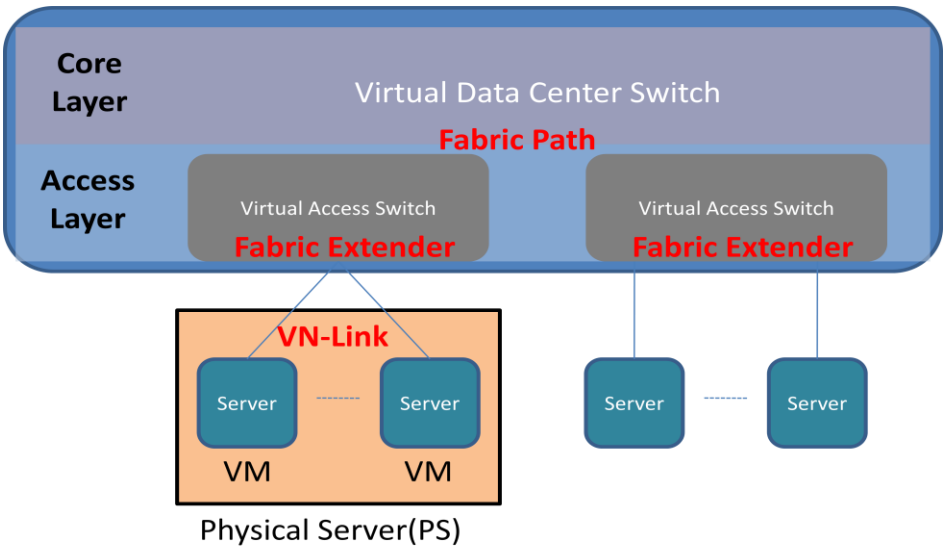
控制平面虚拟化从一定意义上来说是真正的虚拟交换机，能够同时解决统一管理与接口扩展的需求。但是有一个很严重的问题制约了其技术的发展。在前面的云计算多虚一的时候也提到过，服务器多虚一技术目前无法做到所有资源的灵活虚拟调配，而只能基于主机级别，当多机运行时，协调者的角色（等同于框式交换机的主控板控制平面）对同一应用来说，只能主备，无法做到负载均衡。网络设备虚拟化也同样如此，以框式设备举例，不管以后能够支持多少台设备虚拟合一，只要不能解决上述问题，从控制平面处理整个虚拟交换机运行的物理控制节点主控板都只能有一块为主，其他都是备份角色（类似于服务器多虚一中的HA Cluster结构）。总而言之，虚拟交换机支持的物理节点规模永远会受限于此控制节点的处理能力。这也是Cisco在6500系列交换机的VSS技术在更新换代到Nexus7000后被砍掉，只基于链路聚合做了个vPC的主要原因。三层IP网络多路径已经有等价路由可以用了，二层Ethernet网络的多路径技术在TRILL/SPB实用之前只有一个链路聚合，所以只做个vPC就足矣了。另外从Cisco的FEX技术只应用于数据中心接入层的产品设计，也能看出其对这种控制平面虚拟化后带来的规模限制以及技术应用位置是非常清晰的。

数据平面虚拟化

前面说了控制平面虚拟化带来的规模限制问题，而且短时间内也没有办法解决，那么就想个法子躲过去。能不能只做数据平面的虚拟化呢，于是有了TRILL和SPB。关于两个协议的具体细节下文会进行展开，这里先简单说一下，他们都是用L2 ISIS作为控制协议在所有设备上进行拓扑路径计算，转发的时候会对原始报文进行外层封装，以不同的目的Tag在TRILL/SPB区域内部进行转发。对外界来说，可以认为TRILL/SPB区域网络就是一个大的虚拟交换机，Ethernet报文从入口进去后，完整的从出口吐出来，内部的转发过程对外是不可见且无意义的。

这种数据平面虚拟化多合一已经是广泛意义上的多虚一了，相信看了下文技术理解一节会对此种技术思路有更深入的了解。此方式在二层Ethernet转发时可以有效的扩展规模范围，作为网络节点的N虚一来说，控制平面虚拟化目前N还在个位到十位数上晃悠，数据平面虚拟化的N已经可以轻松达到百位的范畴。但其缺点也很明显，引入了控制协议报文处理，增加了网络的复杂度，同时由于转发时对数据报文多了外层头的封包解包动作，降低了Ethernet的转发效率。

从数据中心当前发展来看，规模扩充是首位的，带宽增长也是不可动摇的，因此在网络多虚一方面，控制平面多虚一的各种技术除非能够突破控制层多机协调工作的技术枷锁，否则只有在中小型数据中心里面刨食的份儿了，后期真正的大型云计算数据中心势必是属于TRILL/SPB此类数据平面多虚一技术的天地。当然Cisco的FEX这类定位于接入层以下的技术还是可以与部署在接入到核心层的TRILL/SPB相结合，拥有一定的生存空间。估计Cisco的云计算数据中心内部网络技术野望如下图所示：（Fabric Path是Cisco对其TRILL扩展后技术的最新称呼）



5.2.2 网络一虚多技术

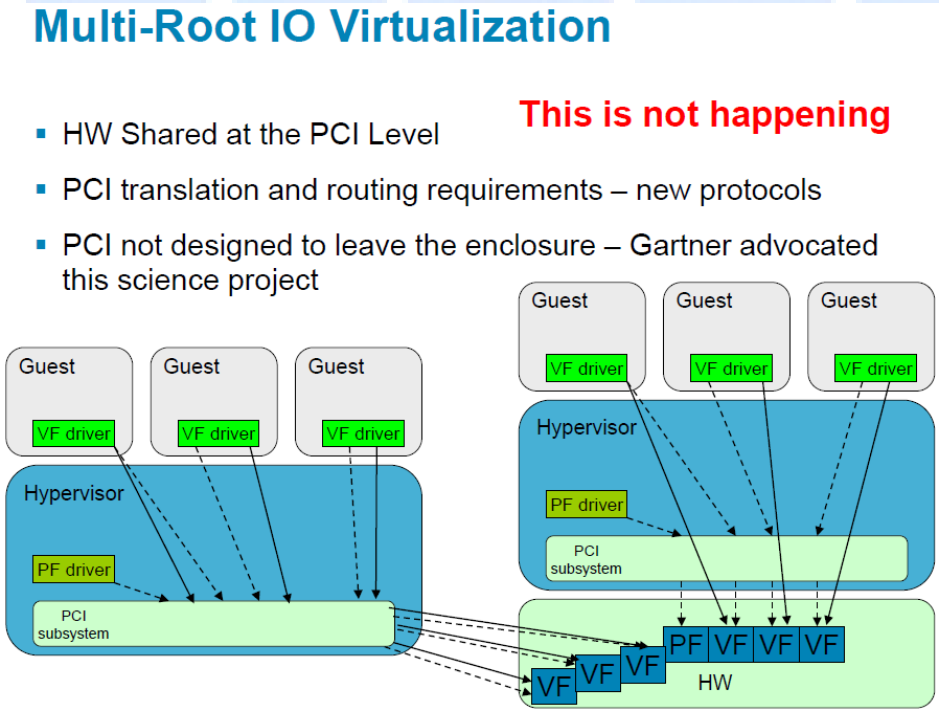
再说网络一虚多，这个可是根源久远，从Ethernet的VLAN到IP的VPN都是大家耳熟能详的成熟技术，FC里面也有对应的VSAN技术。此类技术特点就是给转发报文里面多插入一个Tag，供不同设备统一进行识别，然后对报文进行分类转发。代表如只能手工配置的VLAN ID和可以自协商的MPLS Label。传统技术都是基于转发层面的，虽然在管理上也可以根据VPN进行区分，但是CPU/转发芯片/内存这些基础部件都是只能共享的。目前最新的一虚多技术就是Cisco在X86架构的Nexus7000上实现的VDC，和VM一样可以建立多个VDC并将物理资源独立分配，目前的实现是最多可建立4个VDC，其中还有一个是做管理的，推测有可能是通过前面讲到过的OS-Level虚拟化实现的。

从现有阶段来看，VDC应该是Cisco推出的一项实验性技术，因为目前看不到大规模应用的场景需求。首先转发层面的流量隔离（VLAN/VPN等）已经做得很好了，没有必要搞个VDC专门做业务隔离，况且从当前VDC的实现数量（4个）上也肯定不是打算向这个方向使劲。如果不搞隔离的话，一机多用也没有看出什么实用性，虚拟成多个数据中心核心设备后，一个物理节点故障导致多个逻辑节点歇菜，整体网络可靠性明显降低。另外服务器建VM是为了把物理服务器空余的计算能力都用上，而在云计算数据中心里面网络设备的接口数应该始终是供不应求的，哪里有多少富裕的还给你搞什么虚拟化呢。作者个人对类似VDC技术在云计算数据中心里面的发展前景是存疑的。

SR-IOV

对网络一虚多这里还有个东西要补充一下，就是服务器网卡的IO虚拟化技术。单根虚拟化SR-IOV是由PCI SIG Work Group提出的标准，Intel已经在多款网卡上提供了对此技术的支持，Cisco也推出了支持IO虚拟化的网卡硬件Palo。Palo网卡同时能够封装VN-Tag（VN的意思都是Virtual Network），用于支撑其FEX+VN-Link技术体系。现阶段Cisco还是以UCS系列刀片服务器集成网卡为主，后续计划向盒式服务器网卡推进，但估计会受到传统服务器和网卡厂商们的联手狙击。

SR-IOV就是要在物理网卡上建立多个虚拟IO通道，并使其能够直接一一对应到多个VM的虚拟网卡上，用以提高虚拟服务器的转发效率。具体说是对进入服务器的报文，通过网卡的硬件查表取代服务器中间Hypervisor层的VSwitch软件查表进行转发。另外SR-IOV物理网卡理论上加块转发芯片，应该可以支持VM本地交换（其实就是个小交换机啦），但个人目前还没有看到实际产品。SR（Single Root）里面的Root是指服务器中间的Hypervisor，单根就是说目前一块硬件网卡只能支持一个Hypervisor。有单根就有多根，多根指可以支持多个Hypervisor，但貌似目前单物理服务器里面跑多个Hypervisor还很遥远，所以多根IO虚拟化MR-IOV也是个未来未来时。摘录Cisco胶片对MR-IOV描述如下：（HW为Hardware，PF为Physical Function，VF为Virtual Functions）



SR-IOV只定义了物理网卡到VM之间的联系，而对外层网络设备来说，如果想识别具体的VM上面的虚拟网卡vNIC，则还要定义一个Tag在物理网卡到接入层交换机之间区分不同

vNIC。此时物理网卡提供的就是一个通道作用，可以帮助交换机将虚拟网络接口延伸至服务器内部对应到每个vNIC。Cisco UCS服务器中的VIC（Virtual Interface Card）M81-KR网卡（Palo），就是通过封装VN-Tag使接入交换机（UCS6100）识别vNIC的对应虚拟网络接口。

网络虚拟化技术在下一个十年中必定会成为网络技术发展的重中之重，谁能占领制高点谁就能引领数据中心网络的前进。从现在能看到的技术信息分析，Cisco在下个十年中的地位仍然不可动摇。

5.3 技术理解

进入正式介绍之前再多说两句如何快速理解技术的思路。搞网络的一般最头疼最不情愿的就是去读RFC等标准技术文档了，至少心底里有抵触。各种各样的报文、状态机、数据库、链表充斥于字里行间，再加上标准文档为了避免歧义，一句话能说清楚的也得分成三四句解释来解释去。也许是眼界不够开阔，反正我还真不认识能把草案标准当成小说看的牛人。

这里只是简单介绍一下协议入门的经验，如果想要深入甚至精通，那还得去一字一字的考究了，做学问来不得半点马虎。

学习一门技术前，首先要了解的是由何而来与从何而去，既技术产生的背景和应用的地方，这样对其要解决什么样的问题大致能有个印象。举例来说PBB是运营商城域以太网技术，运营商技术的特点就是组网规模大，节点众多，路径众多。而传统以太网只能使用STP避免环路，阻塞了一堆链路，这个在运营商里面也是不可想象的，那一条条链路都是钱啊。因此PBB肯定是要在避免环路的同时，能够增大以太网组网规模和将所有路径都利用起来的技术。

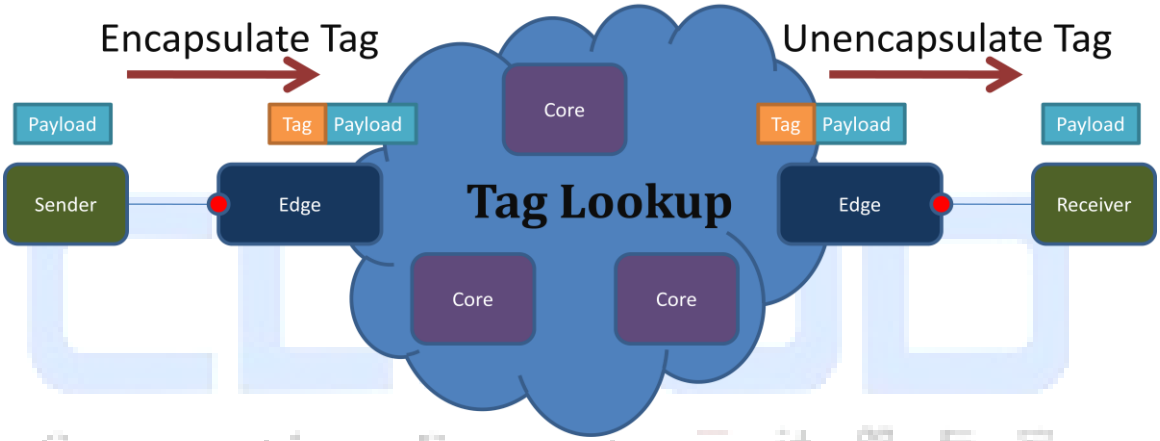
再来就是看技术的类型， Routed Protocol和Routing Protocol两个词很好的对技术组成部分进行了分类。这里的Route可以进行广义理解，不要只限于IP，作者倾向于将Routed解释成封装，Routing解释成寻址。任何一段数据信息从起点A发送到终点B的过程中，中间网络做的事情就是封装与寻址两件事。

由于中间网络只做传输，是不需要了解数据信息的，因此要封装一个可以识别的目的地址Tag，这个Tag可以理解为目的IP/目的MAC/MPLS标签等等，所有中间设备只要能识别这个Tag即可，这就是封装。

再说寻址，网络设备能够识别目的Tag后，还需要知道对应的本地出接口在哪才能将报文转发出去。最傻瓜的处理方式有两种，一个是通过手工配置的方式将Tag静态对应到本地出接口上（如静态路由、静态MAC等），再有就是在所有接口广播了（Ethernet）。高级的

方式则是使用一种寻址用的动态协议，自动的进行邻居发现、拓扑计算和Tag传递等动作，如使用RIP/OSPF/BGP/ISIS/LDP/PIM/MSDP等等。这里需要注意的是传统Ethernet是通过广播来寻址的，注定规模不能太大。STP的唯一作用就是防止环路，通过拓扑计算将多余的路径阻塞掉，与寻址无关。而前面提到的那些寻址协议主要任务都是传递Tag计算转发路径，大部分协议会通过计算拓扑来防止环路，但也有如RIP这种不计算拓扑的协议，搞些水平分割、毒性逆转和最大跳数等机制来避免环路。

封装解封装技术是网络入口与出口节点在原始数据信息前将Tag进行加载剥离动作，寻址技术则是在网络节点之间运行的交互动作。在很多协议技术中提到的数据平面其实就是封装转发，而控制平面就是标识寻址。



图是不是很眼熟，大部分的网络协议够可以照着这个模型去套的。

对于IP来说，Sender和Receiver就是TCP协议栈，Edge就是IP协议栈，Core就是Router，Payload就是TCP数据，Tag就是IP头中的目的IP；

对于Ethernet来说，Sender和Receiver就是IP协议栈，Edge就是网卡接口，Core就是Switch，Payload就是IP数据，Tag即使Ethernet头中的目的MAC；

对于MPLS来说，Sender和Receiver就是CE，Edge就是PE，Core就是P；Payload就是Ethernet/IP数据，Tag就是MPLS标签；

甚至对于分布式结构机框交换机来说，Sender和Receiver就是接口板转发芯片，Edge就是接口板上的交换接口芯片，Core就是交换芯片，Payload就是Ethernet数据报文，Tag就是目的Slot ID和Port ID（交换芯片转发时只看Slot ID，目的接口板查看Port ID）。

传统的FC/IP/Ethernet技术体系上面已经玩不出来花了，现在新的技术大都是在FC/IP/Ethernet等数据载荷外面增加个新的Tag并设计一套对应的寻址协议机制（如MPLS和

下文的FEX/ TRILL等），或者干脆就还使用原有的IP/MAC作为外层封装Tag，只对寻址进行变化。对于后者，作者喜欢称呼其为嫁接技术，神马MACinMAC，IPinIP，MACinIP等等都属于此列。此类技术的好处是兼容，缺点是继承，缝缝补补肯定没有全新设计来得自由。

封装比较好明白，协议理解的难点其实在于寻址。前面说了，静态寻址要手工一条条配置，规模大了能累死人。动态寻址技术配置工作量小了很多，但复杂度就上升了好几个台阶。不劳力就劳心，目前看来大家还是更喜欢劳心一些。回来说动态寻址，除了RIP这种早期的靠广播来传递路由Tag的寻址协议外，后面出来的都是先建邻接，后画拓扑，再传Tag的三步走了，从OSPF/BGP/ISIS到下面要讲到的TRILL/SPB/OTV皆是如此。对寻址技术主要内容简单归纳如下，细的就要看各协议具体实现了，希望有助于读者在学习寻址协议时能够少死些脑细胞。（文学素养有限，合辙押韵就算了吧）

建立邻居靠Hello（Advertise），拆除邻接等超时。各自为根绘周边，主根扩散画整网。Tag同步传更新，本地过期发删除。

技术理解部分就说这些，希望对读者认识新技术时能够有所帮助。下面开始进入技术主题。

5.4 VM 本地互访网络技术

本章节重点技术名词：EVB/VEPA/Multichannel/ SR-IOV/VN-Link/FEX/VN-Tag/ UCS/ 802.1Qbh/802.1Qbg

题目中的本地包含了两个层面，一个是从服务器角度来看的物理服务器本地VM互访，一个是从交换机角度来看的接入层交换机本地VM互访。这两个看问题的角度造成了下文中EVB与BPE两个最新技术体系出发点上的不同。

在VM出现伊始，VMware等虚拟机厂商就提出了VSwitch的概念，通过软件交换机解决同一台物理服务器内部的VM二层网络互访，跨物理服务器的VM二层互访丢给传统的Ethernet接入层交换机去处理。这时有两个大的问题产生了，一是对于VSwitch的管理问题，前面说过大公司网络和服务器一般是两拨人负责的，这个东西是由谁来管理不好界定；二是性能问题，交换机在处理报文时候可以通过转发芯片完成ACL packet-filter、Port Security（802.1X）、Netflow和QoS等功能，如果都在VSwitch上实现，还是由服务器的CPU来处理，太消耗性能了，与使用VM提高服务器CPU使用效率的初衷不符。

Cisco首先提出了Nexus1000V技术结构来解决前面的问题一，也只解决了问题一。为了

解决问题二，IEEE（Institute of Electrical and Electronics Engineers）标准组织提出了802.1Qbg EVB（Edge Virtual Bridging）和802.1Qbh BPE（Bridge Port Extension）两条标准路线了，Cisco由802.1Qbh标准体系结构实现出来的具体技术就是FEX+VN-Link。

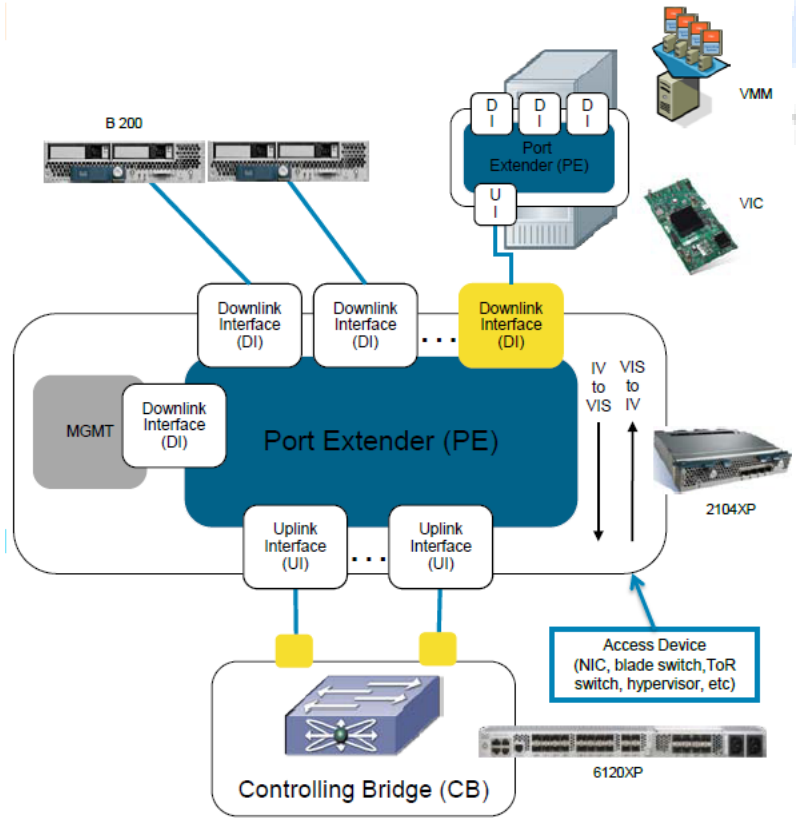
在数据通信世界只有两个阵营：Cisco和非Cisco。而就目前和能预见到的未来而言，非Cisco们都仍是Cisco的跟随者和挑战者，从数据中心新技术发展就可见一斑。在VM本地互访网络技术章节中会先介绍Cisco的相关技术与产品，再讲讲挑战者们的EVB。

5.4.1 Cisco 接入层网络虚拟化

Cisco在其所有的VM接入技术中都有两个主要思路：一是将网络相关内容都虚拟化为一台逻辑的框式交换机来集中由网络进行管理，二是给每个VM提供一个虚拟交换机接口（vETH/VIF）。目的都是以网络为根，将枝叶一步步伸到服务器里面去。

802.1Qbh

先来看下802.1Qbh BPE（Bridge Port Extension），下图是Cisco以UCS系列产品对应的结构图。



802.1Qbh定义的是VM与接入层交换机之间的数据平面转发结构，不包括控制平面。这

里可以将其看为一台虚拟的集中式框式交换机，其中CB可以理解为带转发芯片的主控板，PE就是接口板。PE进入服务器内部是通过硬件网卡来实现的，后续可能在Hypervisor层面会做软件PE来实现。Cisco通过FEX来定义CB到PE以及PE到PE的关系，其数据平面是通过封装私有的VN-Tag头来进行寻址转发；通过VN-Link来定义PE的最终点DI到VM的vNIC之间的关系，提出了Port Profile来定制DI的配置内容。

在802.1Qbh结构中，整个网络是树状连接，每个PE只能上行连接到一个逻辑的PE/CB，因此不存在环路，也就不需要类似于STP这种环路协议。所有的VM之间通信流量都要上送到CB进行查表转发，PE不提供本地交换功能。PE对从DI收到的单播报文只会封装Tag通过UI上送，UI收到来的单播报文根据Tag找到对应的DI发送出去。对组播/广播报文根据Tag里面的组播标志位，CB和PE均可以进行本地复制泛洪。更具体的转发处理流程请参考下文Nexus5000+Nexus2000的技术介绍。

Cisco根据802.1Qbh结构在接入层一共虚拟出三台框式交换机，Nexus1000V（VSM+VEM）、Nexus5000+Nexus2000和UCS。其中1000V还是基于Ethernet传统交换技术的服务器内部软件交换机，没有FEX，主要体现VN-Link；而Nexus5000+Nexus2000则是工作于物理服务器之外的硬件交换机盒子，以FEX为主，VN-Link基本没有；只有到UCS才通过服务器网卡+交换机盒子，完美的将FEX+VN-Link结合在一起。下面来逐台介绍。

Cisco Nexus1000V

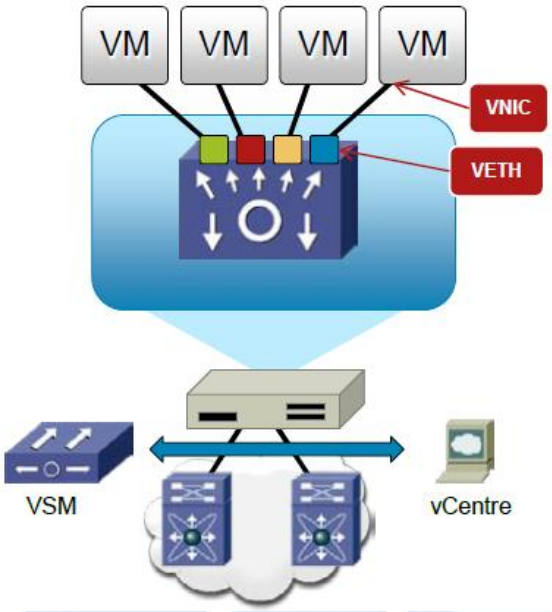
Nexus1000V包含两个组件VSM（Virtual Supervisor Module）与VEM（Virtual Ethernet Module）。看名字就能瞧出VSM对应机框交换机的主控板Supervisor，而VEM对应其接口板。

VEM就是一台安装运行在采用裸金属虚拟化结构的物理服务器中Hypervisor层次的软件交换机，其虚拟接口vETH分为连接VM虚拟网卡vNIC的下行接口和连接到每个物理网卡接口的上行接口，使用Ethernet基于MAC方式进行报文转发。由于其处于网络的末端，不需要运行STP，通过不允许上行接口收到的报文从其他上行接口转发的规则来避免环路的产生。与早期的VSwitch相比多了很多交换机相关功能。

VSM则有两种形态，可以是独立的盒子，也可以是装在某个OS上的应用软件。要求VSM和VEM之间二层或三层可达，二层情况下VSM与VEM之间占用一个VLAN通过组播建立连接，三层情况下通过配置指定IP地址单播建立连接。VSM是一个控制平台，对VEM上的vETH进行配置管理。通过VSM可以直接配置每台VEM的每个vETH。

VSM在管理vETH的时候引入了Port Profile的概念，简单理解就是个配置好的模板，好处是可以一次配置，四处关联。在VM跨物理服务器迁移时，VSM就可以通过vCenter的通知了解到迁移发生，随之将Port Profile下发到VM迁移后对应的vETH上，使网络能够随VM迁移自适应变化。

VN-Link是CISCO在虚拟接入层的关键技术，VN-Link=vNIC+vETH+Port Profile。



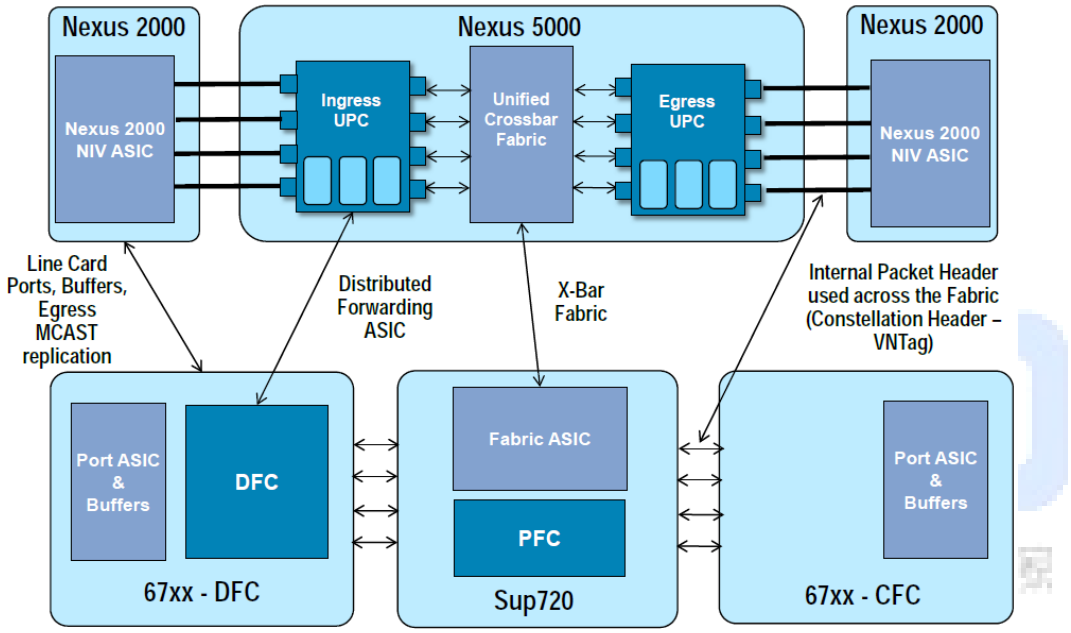
Nexus1000V中的vETH是建立在软件交换机上的，而下文UCS系统里面的vETH就建立在Cisco的网卡硬件上了，对应到UCS虚拟交换机上就是VIF（Virtual Interface），同时UCS通过硬件实现可以把FEX里面要介绍的VN-Tag网络封装标识引入到物理服务器里面。

VEM之间通过普通Ethernet交换机相连，跨VEM转发的流量也是传统以太网报文，因此Nexus1000V虽然可以理解为一台虚拟交换机，但不是集中式或分布式结构，也不存在交换芯片单元，仅仅是配置管理层面的虚拟化，属于对传统VSwitch的功能扩展，只解决了最开始提到的管理边界问题，但对服务器性能仍然存在极大耗费。

从产品与标准的发布时间上看，Nexus1000V是先于802.1Qbh推出的，因此推测Cisco是先做了增强型的VSwitch-Nexus1000V，然后才逐步理清思路去搞802.1Qbh的BPE架构。1000V属于过渡性质的兼容产品，后续应该会对其做些大的改动，如改进成可支持VN-Tag封装的软件PE，帮助N5000+N2000进入物理服务器内部，构造FEX+VN-Link的完整802.1Qbh结构。

Nexus5000+Nexus2000

N5000+N2000组成了一台集中式结构的虚拟交换机，集中式是指所有的流量都要经过N5000交互，N2000不提供本地交换能力，只是作为N5000的接口扩展。对应802.1Qbh结构，N5000就是CB，而N2000就是PE。组合出来的虚拟交换机中，N5000就是带转发芯片和交换芯片的主控板，而N2000则是接口板，整体更像Cisco早期的4500系列机框或使用主控板PFC进行转发的6500系列机框，但是在N5000盒子内部又是以分布式结构处理转发芯片与交换芯片连接布局的，可参考如下的N5000和6500结构比较图。整了半天其实数据平面转发报文还是那几个步骤。

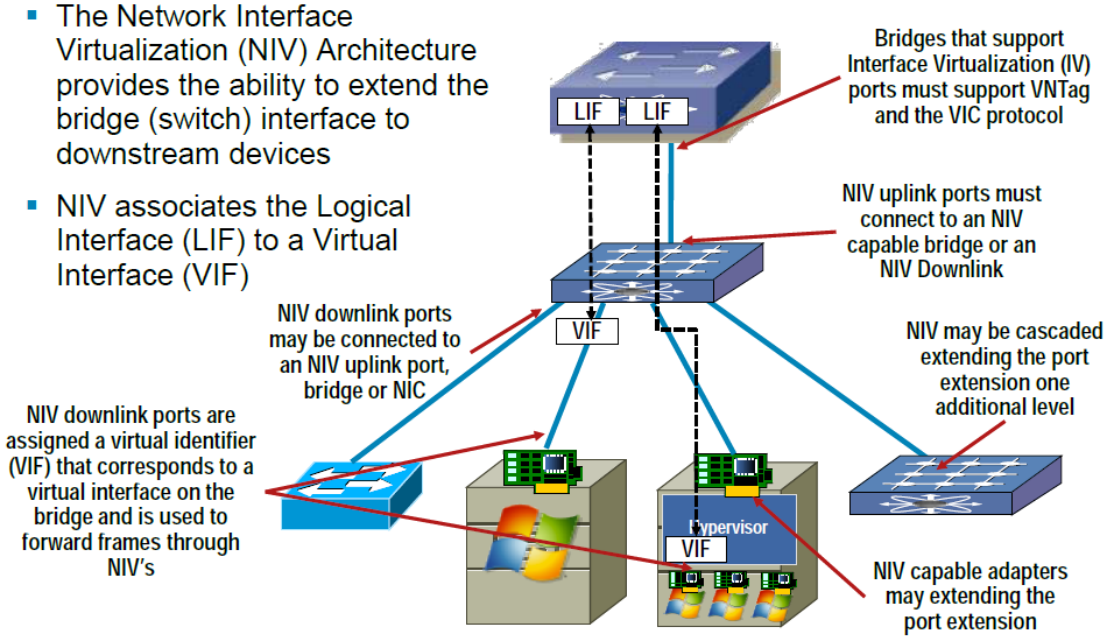


N5000+N2000实现了Cisco的FEX典型结构（Fabric Extend，等同于Port Extend）。在N5000上看到每台N2000就是以一个FEX节点形式出现的接口板。N2000拥有两种物理接口类型，连接下游设备（可以是服务器或N2000，FEX结构支持级联扩展）的HIF（Host Interface）和连接上游N5000和N2000的NIF（Network Interface），此两种接口是固定于面板上的，且角色不可变更。以2248T举例，右侧黄色接口为NIF，其他为HIF。

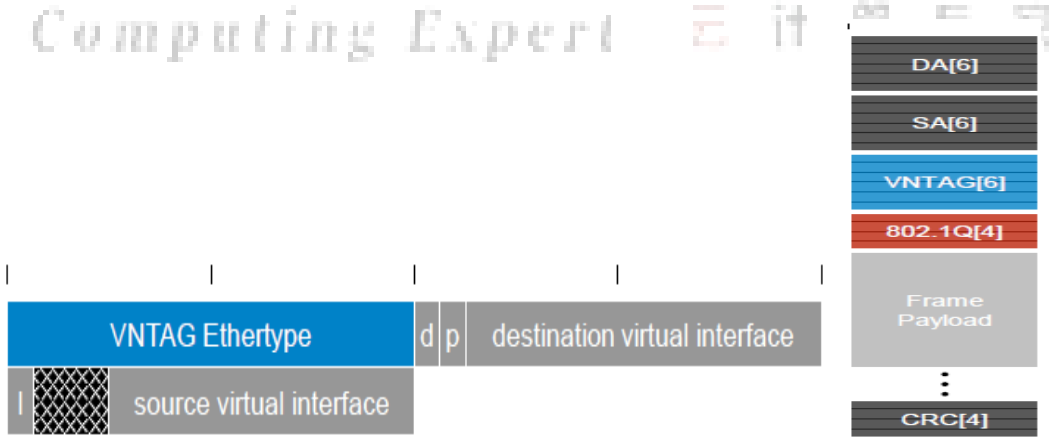
Cisco将FEX结构又称为Network Interface Virtualization Architecture (NIV)，在NIV中将N2000上的HIF称为Virtual Interface (VIF)，将N5000上对应HIF的逻辑接口称为Logical Interface (LIF)。截取Cisco胶片如下描述NIV的内容。

Network Interface Virtualization Architecture (NIV)

- The Network Interface Virtualization (NIV) Architecture provides the ability to extend the bridge (switch) interface to downstream devices
- NIV associates the Logical Interface (LIF) to a Virtual Interface (VIF)



在NIV模型中所有的数据报文进入VIF/LIF时均会被封装VN-Tag传递，在从VIF/LIF离开系统前会剥离VN-Tag。VN-Tag就是在FEX内部寻址转发使用的标识，类似于前面框式交换机内部在转发芯片与交换芯片传输报文时定义槽位信息与接口信息的标识。VN-Tag格式与封装位置如下：



d位标识报文的走向，0代表是由N2000发往N5000的上行流量，1代表由N5000发往N2000的下行流量。

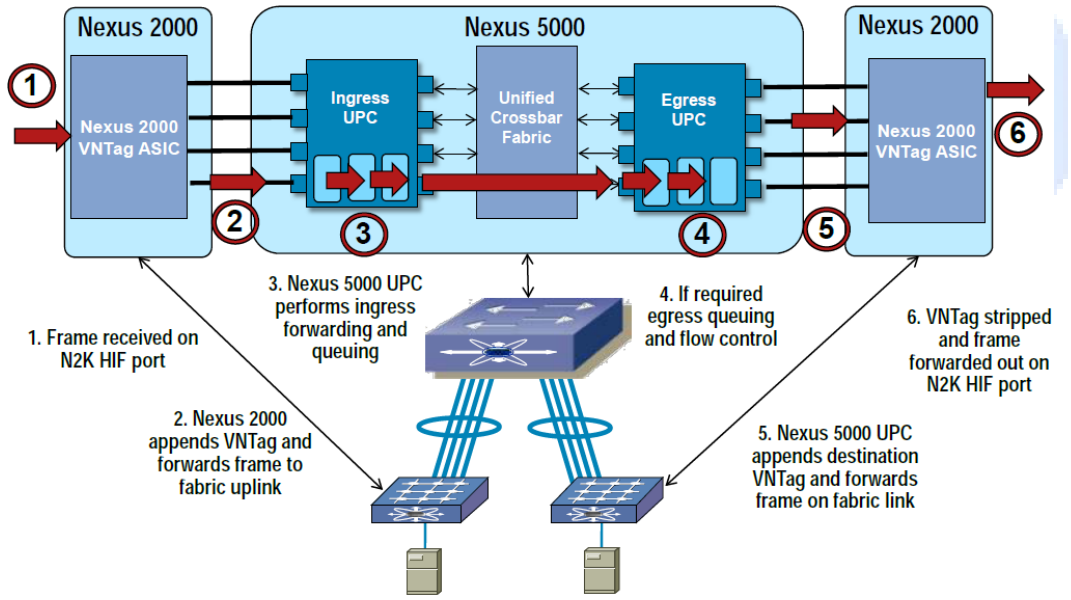
p位标识报文复制，0代表不需要复制，1代表N2000收到此报文后需要向同VLAN的所有本地下行接口复制。此位只有N5000可以置位。

l位标识报文是否返回给源N2000，既0代表源和目的接口不在同一个N2000上，1代表目

的与源接口都在同一个N2000设备上。

DVI（Destination Virtual Interface）标识目的HIF接口，SVI（Source Virtual Interface）标识源HIF接口。每个HIF接口ID在一组FEX系统中都是唯一的。

流量转发时，N2000首先从源HIF收到报文，只需要标识SVI的对应HIF信息，其他位都置0不用管，直接从NIF转发到N5000上即可。N5000收到报文，记录源HIF接口与源MAC信息到转发表中，查MAC转发表，如果目的MAC对应非LIF接口则剥离VN-Tag按正常Ethernet转发处理；如果目的接口为LIF接口，则重新封装VN-Tag。其中DVI对应目的HIF，SVI使用原始SVI信息（如果是从非LIF源接口来的报文则SVI置0），d位置1，如果是组播报文则p位置1，如果目的接口与源接口在一台N2000上则l位置1。N2000收到此报文后根据DVI标识查找本地目的出接口HIF，剥离VN-Tag后进行转发，如果p位置1则本地复制转发给所有相关HIF。每个FEX的组播转发表在5000上建立，所有2000上通过IGMP-Snooping同步。转发过程截取Cisco胶片介绍如下：

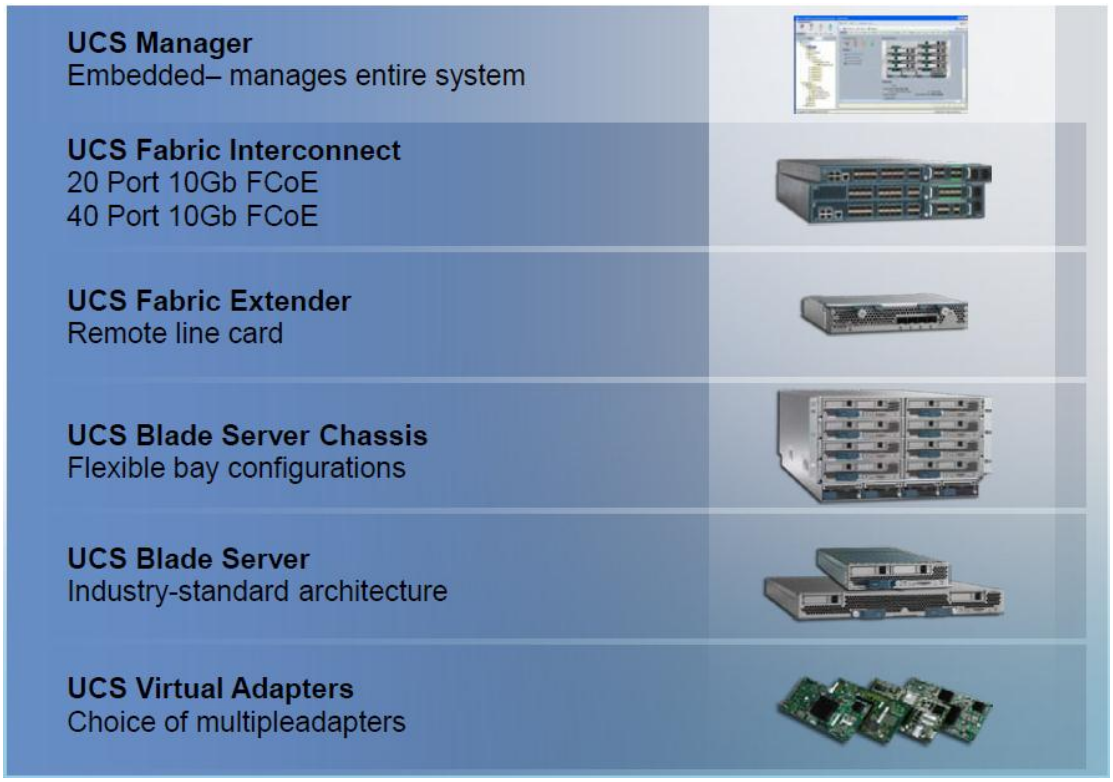


从前面NIV的结构图中可以看到Cisco希望将FEX通过网卡推进到服务器内部，但实际上目前阶段由于Cisco在服务器网卡方面的市场地位，这个还只是一个梦想。N2000还是只能基于物理服务器的物理网卡为基本单元进行报文处理，搞不定VM的vNIC，因此前面说N5000+N2000这台虚拟交换机只实现了FEX，但没有VN-Link。

好吧，搞不定服务器就搞不定网卡，更没有办法推行FEX+VN-Link的802.1Qbh理念。于是Cisco一狠心，先搞了套UCS出来。

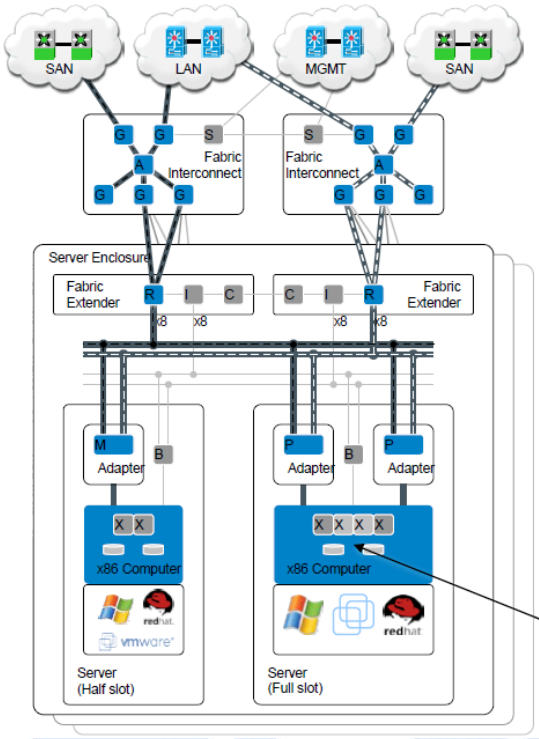
UCS

UCS（Unified Computing System）是包括刀箱、服务器、网卡、接口扩展模块、接入交换机与管理软件集合的系统总称。这里面的各个单元独立存在时虽然也能用，但就没有太大的价值了，与其他同档产品相比没有任何优势，只有和在一起才是Cisco征战天下的利器。UCS产品结构如下图所示：



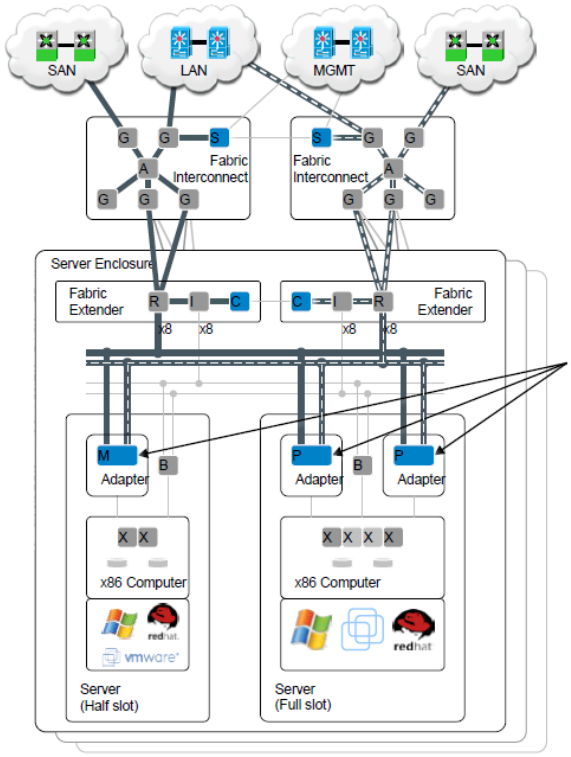
其中服务器、刀箱机框和管理软件都是标准的东西，没啥可多说的。关键部件就是网卡、交换机和刀箱的扩展线卡（Fabric Extender，这个名字个人觉得不好，容易与FEX结构混淆，可以叫个FE Blade或者Fabric Line Card什么的）。Interconnect交换机对应N5000，Fabric Extender对应N2000，这样加上Virtual Adapters就可以实现前面NIV结构中将VIF（HIF）直接连到VM前的期望了，从而也就能够完美实现802.1Qbh BPE（Cisco FEX+VN-Link）的技术体系结构。整个UCS系统结构就在下面这三张图中体现，分别对应数据平面（转发平面）、控制平面、管理平面。由于技术实现上和前面将的FEX和VN-Link没有大的区别，不再做重复赘述，有兴趣的同学可自行细琢磨。

Data Plane

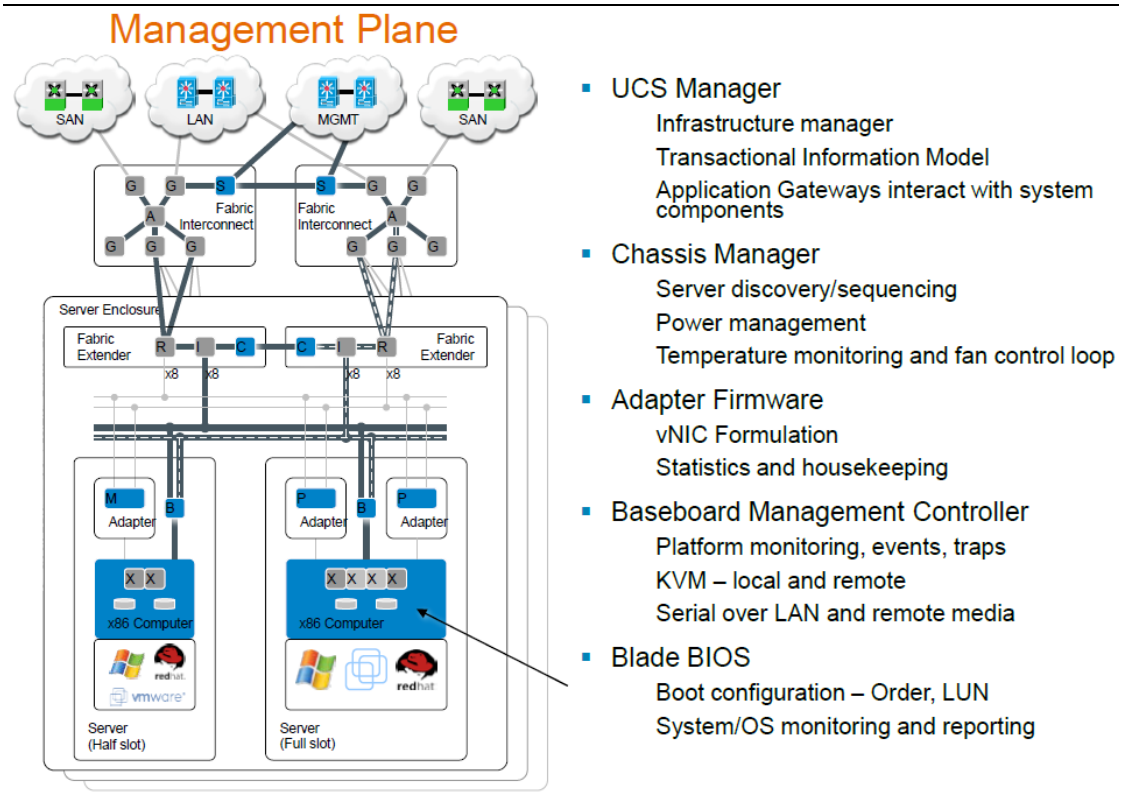


- A) X-bar ASIC
  - 1.12 Tbps Xbar
  - 3 Unicast and 1 Multicast crosspoints
- G) Forwarding ASIC
  - XE/FC/GE Media Access Controllers
  - Forwarding - Ethernet, Fibre Channel, Multipath
  - Policy Engine
  - Packet Buffering
- R) VNTag ASIC
  - Host to uplink traffic engineering
  - Connectivity detection & management portal
- M) DCB ASIC
  - Couple Industry standard NICs/HBAs to ServerArray
- P) Virtualization (SRIOV/VNTAG) ASIC
  - Virtualized adapter for single OS and hypervisor systems
  - Ethernet and Fibre Channel vNICs
    - Direct Data Placement for Fibre Channel
- Memory Controller ASICs
  - Large memory configurations

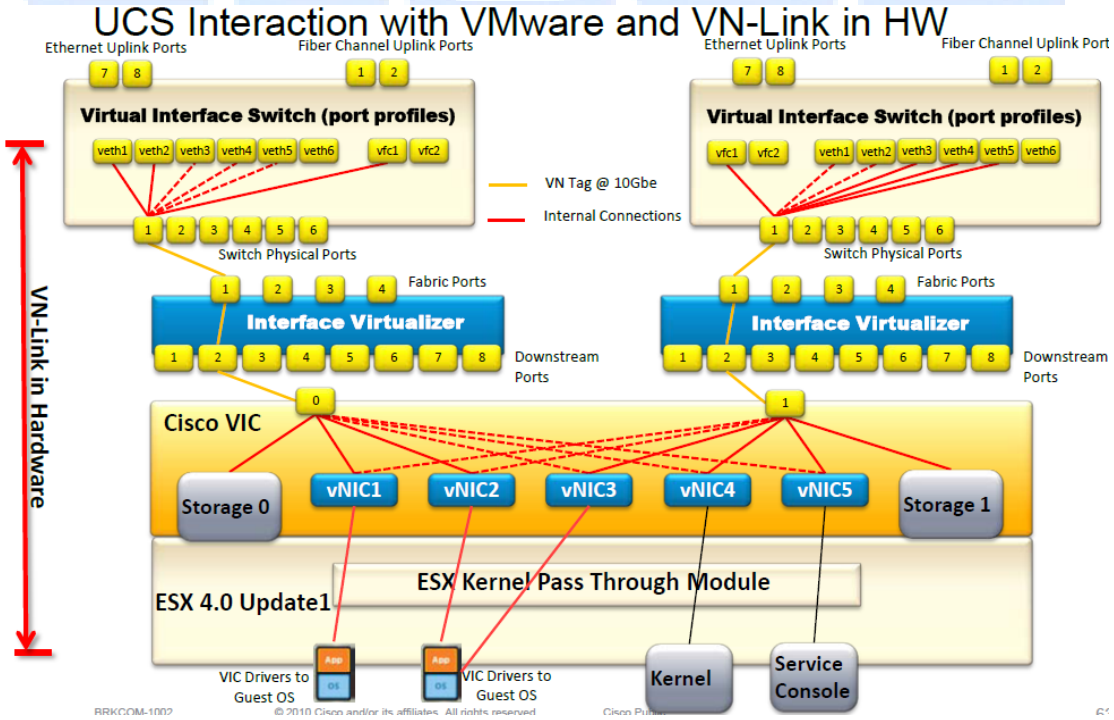
Control Plane



- Interconnect Supervisor
  - *Infrastructure and Ethernet*
  - Consolidated Ethernet/Fibre Channel
  - Network Interface Virtualization
  - Distributed Interconnect Fabric
- Fabric Extender
  - Fabric Connectivity
  - Satellite Interconnect ports and vNIC channels
- Adapter Firmware
  - Network controlled
    - Inaccessible from the host
  - vNIC instantiation
  - Fabric based balancing and failover
  - Fibre Channel/SCSI control suite (M81KR)



再补充一张UCS VN-Link的示意图，可以对应看一下三要素vNIC、vEth和Port Profile的位置连接关系。



UCS就说这么多了，前面介绍的Cisco三台虚拟交换机如果铺开了讲每个都有上百页的内容，本文只是希望能够从结构上帮助大家理解，将作者的知识大厦框架拿出来与读者共享

参考，至于每个人的楼要怎么盖还需自己去添砖加瓦。包括下文的技术点讲解也是如此，作者会将自己认为最重要的关键部分讲出来，细节不会过于展开。

5.4.2 802.1Qbg EVB

说完了Cisco再说说非Cisco阵营，在如802.1Qbg EVB和802.1aq SPB等所谓挑战技术的参与编纂者中，都会看到Cisco的身影。如下图为2009年IEEE Atlanta, GA时发出的EVB所有撰稿相关人名单。

IEEE 802

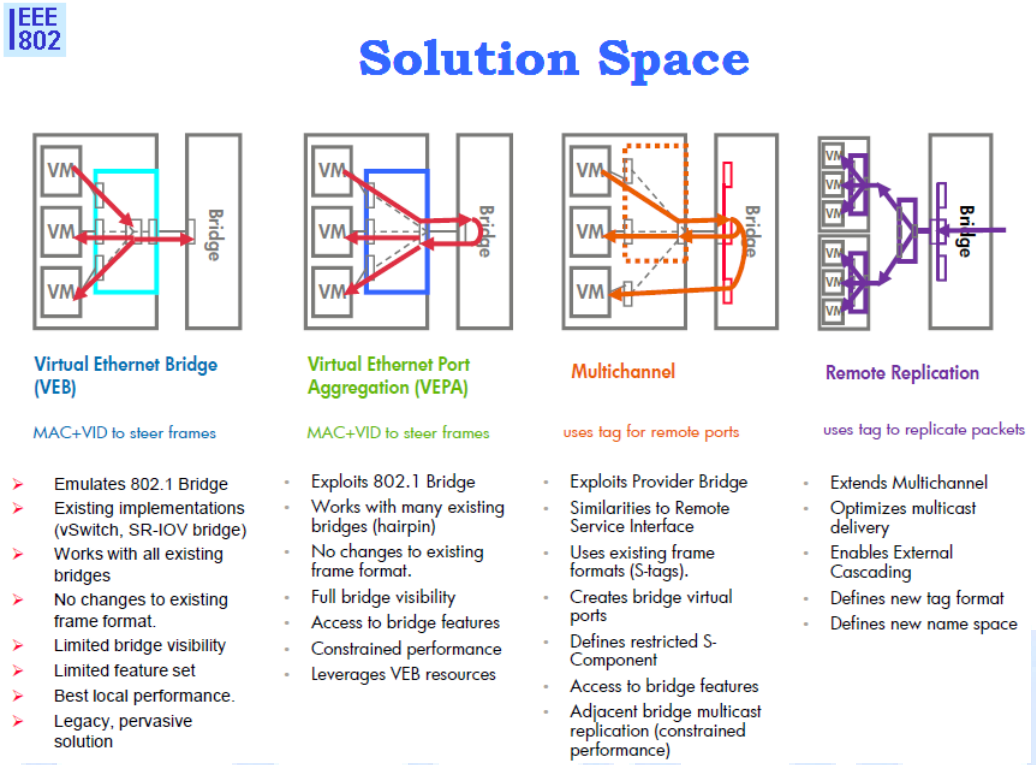
Contributors and Supporters

Siamack Ayandeh	(3Com)	Charles R. (Rick) Maule	(consultant)
Guarav Chawla	(Dell)	Menu Menuchehry	(Marvell)
Paul Congdon	(HP)	Shehzad Merchant	(Extreme)
Dan Daly	(Fulcrum)	Vijoy Pandey	(BNT)
Claudio DeSanti	(Cisco)	Joe Pelissier	(Cisco)
Uri Elzur	(Broadcom)	Peter Phaal	(InMon)
Norm Finn	(Cisco)	Renato Recio	(IBM)
Ilango Ganga	(Intel)	Rakesh Sharma	(IBM)
Anoop Ghanwani	(Brocade)	Jeelani Syed	(Juniper)
Leonid Grossman	(Neterion)	Patricia Thaler	(Broadcom)
Chuck Hudson	(HP)	Neil Turton	(Solarflare)
Brian L'Ecuyer	(PMC-Sierra)	Manoj Wadekar	(QLogic)
Pankaj K Jha	(Brocade)	Martin White	(Marvell)
Jeffry Lynch	(IBM)	Robert Winter	(Dell)
David Koenen	(HP)		

802.1Qbg当时的主要撰写人是HP的Paul Congdon，不过最近几稿主要Draft已经由Paul Bottorff取代。其中Cisco的Joe Pelissier也是802.1Qbh的主要撰写人，而Bottorff也同样参与了802.1Qbh的撰写工作。单从技术上讲，这二者并不是对立的，而是可以互补的，上述两位HP和Cisco的达人都正在为两种技术结构融合共存而努力。具体可以访问 http://www.ieee802.org/1/pages/dcbridges.html 对这两个处于Active阶段的Draft进行学习。

802.1Qbh通过定义新的Tag（VN-Tag）来进行接口扩展，这样就需要交换机使用新的转发芯片能够识别并基于此新定义Tag进行转发，因此目前除了Cisco自己做的芯片外，其他厂商都无法支持。只有等Broadcom和Marvel等芯片厂商的公共转发芯片也支持了，大家才能跟进做产品，这就是设备厂商有没有芯片开发能力的区别。而802.1Qbg就走了另外一条路，搞不定交换机转发芯片就先想办法搞定服务器吧。下面从IEEE截取的图中可以看到EVB的

四个主要组成部分，也可以看做四个发展阶段。当前处于VEPA的成长期，已经出现部分转化完成的产品，而Multichannel还在产品转化前的研究状态。

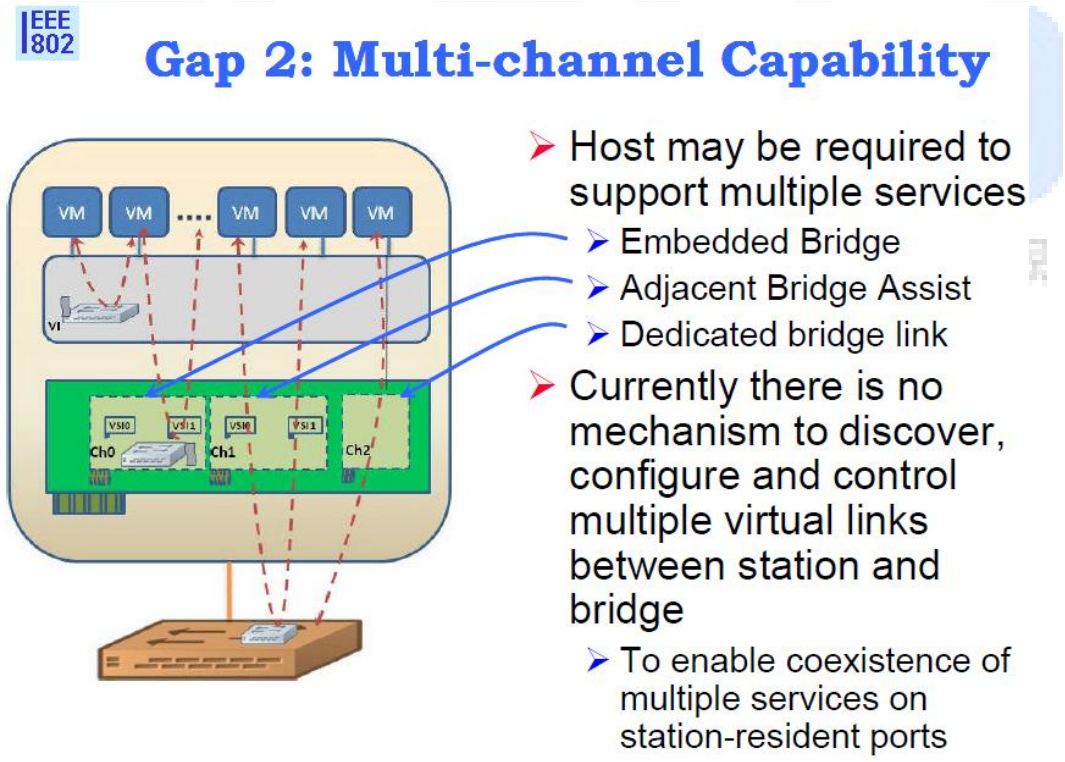


先说VEB，这个最好理解，就是定义了物理服务器内部的软、硬件交换机。软件交换机就是前面提过的VSwitch，硬件交换机就是从SR-IOV演进来的网卡交换机。SR-IOV已经可以使VM的vNIC在物理网卡上一一对应通道化了，那么再加个转发芯片基本就可以做成最简单的交换机了，当然这只是原理上可行，实际中作者还没有见过成熟产品。VEB与普通Ethernet交换机的最大区别是定义了连接交换机的上行口与连接VM的下行口，而VEB的上行口间是不允许相互转发报文的，这样可以在不支持STP的情况下保证无环路产生。Cisco的N1000V 就可以认为是个VEB。VEB的优点是好实现，在Hypervisor层面开发软件或者改造网卡就可以出成品，缺点是不管软件的还是硬件的相比较传统交换机来说能力和性能都偏弱，网卡上就那么点儿大的地方，能放多少CPU、TCAM和ASIC啊。

于是有了VEPA，VEPA比VEB更简单，不提供VM间的交换功能，只要是VM来的报文都直接扔到接入交换机上去，只有接入交换机来的报文才查表进行内部转发，同样不允许上行接口间的报文互转。这样首先是性能提升了，去掉了VM访问外部网络的流量查表动作。其次是将网络方面的功能都扔回给接入层交换机去干了，包括VM间互访的流量。这样不但对整体转发的能力和性能有所提升，而且还解决了前面最开始VSwitch所提出的网络与服务

器管理边界的问题。相比Cisco将网络管理推到VM的vNIC前的思路，这种做法更传统一些，将网络管理边界仍然阻拦在服务器外面，明显是出于服务器厂商的思路。在传统Ethernet中，要求交换机对从某接口收到的流量不能再从这个接口发出去，以避免环路风暴的发生。而VEPA的使用要求对此方式做出改变，否则VM之间互访流量无法通过。对交换机厂商来说，这个改变是轻而易举的，只要变动一下ASIC的处理规则即可，不需要像VN-Tag那样更新整个转发芯片。从理论上讲，如VEB一样，VEPA同样可以由支持SR-IOV的网卡来硬件实现，而且由于需要实现的功能更少，因此也更好做一些。个人认为VEPA的网卡可能会先于VEB的网卡流行起来。

下面说说Multichannel，这个东西就有点儿意思了。802.1Qbg的说法是在混杂场景中，物理服务器中同时有VEB、VEPA和需要直接通过SR-IOV连到交换机的vNIC，而当前对这多种流量在网卡到交换机这条链路上是无法区分识别的，于是整出个Multichannel。参见IEEE的胶片原文如下：

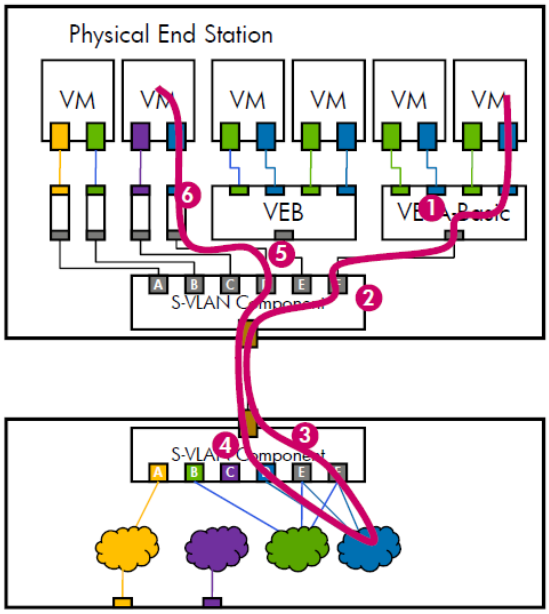


想在一条通道内对相同类型流量进行更细的分类，看了前面技术理解一节大家应该有个思路了，加Tag呗。Multichannel借用了QinQ中的S-VLAN Tag（就是个VLAN标签）。在数据报文从网卡或交换机接口发出时封装，从对端接口收到后剥离。简单的转发过程如下：

IEEE 802

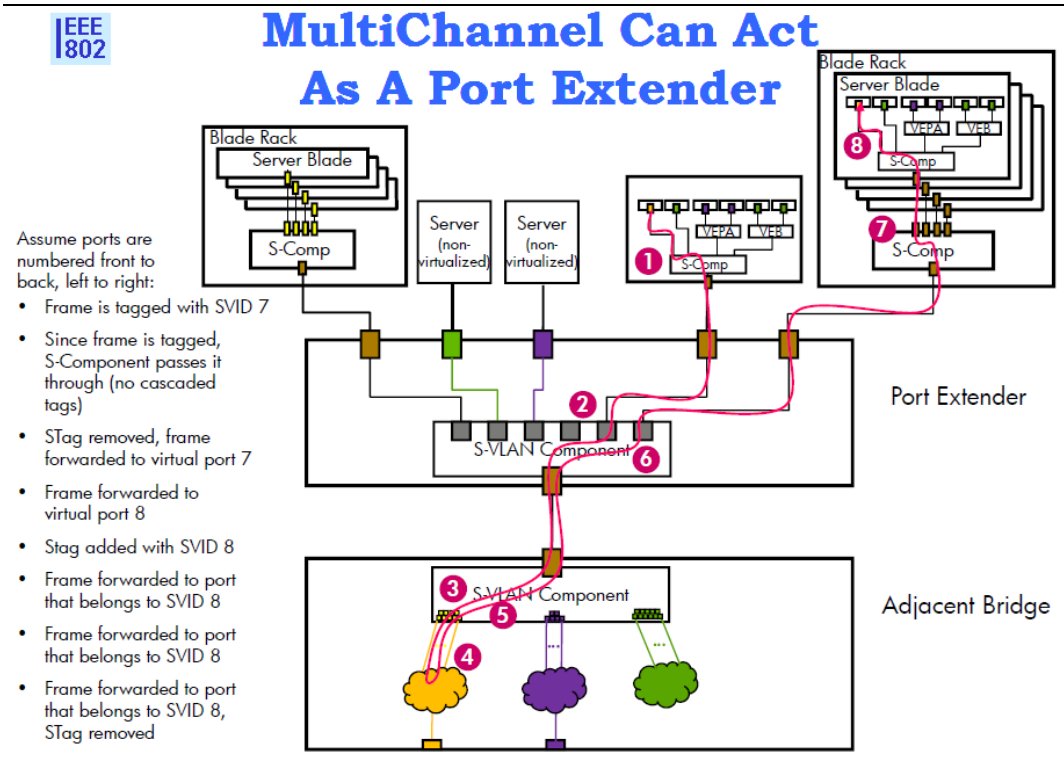
MultiChannel Approach

Example: VM through VEPA to Directly Accessible VSI



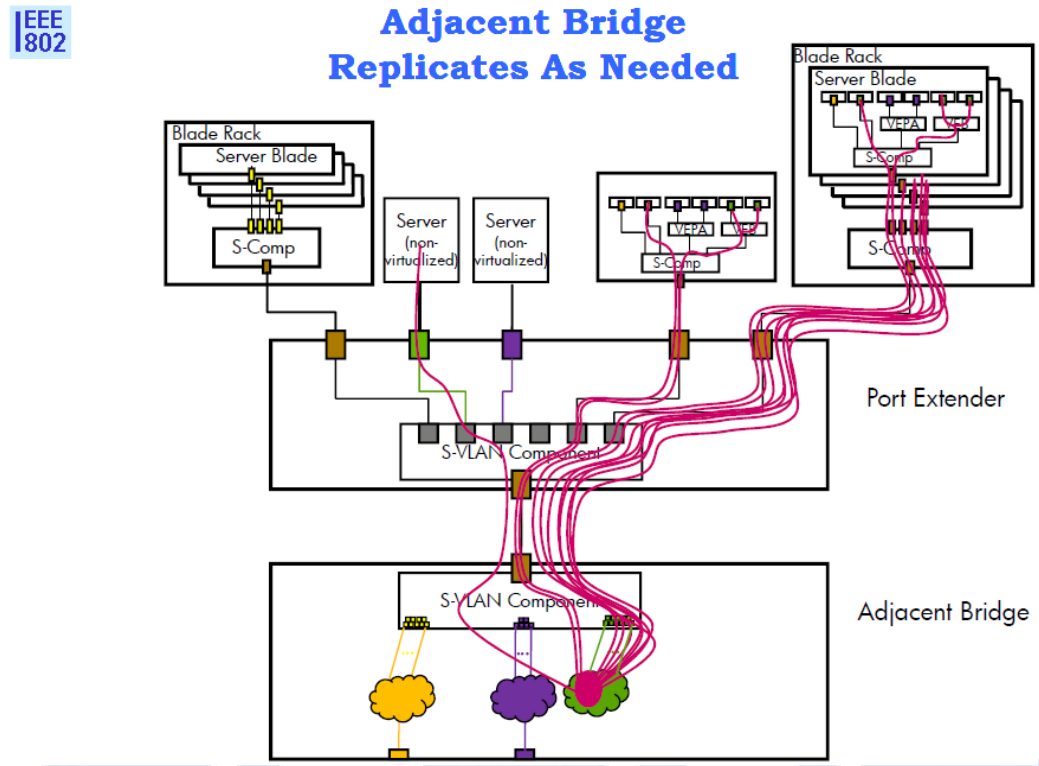
1. VEPA ingress frame from VM forwarded out VEPA uplink to S-Component
2. Station S-Component adds SVID (F)
3. Bridge S-Component removes SVID and forwards to port F
4. Frame is forward back to port D, S-Component adds SVID D
5. Station S-Component removes SVID D
6. S-Component forwards frame on Port D on Blue VLAN.

诸位看官看到这里可能会产生疑问，这个和Cisco的BPE很像啊，无非是用S-VLAN取代了VN-Tag作用在网卡和交换机之间。作者个人觉得Multichannel真正瞄准的目标也不是什么多VEB和VEPA之间的混杂组网，至少目前做虚拟化的X86服务器上还没有看到这种混杂应用的需求场景。真正的目标应该就是通过S-VLAN Tag建立一条VM上vNIC到交换机虚拟接口的通道，和Cisco FEX+VN-Link的目标是等同的，只是没有考虑网络接入层上面的FEX扩展而已。Cisco的达人Joe Pelissier目前在EVB工作组中做的事情也是将其与802.1Qbh在Port Extend方面做的尽量规则一致。可参考如下胶片内容：

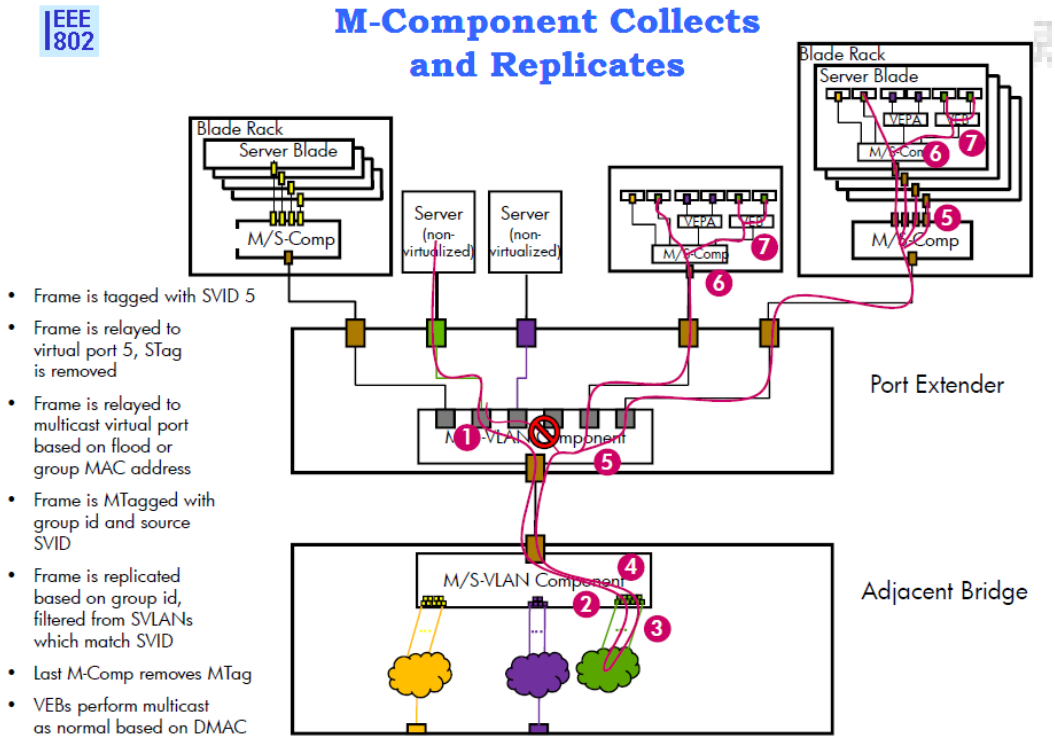


Multichannel相比VN-Tag的优势是交换机目前大部分的转发芯片就已经支持多层VLAN标签封装的QinQ技术了，而网卡封装VLAN Tag也是现成的，只要从处理规则上进行一些改动就可以完全实现。但由于其未考虑网络方面的扩展，S-VLAN还不能进行交换机透传，只能在第一跳交换机终结，所以从接入层网络部署规模上很难与FEX抗衡。

最后一个是Remote Replication复制问题。Ethernet网络当中广播、组播和未知单播报文都需要复制，而前面的Multichannel结构所有的复制工作都会在交换机完成，于是会造成带宽和资源的极大浪费，如下图所示：



此时需要定制一个标志位，以通知每个S-VLAN组件进行本地复制。当前在802.1Qbg中此标志位叫做M标识，其实和VN-Tag字段中的p标志位一个作用，因此这块也是由Cisco的达人Joe Pelissier在完善。M标识的位置和作用如下图所示：



整个EVB就这些内容了，虽然目前还是在不断改动，但都是些实现细节方面的东西，大

的框架结构应该就如上所述不会再变化了。从工作内容上就可以看出，HP等服务器厂商的思维范畴就到VEPA了，Multichannel只是提出个概念，实际上后续的东东还都要是借鉴Cisco的网络思路来实现，个人甚至怀疑连Multichannel都可能是Pelissier提出的，谁专业谁知道。从大体上来说，EVB总的思路就是希望在尽量对现有设备最小变更的情况下解决VM接入互访的问题，从改变Ethernet交换机接口转发规则到增加S-VLAN标签都是如此，但到M标识就不见得还能控制得住了。不过就目前协议技术完善和进行产品转化的速度来看，还有得时间进行考虑变化。

5.4.3 小结

又到了小结的时间。该有人问了，讲了半天802.1Qbh和802.1Qbg这两个技术体系谁优谁劣，谁胜谁败啊？作者不是半仙也不是裁判，只能推测无法判断。从技术角度讲，尤其从网络技术角度讲，802.1Qbh BPE提出了一整套的网络虚拟化解决方案，而802.1Qbg EVB则只是提出了解决几个VM网络接入问题的办法，二者的技术深度不可同日而语。然而对于市场上来说，一时的技术优势并不能完全左右胜负，各方博弈会使结局充满不可预测性，Nortel和3Com的没落就是例子。从一名技术至上者的角度出发，作者更倾向于Cisco，不过一切都有待于市场的检验。当然在这个世界上还有些地方存在可以代表市场直接做出裁决的裁判们，他们的裁决结果看看各公司在当地的财务贡献就可以简单预测，和真正的市场选择有没有关系，你懂的。

顺便说一句，802.1Qbh需要变更交换机的转发芯片以适应VN-Tag转发，而802.1Qbg的VEPA和Multichannel目前则只需要交换机做做软件驱动方面的变动即可支持。不论将来谁成为了市场技术主导，大家觉得Cisco设备同时支持两套标准会有什么难度。从市场发展上大胆预测一下，Cisco会在802.1Qbg标准成熟后，在新一代N2000位置产品上实现对S-VLAN组件和M-Tag的支持，以后的主流结构就是服务器内部用VEPA+SR-IOV，网卡和N2000之间使用S-VLAN区分通道，N2000再往上到N5000还是封装VN-Tag的FEX。BPE+EVB才是王道。

YY一下，还有木有啥其他的技术可以起到类似的作用呢？目前802.1Qbg和802.1Qbh都是通过定义一个Tag来为接入层交换机标识VM（VN-Tag/S-VLAN），那么实际上还有个现成的可用Tag，就是MAC，每个VM的vNIC都拥有独一无二的MAC，那么是否可以让交换机根据源和目的MAC来建立VIF去对应处理每个VM的vNIC流量呢。当然细节上还要设计很多

机制来保障各种情况的正常处理，但是暂时从方向上感觉应该有些搞头，回头有时间细琢磨一下吧。能不能成标准无所谓，当做头脑游戏锻炼思维了。

短期来看，上述厂家标准有得一争，但从长远来看，其实硬件交换机进入服务器内部才是王道。毕竟转发芯片会越来越便宜，性能会越来越高，再发展几年，不管是放在网卡上还是集成在主板上都没有太大难度。

5.5 Ethernet 与 FC 网络融合技术-FCoE

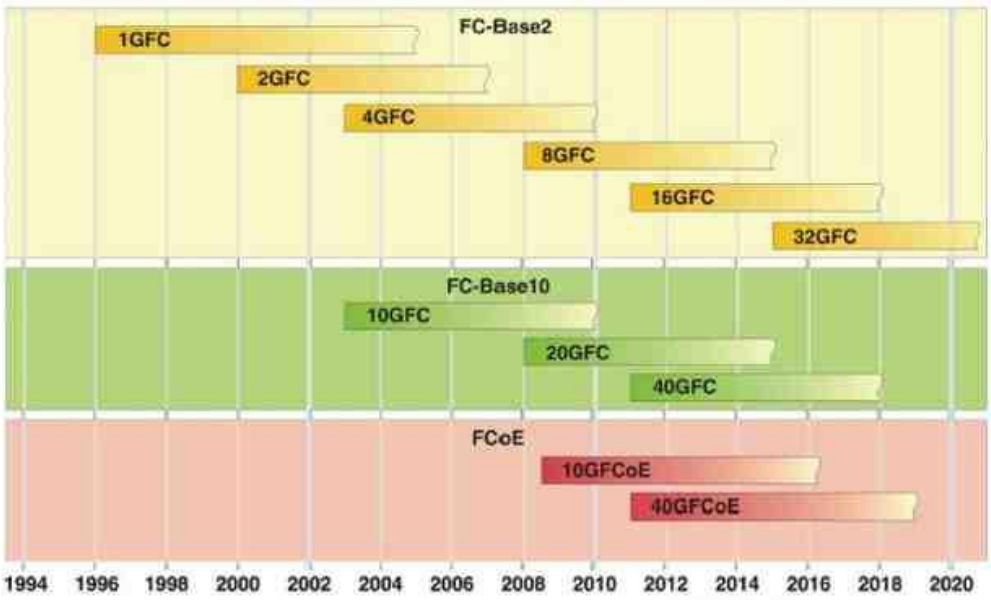
本章节重点技术名词：FC/FC ID/FCoE/FIP/FCF/DCB/NPV

服务器后端连接存储设备的FC网络与前端Ethernet网络融合是目前传统以太网交换机厂商进军后端存储网络的阳谋，Cisco称其为统一IO。下面会先介绍下FC，然后是FCoE，最后说说NPV。

5.5.1 FC

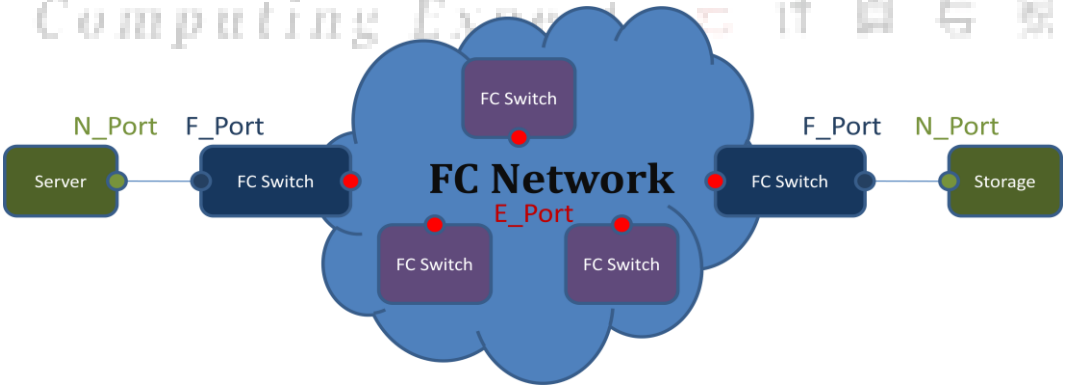
FC（Fibre Channel）在1994年由ANSI T11制定首个标准，从开始就达到1G带宽，大家可以想想那时候Ethernet还是什么年代。同时由于其无丢包的协议特性，受到了存储网络的青睐，成为Server到Storage之间SAN网络的霸主，相信由于Ethernet 40G/100G的缓慢发展速度，FC的地位至少在下个5年内还是无可动摇的。在FC通信领域是典型的寡头独占市场，前三位中Brocade占据了70%的市场份额，Cisco占20%，Qlogic占个位数。2010年统计整个FC Switch市场销售额约$2B，对众多的传统交换机厂商来说，谁不想通过FCoE进去分杯羹呢，即使是Cisco也忍不住想翻身把歌唱。

FC包含Base2与Base10两套演进道路，Base10主要应用在FC交换机之间，使用较少，已经快消亡了。下面截图可以看出其演进情况。



FC拥有自己的独立层次结构，FC-0到FC-4对应OSI模型的1-5层，但也并非一一对应，完整协议内容请大家自行查阅标准文档。其中FC-2定义了数据通信的内容，是与网络方面息息相关的，下面介绍的内容也都是以FC-2为主。

在FC网络中一共有三种主要的接口角色，NPort，FPort和EPort，其中N是服务器或存储等终端节点连接FC网络的的接口，F是FC交换机设备连接服务器或存储等终端节点的接口，E是FC交换机互联接口。还记得前面技术理解里面的典型结构么？



FC设备都拥有2个重要标识：

WWN（World Wide Name）：64bit，节点和每个接口都有各自固定的WWN且所有的WWN均是唯一的，WWN的作用是为了身份识别和安全控制，有些类似于MAC，但不做转发寻址使用。

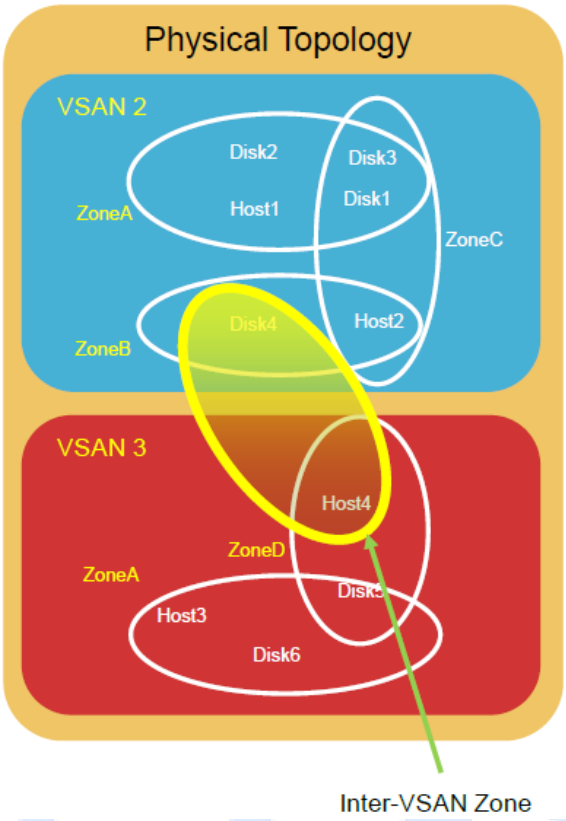
FC ID：24bit，由8个bit的Domain ID，8bit的Area ID和8bit的Port ID组成，每个Domain ID代表一台FC Switch（由此可以算出每个FC网络最多支持256个Switch节点，减去部分保留ID，

实际能够支持最多239个Switch）。终端节点的FC ID是基于接口的，每个NPort的FC ID是由直连的FC Switch动态分配。FC ID的主要作用就是供数据报文在FC网络中寻址转发。

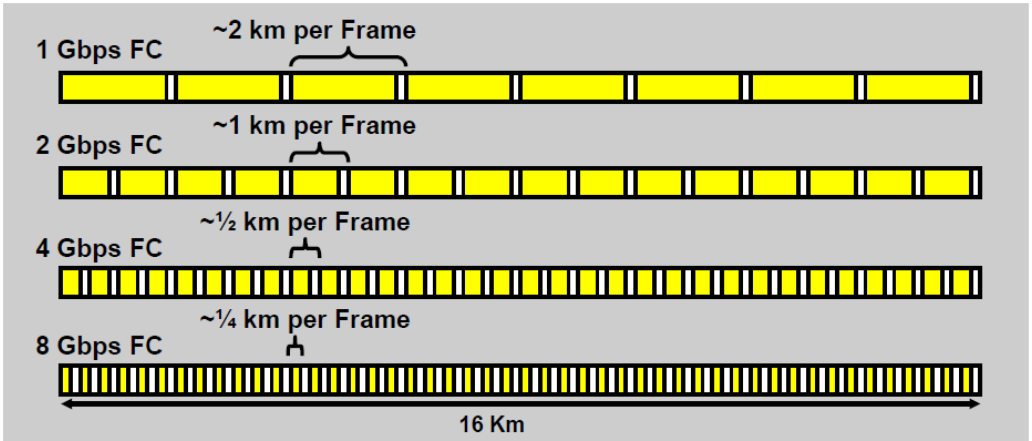
有了标识的Tag，那么就需要个动态协议供FC Switch互相学习了，FC网络使用FSPF（Fabric Shortest Path First）进行FC ID的寻址学习，看名字就知道其协议机制和OSPF没有什么大的区别，不多说了。

FC网络中的另外两个重要概念就是VSAN和Zone，VSAN和VLAN很类似，都是手工配置的，不同VSAN的流量相互隔离，这样不同VSAN中可以分配相同的FC ID。而且由于VSAN是非公有定义协议字段，各个厂家实现并不见得一致，因此实际的FC组网中很难见到不同厂商设备的混合组网。Zone则是类似于ACL的安全特性，配置为同一个Zone的成员可以互访，不同Zone的就会被隔离。Zone是作用于VSAN内部的，可以理解VSAN是底层物理隔离，Zone是上层逻辑分隔。同一个设备节点可以属于不同的Zone，Zone成员以WWN进行标识，可以简单类比为ACL中的同一个源IP地址可以配置在不同的Rule中对应不同的目的IP，以匹配不同的流量。Zone的控制可以是软件实现，也有相应的ASIC可以做硬件处理。

当然有隔离还得能互通，就好像做了VPN后还惦记着跨VPN互访，有了FW还搞个HTTPS翻墙（目前又在研究怎么控制HTTPS内容了），FC中也有IVR（Inter-VSAN Routing）Zone的概念，就是通过一些静态配置的手段翻越已有的VSAN隔离。人们总是处在不断的制造藩篱和打破藩篱的循环中。Zone、VSAN和IVR Zone的关系如下Cisco资料截图所示：

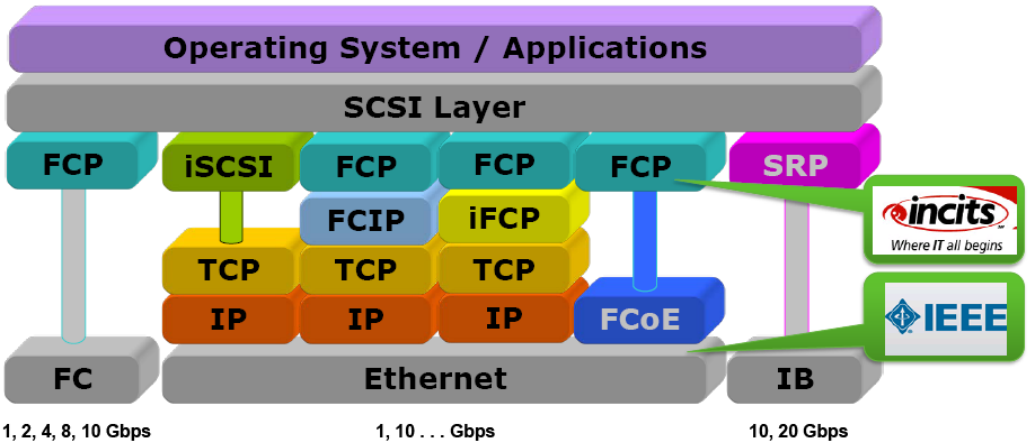


FC技术体系还有最重要的一个关键流控技术Buffer to Buffer Credits用来确保无丢包转发。BB Credits和TCP滑动窗口相似，规则很简单，两个相邻FC节点在连接初始化的时候先协商一个度量收包设备Buffer大小的数值N出来，发包设备每发一个数据报文就做N-1，收包设备每收一个报文就回一个R_RDY报文回来，发包设备每收到一个R_RDY就做N+1，当N=0时，发包设备就停止发包。这样当突发拥塞时，上游设备们都把报文存在本地缓存中等着，下游有空间时再发，可以最简单的避免丢包。BB Credits是以报文数目衡量buffer能力，与报文长度无关（FC报文最大长度2112Byte）。另外Credits协商数目大小与带宽和距离存在比率关系，可参考如下图示的Cisco建议：



FC设备（一般指服务器，称为Initiator）在传输数据之前需要进行两步注册动作，NPort先通过FLOGI（Fabric Login）注册到最近的Fabric交换机上，获取FC ID及其他一些服务参数并初始化BB Credits。然后再通过PLOGI（Port Login）注册到远端的目的设备（一般指存储，称为Target）的NPort上建立连接，并在P2P直连的拓扑下初始化BB Credits。

FC从标准建立伊始就开始被研究跨传统TCP/IP/Ethernet网络传播，目前主要有iSCSI（IP SAN）、FCIP、iFCP和FCoE四条道路。其中FCIP和iFCP应用最少，iSCSI缓慢增长，FCoE后来居上。

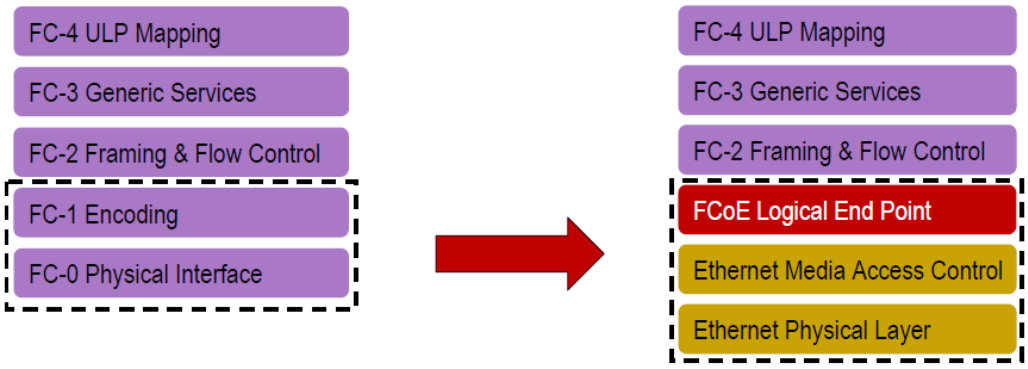


SCSI不熟，这里不多说。FCP（Fibre Channel Protocol）是用来协助SCSI进行寻址的协议。iSCSI、FCIP和iFCP都是依靠TCP的可靠连接确保无丢包，但封的报头多了开销很大。iSCSI由于需要全新的存储设备支持，过于激进，目前虽然有发展，但是受传统存储设备厂商制约始终很缓慢。FCIP和iFCP都是支持FC网络跨IP核心网传输时用到的网络协议，由于目前SAN还是本地组网或使用光纤直连方式的远程组网较多，此场景并不多见，因此也应用很少，其中FCIP已经成为RFC，而iFCP止步于Draft。FCoE相比较来说对上层协议改动较少，开销较低，且有利于减少服务器网络接口数量，在传统交换机厂商的大力鼓吹下当前发展最为迅猛，数据中心网络毕竟会是交换机的天下。

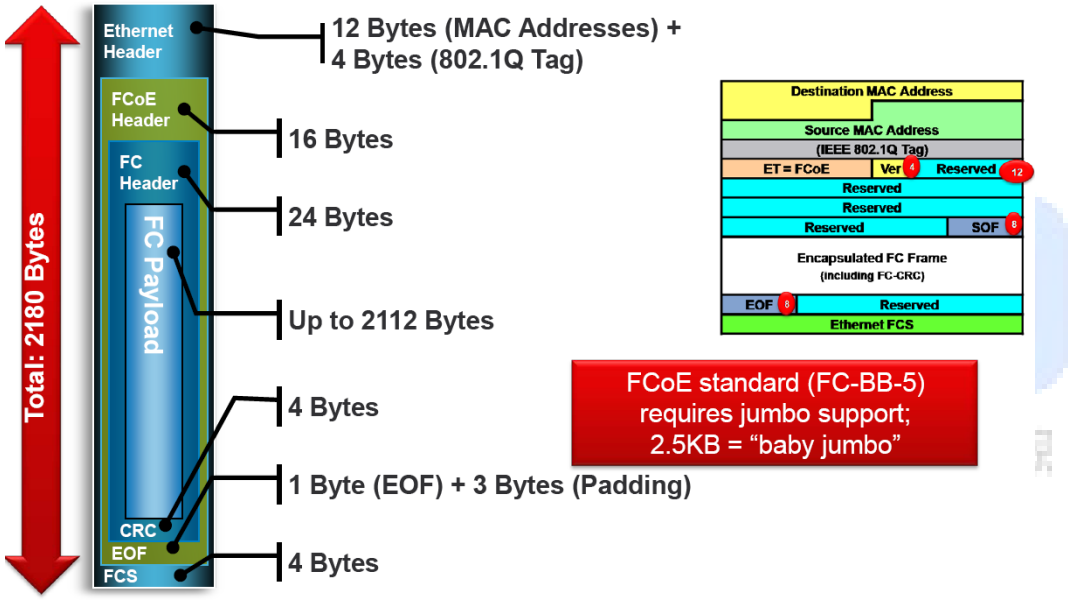
5.5.2 FCoE

FCoE是在2007年INCITS(国际信息技术标准委员会)的T11委员会（和FC标准制定是同一组织）开始制定的标准，2009年6月标准完成（FC-BB-5）。FCoE基于FC模型而来，仍然使用FSPF和WWN/FC ID等FC的寻址与封装技术，只是在外层新增加了FCoE报头和Ethernet

报头封装和相应的寻址动作，可以理解为类似IP和Ethernet的关系。



FCoE标准定义了数据平面封装与控制平面寻址两个部分。封装很好理解，大家看看下面这张图就了然了。

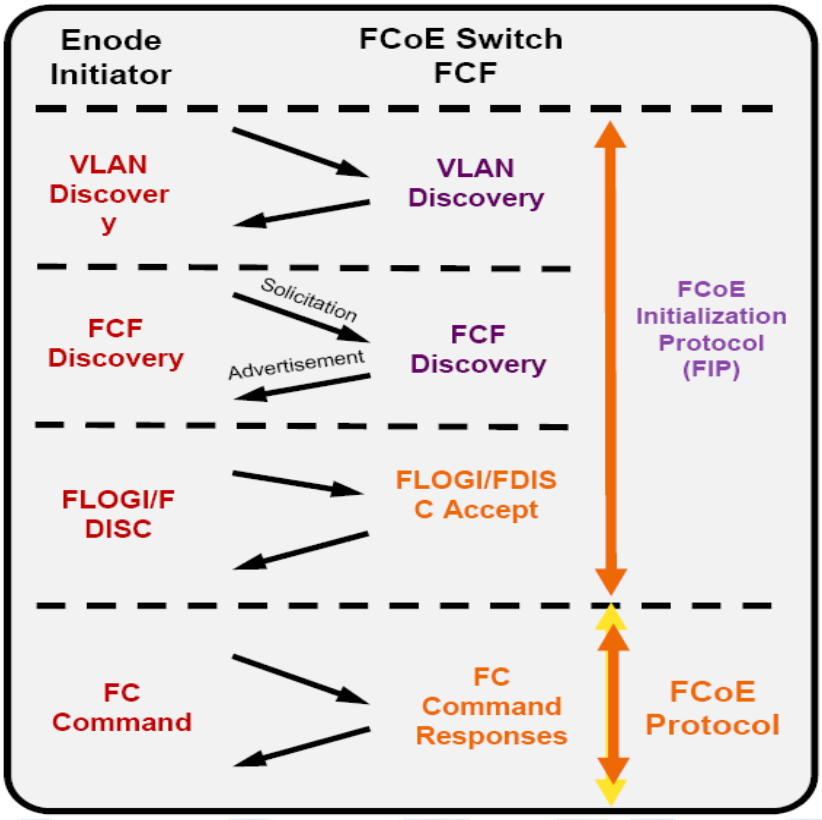


寻址稍微说一下，FCoE使用FIP（FCoE Initialization Protocol）进行初始化连接，FIP运行于VFPort和VNPort之间或VEPort之间，所谓的V就是前面介绍FC的接口角色中的名称前面加了个Virtual。FIP在接口使能后一共做了三件事：

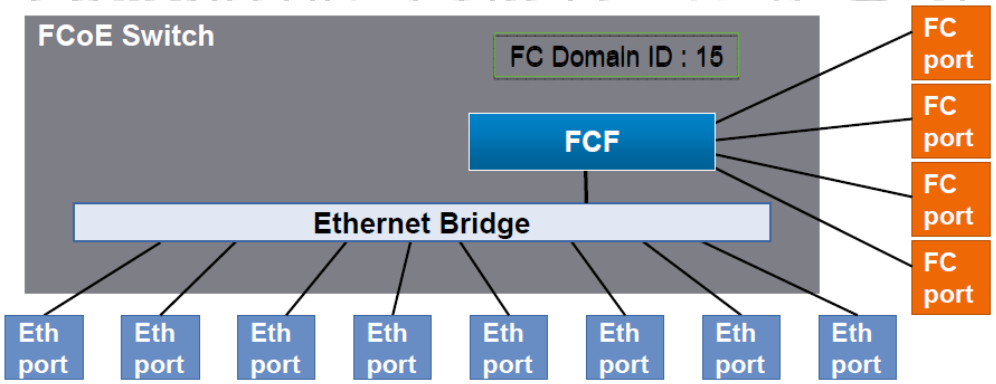
1、使用本地VLAN（如VLAN1）确认FCoE数据报文将要使用的VLAN ID。

2、和FCF建立连接。

3、FLOGI/FDISC（Discover Fabric Service Parameters，FC节点设备第一次向FC交换机注册请求FC ID时使用FLOGI，后面再续约或请求其他FC ID时都使用FDISC）



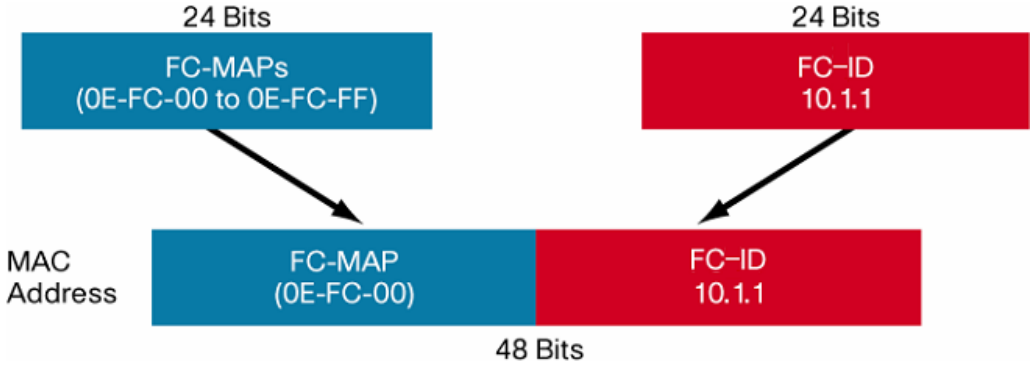
FCF（Fibre Channel Forwarder）是FCoE里面重要的角色，可以是软件或者芯片硬件实现，需要占用Domain ID，处理FCoE交换机中所有与FC相关的工作，如封装解封装和FLOGI等。



Enode是指网络中所有以FCoE形式转发报文的节点设备，可以是服务器CAN网卡、FCoE交换机和支持FCoE的存储设备。FCoE外层封装的Ethernet报头中MAC地址在Enode间是逐跳的，而FC ID才是端到端的。（不好理解就琢磨下IP/Ethernet转发模型，将Enode想成路由器和主机，一样一样滴）

与三层交换机中的VLAN接口一样，每个FCF都会有自己的MAC，由于FC ID是FCF分

配给Enode的，继承下来的终端Enode MAC也是由FCF分配的并具有唯一性，这个地址叫做FPMA（Fabric Provided MAC Address）。FPMA由两部分组成，FC-MAP与FC ID，结构如下所示，这样当FCoE交换机收到此报文后可以根据FC-MAP判断出是FC报文，直接送给FCF，FCF再根据FC ID查表转发，处理起来更简单。

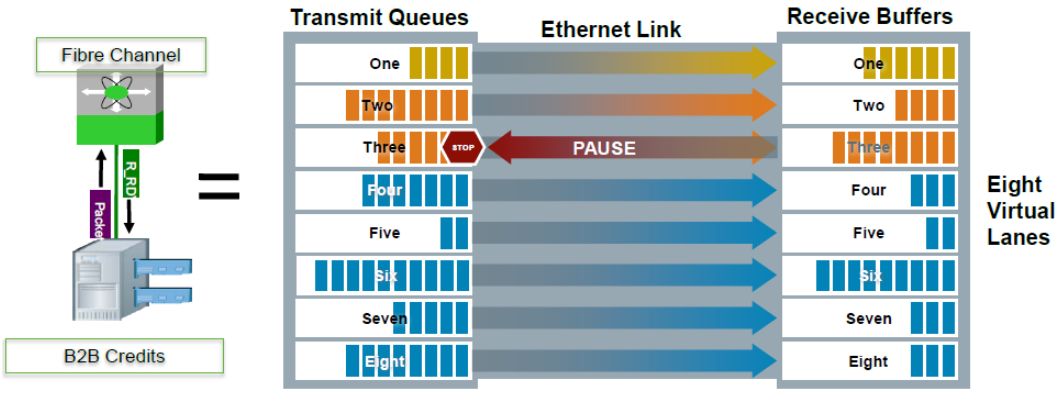


由上面FC-MAP的定义也可以看出，每个FCF下联的Enode终端最多也就255个（00-FF）。

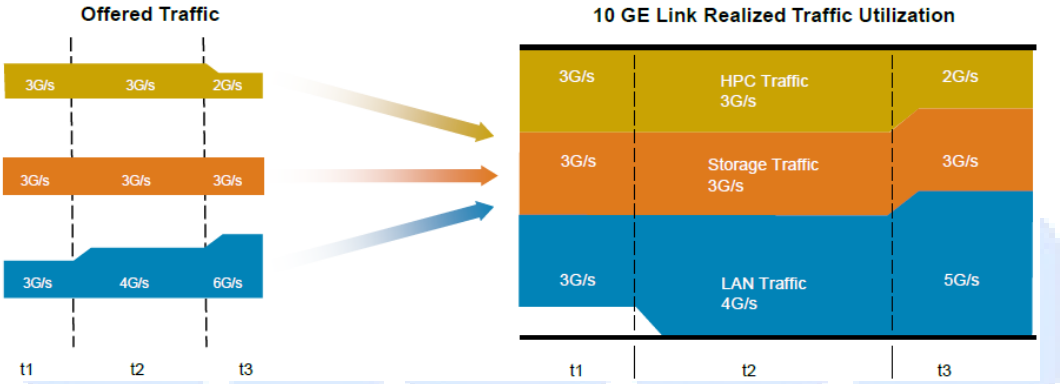
DCB

由于Ethernet是冲突丢包的，为了保证FCoE的无丢包，IEEE引入了一系列的无丢包以太网技术（Lossless Ethernet），都定义在802.1Q DCB（Data Centre Bridging）标准系列中。DCB等同于DCE（Data Centre Ethernet）和CEE（Converged Enhanced Ethernet）的含义，就是不同厂商和工作组的不同称谓，内容都是一致的。DCB是IEEE为了在数据中心对传统以太网技术进行扩展而制定的系列标准，前面说过的VM接入技术标准中802.1Qbg和802.1Qbh都是DCB中的一部分，另外还有802.1Qau CN（Congestion Notification），802.1Qaz ETS（Enhanced Transmission Selection）和802.1Qbb PFC（Priority-based flow control）。其中802.1Qau CN定义了拥塞通知过程，只能缓解拥塞情况下的丢包，加上其必须要全局统一部署与FCoE逐跳转发的结构不符，因此不被算成无丢包以太网技术的必要组成部分。常见的无丢包技术主要是PFC和ETS，另外还有个DCBX（Data Center Bridging Exchange Protocol）技术，DCBX也是一起定义在802.1Qaz ETS标准中。

PFC对802.3中规定的以太网Pause机制进行了增强，提供一种基于队列的无丢包技术，实际达到的效果和FC的BB Credits一样。简单理解如下图所示。



ETS是带宽管理技术，可以在多种以太网流量共存情况下进行共享带宽的处理，对FCoE的流量报文进行带宽保障。简单理解如下图所示。

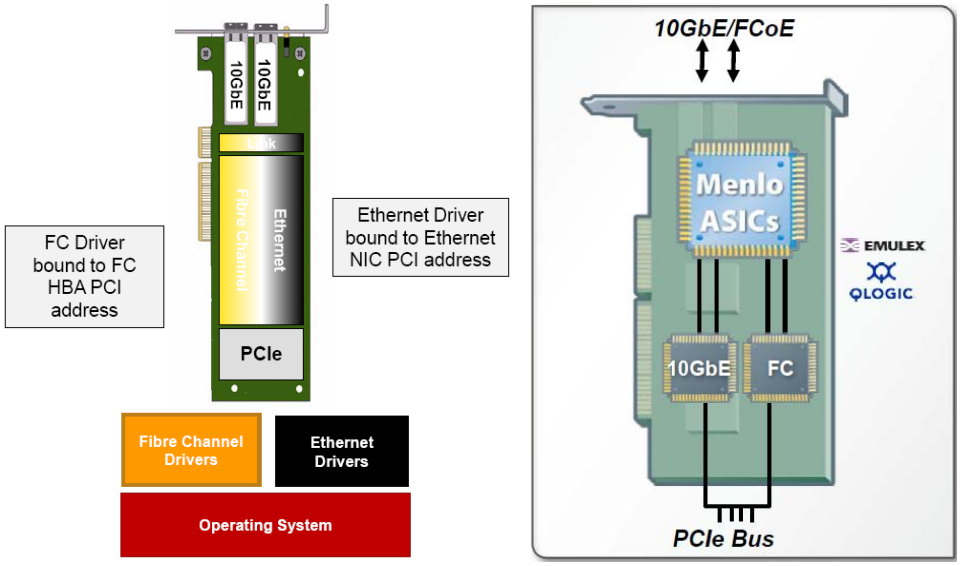


DCBX定义了通过LLDP在两个相邻Enode之间进行PFC, ETS等参数自协商交互的过程。

DCB的几个标准目前都还处于Draft阶段，其中PFC是由Cisco的Claudio DeSanti主编，ETS由Qlogic的Craig Carlson主编。

CNA

再补充一句服务器上的FCoE网卡CNA（Converged Network Adapter），这个东西就是万兆Ethernet和FC HBA（Host Bus Adapter）网卡的合体，里面包含两个独立芯片处理Ethernet和FC各自的流量，在操作系统上看到的就是两个独立的Ethernet和FC网络接口，其上再增加第三个芯片进行流量混合封包处理。可参考下面两张图示。

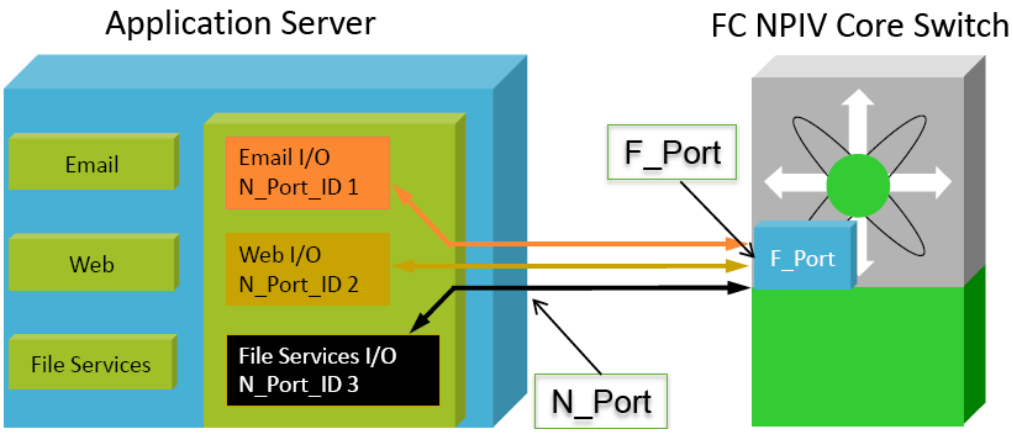


FCoE的技术要点就这么多了，需要记住的关键是，FCoE标准提供的是服务器到存储设备端到端的网络连接模型，FCF是FCoE交换机的关键特性。目前已经有支持FCoE的存储设备出现，估计服务器到存储的全FCoE商用项目组网也很快就会在市场上出现。

5.5.3 NPV

目前市面上80%以上的标榜自己实现了FCoE的交换机产品其实都是只实现了NPV功能，和前面描述的FCoE标准内容沾不上多少边儿。那么NPV是啥呢？

先说NPIV（NPort ID Virtualization），这个还是FC里面的概念。前面说了Server的NPort需要向FC Switch进行FLOGI注册获取FC ID进行路由，那么如果一台物理服务器里面搞了好多虚拟机后，每个VM都打算弄个FC ID独立通信，但只有一块FC HBA网卡咋办呢。FC中通过NPIV解决了这种使用场景需求，可以给一个NPort分配多个FC ID，配合多个pWWN（private WWN）来进行区分安全控制。

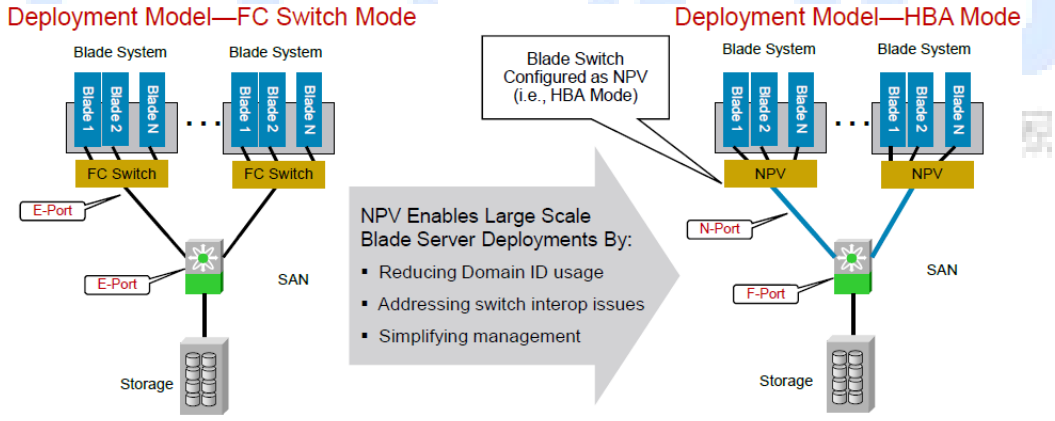


理解了NPIV就好说NPV了，我们把上图中的NPort拿出来作为个独立设备给后面服务器代理进行FC ID注册就是NPV（NPort Virtualization）了。NPV要做的两件事：

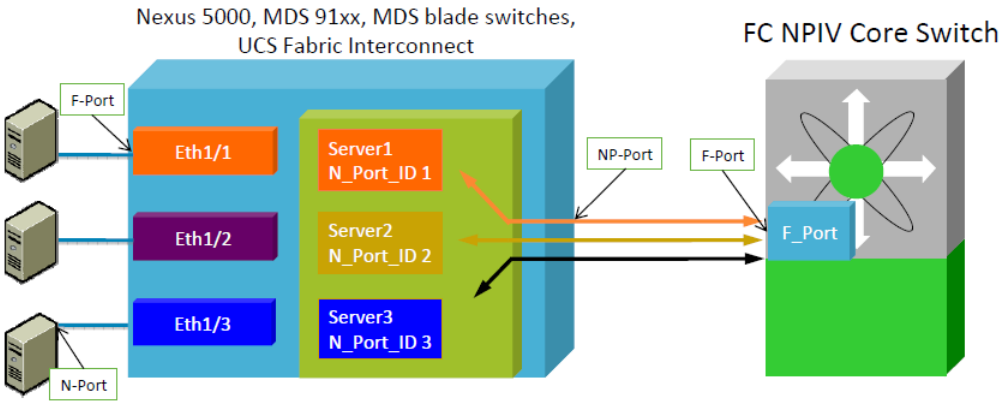
1、自己先通过FLOGI向FC Switch注册去要个FC ID

2、将后续Server过来的FLOGI请求代理成FDISC请求，向FC Switch再去申请更多的FC ID

NPV的好处是可以不需要Domain ID（每个FC区域最多只有255个），同时能将FC交换机下联服务器规模扩大。NPV在FC网络中最常见的应用是在刀片交换机上。



随之有人将FCoE的脑筋动到了NPV与服务器之间的网络上，如下图所示：



在FCoE中的NPV相比较FC中要多做三件事，参考前面FIP流程：

1、回应节点设备关于FCoE承载VLAN的请求

2、回应节点设备的FCF查找请求，根据自己初始化时从FC Switch得到的FC ID生成仿冒FCF使用的MAC地址

3、在CNA网卡和FC Switch之间对转发的数据报文进行FCoE头的封包解包。

NPV不是FCoE标准中定义的元素，因此各个厂家在一些细节上实现起来都各玩各的。比如都是将连接服务器的Ethernet接口和连接FC Switch的FC接口绑定起来使用，但是对应的绑定规则就可能不同。再有如FC接口故障时，如何将服务器对应的通道切换到其他FC接口去，是否通知服务器变化重新进行FLOGI注册，及通知等待时长等设定都会有所区别。

说说NPV的好处，首先是实现容易，之前描述的那几件主要的任务现在都已经有公共芯片可以直接搞定，所以包装盒子就是了。其次是部署简单，不需要实现FCF，不用管FC转发，不计算FSPF，不占Domain ID。最后是扩展方便，使用FC Switch的少量接口就可以连接大量的服务器。

由于NPV与服务器之间网络为传统以太网，因此NPV交换机也必须支持DCB标准中相关的无丢包以太网技术。

严格来讲，NPV交换机不是FCoE标准中定义的FCoE交换机，但可以在接入层交换机上实现与服务器之间的Ethernet网络复用，减少了服务器的物理网卡数量（并未减少操作系统层面的网络通道数量），扩展了FC网络接入服务器节点的规模，适用于云计算大规模服务器部署应用。

补充一个ENPV（Ethernet NPV）的概念，这个东东是Cisco提的，就是在服务器与FCoE交换机（FCF）之间串个NPV进去，还是做些代理的工作，可以对FIP进行Snooping，监控FIP注册过程，获取VLAN/FC ID/WWN等信息，对过路流量做些安全控制啥的。这种东东存在既有理，但以后有没有搞头就不好说了，市场是检验技术的唯一标准。

5.5.4 小结

FCoE是端到端的，FCF是不可少的，NPV是干代理的，目前是最合适的。

个人觉得NPV这个东西真的很彪悍，设计一套FCoE标准虽然是技术含量很高的活儿，但第一个搞出来NPV的才是真正的人精。如果没有NPV，FCoE想从FC口里夺食难度至少会

增加上百倍，没准儿就跟iSCSI一样落得个鸡肋的地步。强烈建议FCoE标准尽快将NPV搞进来，要不单独出个FC-BB-7/8啥的独立标准体系也不错。

随着互联网的发展，对网络最大的需求就是带宽增长，云计算更是如此，因此如果FC的带宽演进继续这么不紧不慢的话，势必会被100G Ethernet取代，至于时间点就要看带宽需求增速了，个人估计不会超过10年，到时就有FCoE的用武之地了，至于之前这段时间应该都还是NPV在前面冲锋陷阵。

继续大胆YY，可否搞个什么技术冲击下FCoE呢？由于IP是无连接的，Ethernet是冲突丢包的，因此想保证数据传输的可靠性就只能像iSCSI一样在TCP上做文章，但是层次做高了，报头一多开销又太大，矛盾啊矛盾。如果完全替代Ethernet，那就类似于重建一套FC协议了，但是目前也看不到带宽速率发展能超过Ethernet的替代技术。那么中间手段就是搞下IP这个层面，像FCoE的思路可以理解为用SCSI/FCP替代TCP/IP在Ethernet上传输，由于SCSI/FCP这套协议和Ethernet已经很成熟了，只是搞个接口（FCoE报头）在中间承上启下就够了。不过FCoE需要引入无丢包以太网的设计，Pause帧会不会降低Ethernet的转发效率还不好说。作者思路是设计一套带连接状态的传输机制（类似TCP带重传确认，而且可以像IP/FC一样能够寻址）替代IP/FCP这个层面，上面还是承载SCSI，下面跑着传统以太网。不见得靠谱，仅供拓展一下思路，有兴趣的同学可深思。

5.6 跨核心层服务器二层互访

本章节重点技术名词：L2MP/ VSS/IRF/ vPC/ TRILL/SPB/FabricPath/QFabric/VDC/VPN

在服务器跨核心层二层互访模型中，核心层与接入层设备有两个问题是必须要解决的，一是拓扑无环路，二是多路径转发。但在传统Ethernet转发中只有使用STP才能确保无环，但STP导致了多路径冗余中部分路径被阻塞浪费带宽，给整网转发能力带来了瓶颈。因此云计算中需要新的技术在避免环路的基础上提升多路径带宽利用率，这是推动下面这些新技术产生的根本原因。

前面网络虚拟化章节部分提到了两个解决上述需求的思路。

首先是控制平面多虚一，将核心层虚拟为一个逻辑设备，通过链路聚合使此逻辑设备与每个接入层物理或逻辑节点设备均只有一条逻辑链路连接，将整个网络逻辑拓扑形成无环的树状连接结构，从而满足无环与多路径转发的需求。这种思路的代表技术就是VSS/IRF/vPC，前两者都是控制平面全功能同步的整机虚拟化技术，vPC则是精简后只处理控制平面与跨设

备链路聚合使用相关功能的多虚一技术。此类技术必定都是私有技术，谁家的控制平面都不可能拿出来完全开放。

另一个思路是数据平面多虚一，在接入层与核心层交换机引入外层封装标识和动态寻址协议来解决L2MP（Layer2 MultiPath）需求，可以理解这个思路相当于在Ethernet外面搞出一套类似IP+OSPF的协议机制来。对接入层以下设备来说，整个接入层与核心层交换机虚拟成了一台逻辑的框式交换机，Ethernet报文进Ethernet报文出，中间系统就是个黑盒，就好像IP层面用不着了解到Ethernet是怎么转发处理的一样。这种思路的代表技术是IETF（Internet Engineering Task Force）标准组织提出的TRILL和IEEE提出的802.1aq SPB两套标准，以及一些厂商的私有技术。如FabricPath是Cisco对TRILL做了一些变更扩展后的私有技术称谓（以前也有叫E-TRILL），QFabric则是Juniper的私有技术，推测是基于MACinMAC封装和自己搞的私有寻址协议来做的。

这里唠叨两句私有协议，从有网络那天开始，私有协议就始终相生相随。从早期的EIGRP和HSRP到现在的VSS/IRF/vPC/OTV/QFabric都是各厂家的私货。即使是标准还有IETF/IEEE等不同的标准化组织呢，哪会有厂家就大公无私到研究出个啥东西都恨不得全人类共享，逐利才是企业发展的根本。私有协议还有两个有趣的现象，首先单从技术角度来说，私有协议基本都代表了同时代同类技术的最先进生产力，如果是个落后的技术，只有脑袋进水了才会砸钱进去研发。还有就是只有在市场上占了一定地位的重量级厂商才会经常推自己的私有协议，而且推得越多也代表着其地位越高。

从市场上看，最早由于没得选，大家基本上都能接受使用私有协议，后来出于不想将所有鸡蛋放在一个篮子里的心理，开始对私有协议有了抵触情绪，尤其是运营商级别用户明确要求不能使用私有协议。但就目前随着云计算的爆炸式增长，数据中心网络技术面临着一次新的飞跃，传统技术已经无法满足需求，因此私有协议再次进入了人们的视线。预计随后几年中，新的云计算数据中心会以站点Site为单位来部署单一厂商设备，如可以看到全Cisco设备运行FabricPath/vPC的Site，全Juniper设备QFabric的Site，或全H3C设备运行IRF的Site等等，站点之间或对外再采用一些公有协议如MSTP、RPR、BGP等进行连接。

下面会分别介绍几项主要技术，顺序为控制平面多虚一技术VSS/IRF/vPC，数据平面多虚一技术TRILL/SPB/Fabric Path/QFabric和控制平面一虚多技术VDC。

5.6.1 控制平面多虚一技术

VSS/IRF

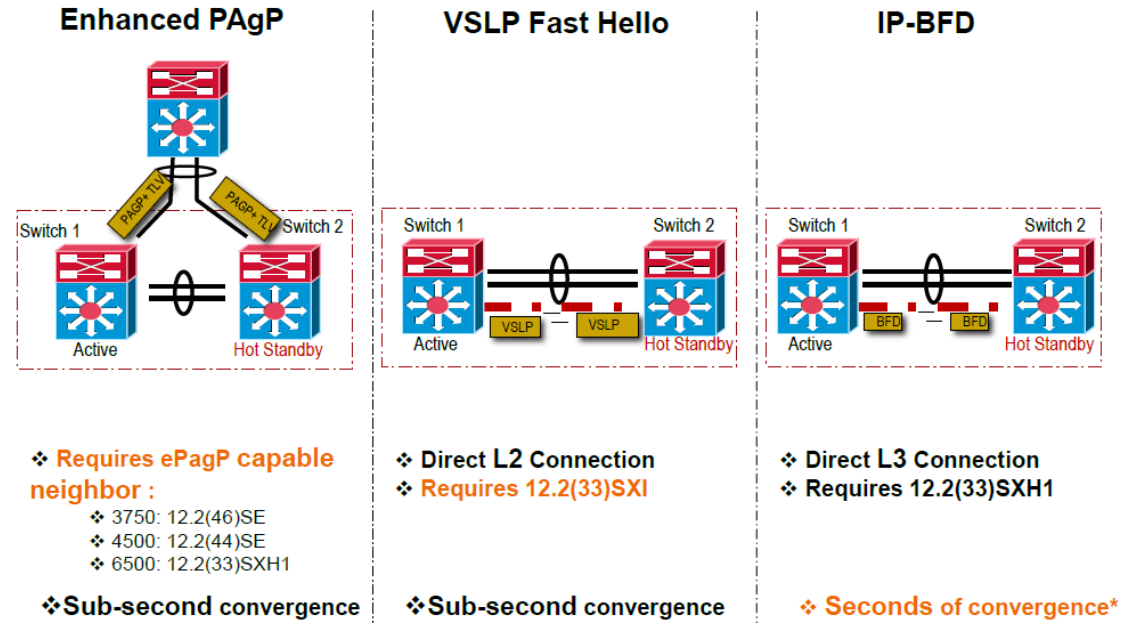
先说下VSS/IRF，这两个技术基本上没有啥差别。VSS（Virtual Switching System）是只在Cisco 6500系列交换机上实现的私有技术，IRF（Intelligent Resilient Framework）是在H3C所有数据中心交换机中实现的私有技术。二者的关键技术点如下：

1、专用链路跑私有协议：VSS使用VSL（Virtual Switch Link），IRF使用IRF link来承载各自的控制平面私有交互协议VSLP和IRF。专用链路使用私有协议来初始化建立邻接、协商主备（描绘拓扑）、同步协议状态，同时会在虚拟化完成后，传输跨机框转发的数据流量。二者都推荐使用10GE链路捆绑来做专用链路，说明私有协议交互和跨框传输的流量会很大。

2、基于引擎的主备模式：二者的控制平面都是只有一块主控引擎做为虚拟交换机的主控制引擎，其他的引擎都是备份。所有的协议学习，表项同步等工作都是由这一块引擎独立完成。好在这些设备大都是分布式交换，数据转发的工作由交换板自己完成了，只要不是类似OSPF邻居太多，拓扑太大等应用情况，一块主控大部分也都能搞定了。注意Cisco 6500必须使用Supervisor720主控配合带转发芯片的接口板才能支持VSS。

3、跨设备链路聚合：前面说了网络虚拟化主要是应对二层多路径环境下防止环路，因此跨设备链路聚合就是必须的了。Cisco配合VSS的专用技术名词是MEC（Multichassis EtherChannel），IRF倒是没有啥专门的名词，其链路聚合也和单设备上配置没有区别。

4、双活检测处理：当VSL或IRF link故障后，组成虚拟化的两个物理设备由于配置完全相同会在网络中出现双活节点，对上下游设备造成IP网关混乱。因此VSS/IRF都设计了一些双活处理机制以应对专用链路故障。1）首先网络中如果有跨设备链路聚合时，VSS使用PAgP、IRF使用LACP扩展报文来互相检测通知；2）如果有富裕接口在虚拟化的两台物理设备间可以单独再拉根直连线路专门用做监控，VSS使用VSLP Fast Hello、IRF使用BFD机制进行检测通知；3）另外VSS还可以使用IP BFD通过互联的三层链路进行监控，IRF则支持使用免费ARP通过二层链路进行监控。上述几种方式都是监控报文传输的链路或者外层承载协议不同。当发现专用链路故障时，VSS/IRF操作结果目前都是会将处于备份状态的物理机框设备的所有接口全部关闭，直到专用链路恢复时再重新协商。需要注意这两种虚拟化技术在进行初始协商时都需要将角色为备份的机框设备进行重启才能完成虚拟化部署。下面以Cisco VSS的三种故障检测方式举例，IRF也差不多。



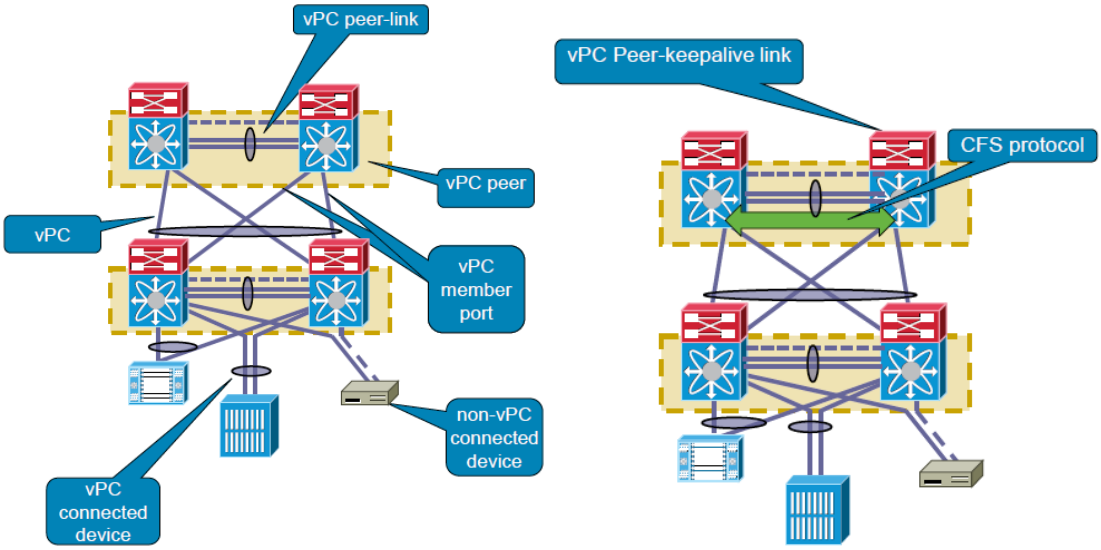
除了上述4个关键技术点外，VSS/IRF还有一些小的相似技术设定，如Domain的设定、版本一致性检查、三层虚接口MAC协商等等，都是基于方方面面的细节需求来的。由于应用环境相似，因此实现的东西也区别不大。想想RFC中的OSPF和BGP到现在都还在不断的推出新的补充标准和Draft来查漏补缺，就知道细节的重要性了。

VSS特色一些的地方是结合了Cisco 6500的NSF（None Stop Forwarding）和SSO进行了主控板故障冗余和版本升级方面的可靠性增强。而IRF则是将虚拟化延伸到了H3C接入层设备5800系列盒式交换机上（最大支持8或9台物理设备虚拟化为一台逻辑设备），可以打造逐层虚拟化的数据中心网络。

VSS和IRF都是当前较为成熟的虚拟化技术，其优点是可以简化组网，便捷管理，缺点则是扩展性有限，大量的协议状态同步工作消耗系统资源，而且纯主备的工作方式也导致了主控引擎的资源浪费。

vPC

Cisco在其新一代的数据中心交换机Nexus7000和5000系列中摒弃掉VSS，推出了vPC（virtual Port-Channel）特性。简单一些理解，VSS/IRF是整机级别的虚拟化，vPC是接口级别的虚拟化，而且从名称就可以看出是只支持链路聚合的虚拟化技术。从下图的vPC结构上就能看出，vPC和VSS从技术体系构成上其实没有大的区别。



其中Peer-Link对应VSL，Peer-Keepalive Link对应前面做双活检测的VSLP Fast Hello链路，CFS Protocol对应VSLP。在vPC中只需要对成员接口进行链路聚合相关的信息同步即可，不需要对整机的所有协议进行状态同步，大大减少了资源消耗和交互协议的复杂度。但是因为其控制平面机制与VSS一样要协商出主备角色，并由唯一的Master来管理vPC接口组，因此在扩展性方面同样被限制得很死，无法大规模进行部署，所以与VSS/IRF一同被归类为控制平面多虚一技术。

另外vPC由于其只关注了二层链路聚合，因此在组网设计上无法离开HSRP/VRRP等多网关冗余协议或OSPF等多路径路由协议，需要在路由层面独立部署。从Cisco放出的vPC技术胶片就能看到，画了一堆组网限制和注意事项，部署起来比VSS/IRF要更加复杂。

另外Arista的MLAG（Multi-Chassis Link Aggregation）技术和vPC很相似，学习时可以借鉴参考。

小结

控制平面多虚一技术总的来说比较成熟，VSS/IRF都已经有了不少商用案例，vPC虽然协议简单，但由于配合L3部署起来太复杂，应用案例还不是很多。此类技术最大的缺点就是受主控引擎性能影响导致部署规模受限，当前规模最大的是H3C 12518两台物理交换机进行IRF虚拟化后，用一块主控板管理支撑36块接口板处理数据，而VSS最多的是Cisco 6513两台交换机虚拟化后共管理22块接口板。目前VSS/IRF/vPC在市场上宣传部署的都是只支持两台机框进行虚拟化，还没有见到谁家的设备放出来支持4台机框虚拟化的版本做商用推广。估计得等到能够支撑上百块接口板的Super Supervisor出现，或者更完善的算法以提供多主控

负载均担，才能打破当下控制平面的多虚一规模限制了。

从Cisco Nexus系列产品的技术发展来看，在网络虚拟化的路线上Cisco已经开始偏向于数据平面虚拟化的TRILL等新兴网络技术，VSS/vPC等技术受主控引擎性能影响的部署规模局限性和协议私有化特征是制约其发展的硬伤，终将逐渐淡出大家的视线。

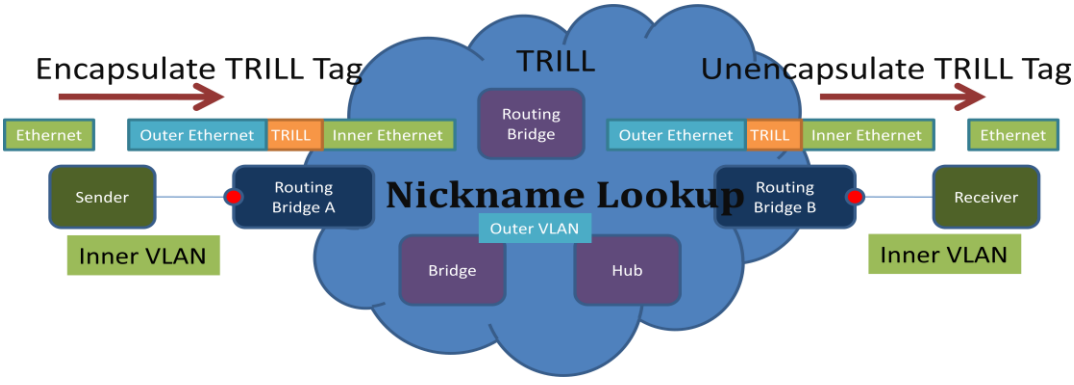
5.6.2 数据平面多虚一技术

数据平面多虚一技术的统一特征就是在二层Ethernet报文外面再封装一层标识用于寻址转发，这样基于外层标识就可以做些多路径负载均衡和环路避免等处理工作了。TRILL/SPB都是属于此列，QFabric目前开放出来的资料较少，猜测其应该也使用了类似MACinMAC之类的方式在其网络内部传输报文。

TRILL

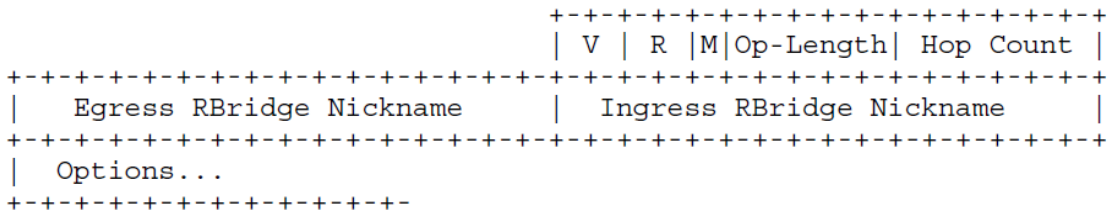
先说TRILL（TRansparent Interconnect of Lots of Links）和FabricPath。2010年3月时TRILL已经提交了IETF RFC 5556规范（Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement），此RFC只是描述了TRILL要解决的问题以及应用范围，定义协议细节的文档目前都还处于Draft阶段，形成完整的协议标准体系应该还得1-2年。

TRILL并不是专门为数据中心开发的技术，其定义的是在大型Ethernet网络中解决多路径问题的方案。FabricPath是Cisco在TRILL标准之上加入了很多私货的专门为数据中心而设计的一个超集，基本的控制平面与数据平面二者没有明显区别。



控制平面上TRILL引入了L2 ISIS做为寻址协议，运行在所有的TRILL RB（Routing Bridge）之间，部署于一个可自定义的独立协议VLAN内，做的还是建立邻接、绘制拓扑和传递Tag那几件事。数据平面在内外层Ethernet报头之间引入了TRILL报头，使用NickName作为转发标识，用于报文在TRILL网络中的寻址转发（可理解为类似IP地址在IP网络里面转

发时的作用）。每个RB都具有唯一的Nickname，同时维护其他RB的TRILL公共区域MAC地址、Nickname和私有区域内部MAC地址的对应关系表。因为TRILL封装是MACinMAC方式，因此在TRILL公共区域数据报文可以经过传统Bridge和Hub依靠外部Ethernet报头转发。TRILL报头格式如下图所示：



V (Version): 2 bit，当前Draft定义为0。

R (Reserved): 2 bits，预留。

M (Multi-destination): 1 bit，0为已知单播，1为未知单播/组播/广播，此时Egress RBridge Nickname意味着当前转发使用多播树的根。

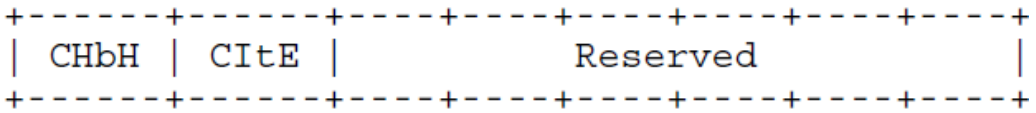
Op-Length (Options Length): 5 bit，Option字段长度。

Hop Count: 6 bit，最大跳数，逐跳减一，为0丢弃，防止环路风暴。

Egress RBridge Nickname: 16 bit，已知单播标示目的私网MAC对应的RB，多播则标示多播树根RB。中间传输RB节点不能改变此字段值。

Ingress RBridge Nickname: 16 bit，标示报文进入TRILL区域的初始边缘RB，中间传输RB节点不能改变此字段值。

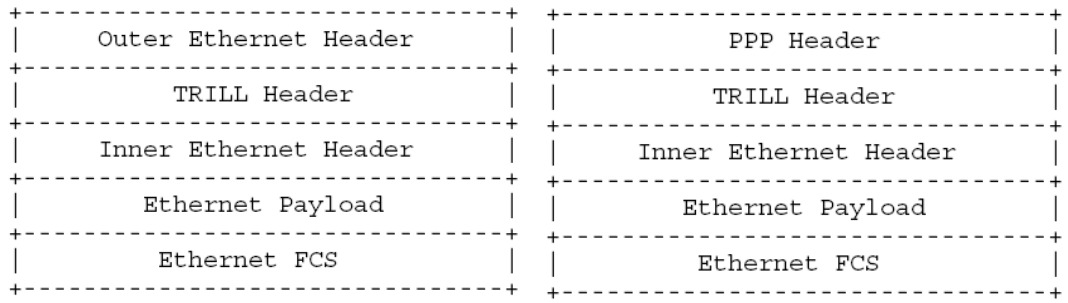
Options:目前只定义了CHbH (Critical Hop by Hop)和CItE (Critical Ingress to Egress)两个1bit的标志位，用于说明后面的Option预留内容是需要逐跳设备识别处理的或是首末端设备必须识别处理的。至于真正的Option目前都还没有定义。下图为Option字段内容：



普通Ethernet报文在首次从TRILL边缘RB设备进入TRILL区域时，作为未知单播还是依照传统以太网传播方式，广播给所有其他的RB节点。但是除了边缘RB外，TRILL区域中间的RB和传统Bridge都不会学习此数据报文中私有区域内部MAC地址信息，有效的降低了中间设备的MAC地址表压力。为了防止环路同时做到多路径负载均衡，TRILL的每个RB在初始建立邻接绘制拓扑时，都会构造出多个多播树，分别以不同的Nickname为根，将不同的未知单播/组播/广播流量Hash到不同的树，分发给其他所有RB。由于全网拓扑唯一且构造树

时采用的算法一致，可保证全网RB的组播/广播树一致。在RB发送报文时，通过将报文TRILL头中的M标志位置1来标识此报文为多播，并填充树根Nickname到目的Nickname字段，来确保沿途所有RB采用同一颗树进行广播。组播与广播报文的转发方式与未知单播相同。已知单播报文再发送的时候，会根据目的RB的Nickname进行寻路，如果RB间存在多条路径时，会逐流进行Hash发送，以确保多路径负载分担。

另外TRILL除了支持外层Ethernet封装在传统以太网中传输外，还规定了一种外层PPP封装方式可以跨广域网技术传输。以下是两种典型的TRILL报文封装方式：



TRILL的主要技术结构就是上面这些了，对更细节内容感兴趣的同学可以自行去IETF翻翻相关Draft。目前各个芯片厂商都已经进入TRILL Ready的阶段，只要技术标准完善发布并被广泛客户所接受，相关产品商用是So快的。

FabricPath

FabricPath是Cisco 2010年6月底正式发布的专门针对数据中心设计的私有技术，以前也叫做L2MP/E-TRILL，Cisco资料上的原话是：

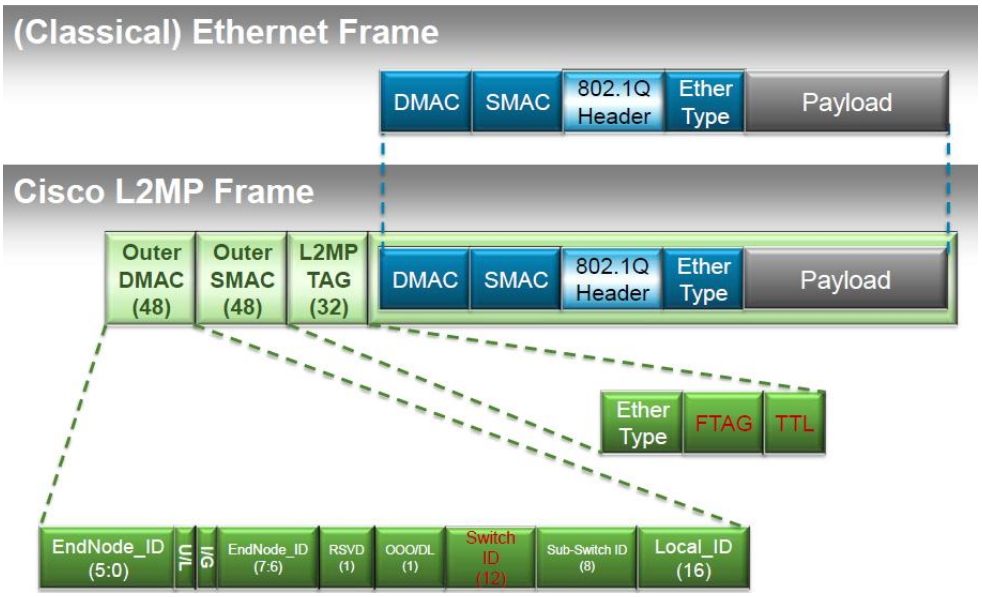
The Cisco engineers who developed L2MP only pushed part of it to IETF TRILL.

Functionality-wise, L2MP is a superset of TRILL

L2MP = TRILL + Cisco extensions

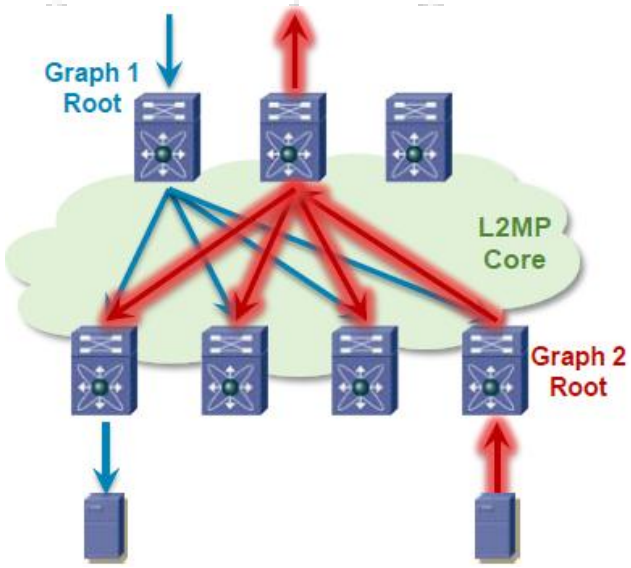
控制平面和转发规则上FabricPath和TRILL没有大的区别，一些主要变化如下，也可以说这些是Cisco专门针对数据中心网络对TRILL做出的变更：

1、FabricPath只支持RB间的点到点直连，不能加入传统Bridge等设备。因此FabricPath的报文格式相比较TRILL就更加简化，不再需要依靠外层的目的MAC进行Ethernet转发，数据报文封装有较大不同，FabricPath的报文格式如下图所示。



其中的Switch ID就是TRILL里面的Nickname。TTL字段用于避免环路风暴

2、采用FTAG（Forwarding TAG）标示不同的多播树Graph，用于多拓扑中未知单播/组播/广播报文的转发。多拓扑指可以在同一套FabricPath网络中支持不同的拓扑转发，而目前TRILL的多拓扑还未明确定义。每套拓扑缺省使用2个Graph，每个Graph可以用一个FTAG标示。目前NOS发布版本只开放支持2棵树Graph，既一套拓扑，据称最多可扩展至64棵树。多拓扑结构如下图所示。



3、MAC基于会话进行学习。当FabricPath的边缘设备从FabricPath区域中收到报文进行MAC地址学习时，会进行目的MAC地址检查，只有目的MAC在本地私有区域存在的，才会学习报文的源MAC到地址表中，这样可以避免MAC的不必要扩散。TRILL中RB设备还是传

统的Ethernet方式，收到报文就会学习源MAC，不做判断。

4、FabricPath支持基于vPC、FHRP等Cisco私有协议的组合应用扩展。

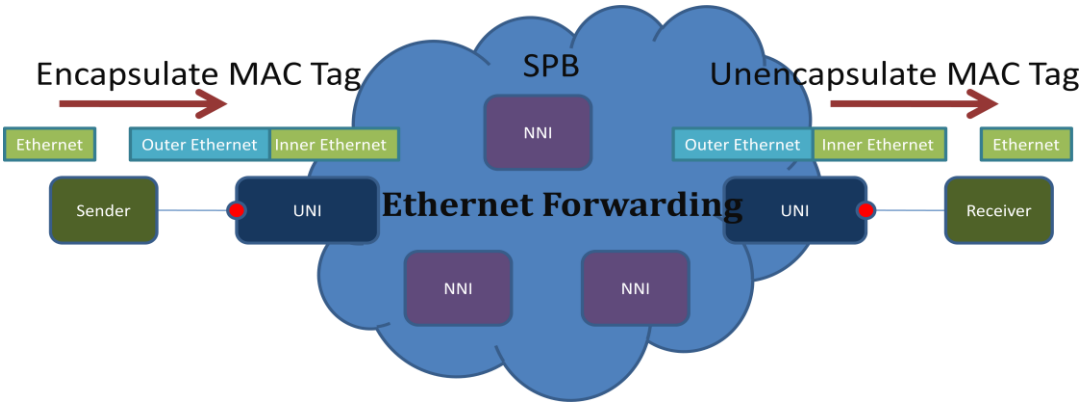
在当前TRILL还未有完整明确的标准出台情况下，Cisco已经用FabricPath走在了所有人前面，可以支持云计算大规模节点二层通信的数据中心建设，当然其主要的被攻击点就是私有协议。另外在Cisco的发布资料中也指出，其产品均已进入TRILL Ready状态，以后只需要命令变更就可以切换设备分别运行于纯粹的TRILL和扩展的FabricPath模式下。

SPB

要说SPB需要先谈谈PBB。PBB（Provider Backbone Bridging）是IEEE于2008年完成的802.1ah标准，为运营商城域以太网定义了一整套MACinMAC的转发机制。但PBB只定义了转发平面的封装内容，当报文封装上外层Ethernet报头在运营商骨干区域二层网络中时，仍然需要依靠传统的STP进行环路避免和转发控制。于是IEEE在2009年又定义了802.1Qay PBB-TE（Provider Backbone Bridge Traffic Engineering），用于在运营商的骨干区域中进行拓扑管理与环路保护，说白了就是通过手工方式配置一堆指定路径取代STP的自动收敛。目前IEEE还有个相关的标准P802.1Qbf, PBB-TE infrastructure protection处于草案阶段，预计2011年发布。

PBB-TE静态规划转发路径，明显无法适用于大型二层网络扩展，于是IEEE再搞出个P802.1aq SPB（Shortest Path Bridging）来，当前也还处于草案阶段。从IEEE的资料上看SPB主要是为了解决STP阻塞链路浪费带宽的问题而研究出来的。从实现上来看，同样是采用了L2 ISIS作为其控制平面协议进行拓扑学习计算，用MACinMAC封装方式在SPB区域内部进行报文传输。和TRILL很像吧，好在IEEE和IETF都是开放的标准化组织，不存在专利之争，不然肯定要掐架了。

SPB可细分为SPBV（VLAN QinQ）和SPBM（MACinMAC）两个部分，目前看主要用到的是SPBM。



SPBM是标准的MACinMAC封装，在SPB区域中数据报文也都是依靠外层MAC做传统Ethernet转发。外层Ethernet报头中的源目的MAC就代表了SPB区域边缘的UNI设备，此设备MAC是由L2 ISIS在SPB区域中传递的。

由于在SPB网络中还是采用传统Ethernet进行转发，因此需要定义一系列的软件算法以保证多路径的广播无环和单播负载均衡。下面介绍几个主要的部分：

1、首先SPB定义了I-SID来区分多个拓扑，I-SID信息在数据报文中以BVID（外层Ethernet报头中的VLAN Tag）形式携带，这样可以解决不同业务多拓扑转发的问题。

2、每个SPB节点都会为每个I-SID计算三棵树：到达所有相关UNI节点的SPT（Shortest Path Tree）用于单播与组播报文的转发；ECT（Equal Cost Tree）以处理两个UNI间存在多条等价路径时负载均衡转发；自己为根的多播树MT（Multicast Tree）用于未知单播与广播报文转发。

3、任意两点间的Shortest Path一定是对称的；ECT的负载均衡是基于不同I-SID分担的；

总的来说，SPB和TRILL/FabricPath相比主要有以下不同：

	SPB	**TRILL**	**FabricPath**
多拓扑	支持	研究中	支持
外层封装	标准Ethernet	Ethernet+TRILL	Ethernet+L2MP
转发标识	目的MAC	Nickname	Switch ID
多播树	各自为根	全局统一	全局统一
多路径负载分担	ECT端到端分担	逐跳Hash	逐跳Hash
环路避免	RPFC（反向路径检测）	Hop Count/RPFC	TTL/RPFC
标准组织	IEEE	IETF	Cisco

转发芯片支持	现有芯片	新一代芯片	Cisco自有芯片

SPB目前的最大困扰是转发路径靠软件算法保障，尤其在多路径负载分担时，对CPU计算压力远远超过TRILL和FabricPath，因此实际转发效率令人存疑。而且SPB的出发点是运营商的城域以太网环境应用，是否能适用于数据中心网络还有待观察。当前802.1aq SPB已经进入到Draft4.0，对其细节有兴趣的同学可以去IEEE网站注册下载学习。

多说一句，SPB是纯软件的解决方案，不需要更新转发芯片去支持，因此只要其标准化后，任何厂家都可以很快推出支持的版本，包括Cisco。

QFabric

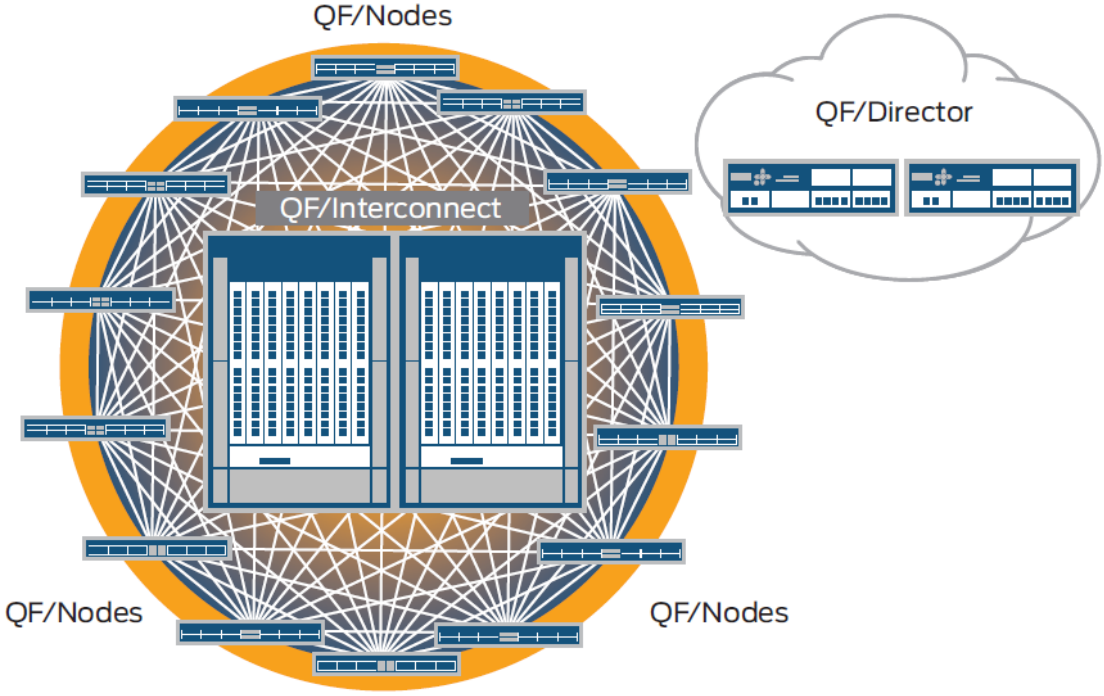
Juniper的QFabric也是目前喊得很大声的数据中心下一代网络技术，但由于还没有正式发布，开放的技术原理性文档基本没有，大都是些市场方面的资料。个人理解有以下几个要点：

1、首先控制平面一定是Juniper的私有协议，肯定要全J设备建设才成

2、由于J是自己没有芯片研发能力的，因此采购的基本只能是Broadcom/Marvel等几家通用芯片厂商的片子，再因此其转发肯定是基于MACinMAC的标准报文封装方式。

3、由1和2可以推断其实现方式应该是转发平面公共化，控制平面私有化。

4、从其如下的结构图中可以看出，对于虚拟后的逻辑交换机，可以理解Director、Interconnect和Nodes应该分别对应一台框式交换机的主控引擎、交换网板和接口板。



目前Juniper只发布了Nodes节点的QFX3500设备，等Interconnect和Director都出来估计怎么也得2012了。

小结

TRILL/FabricPath/SPB/QFabric都引入了控制平面协议来处理拓扑管理和转发路径判定的工作，都肯定会导致转发效率上，相比较传统Ethernet的下降，同时引入拓扑变化影响流量路径变更收敛速度的问题。但是这些技术毕竟比以前的STP在带宽上多了一倍的扩充，组网规模上也得到扩展，更适用于云计算数据中心的网络需求，总体来讲得大于失，属于更先进的生产力。

从开放标准上讲，个人倾向于IETF的TRILL。毕竟SPB出身不正，定位于运营商城域互联应用，而且就连IEEE都没有将其放入DCB（DataCenter Bridging）的大技术体系中。

从私有技术来看，VSS/IRF受到组网规模有限的硬伤限制，随着云计算网络规模的增大会逐渐退出大型数据中心的舞台，但用户服务器规模也不是说上就能上来的，至少还能有2、3年的赚头。而FabricPath/QFabric会较前者有更宽广的舞台和更长久的生存期，但由于其私有化的特征，当开放标准成熟铺开后，部署规模也只会日渐萎缩。可以想想OSPF和EIGRP的昨天和今天。

下面说些个人极度主观的预测（每写到这里都有些神棍的感觉）：

1、未来2-3年投入使用的云计算数据中心将是FabricPath/IRF/QFabric这些私有技术乱战的天下。在旧有开放技术不能满足使用，而新的标准仍未完善的情况下，私有技术成了人们唯一的选择。（VSS由于基于Cisco6500系列产品，受设备性能影响1-2年内会完全退出数据中心的历史舞台）

2、未来5年左右大型数据中心网络将会是TRILL一统天下，SPB半死不活，而各种私有技术也会延续之前的一部分市场，但很难得到进一步普及和推广。

最后完全依据个人喜好将上述技术做个排位，仅供参考，请勿拍砖。

FabricPath > TRILL > QFabric > IRF/VSS > SPB

5.6.3 控制平面一虚多技术

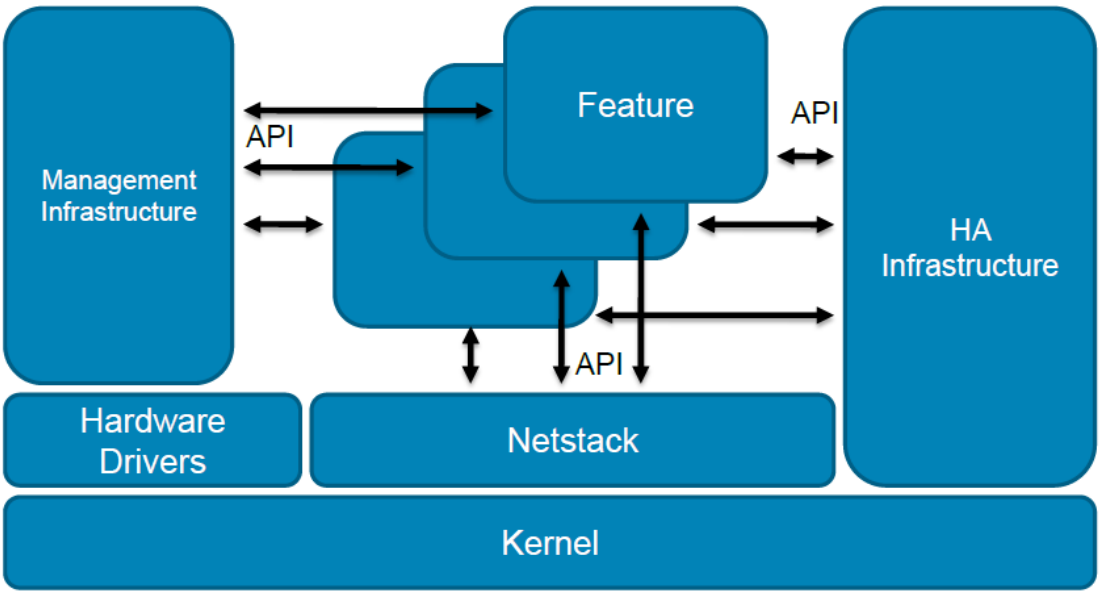
这个目前就是VDC了，没有看到任何厂家有类似的技术出来，另外由于此技术完全是本地有效的使用范围，也不会存在什么标准化的互通问题，就算有人去提个标准大家也不会

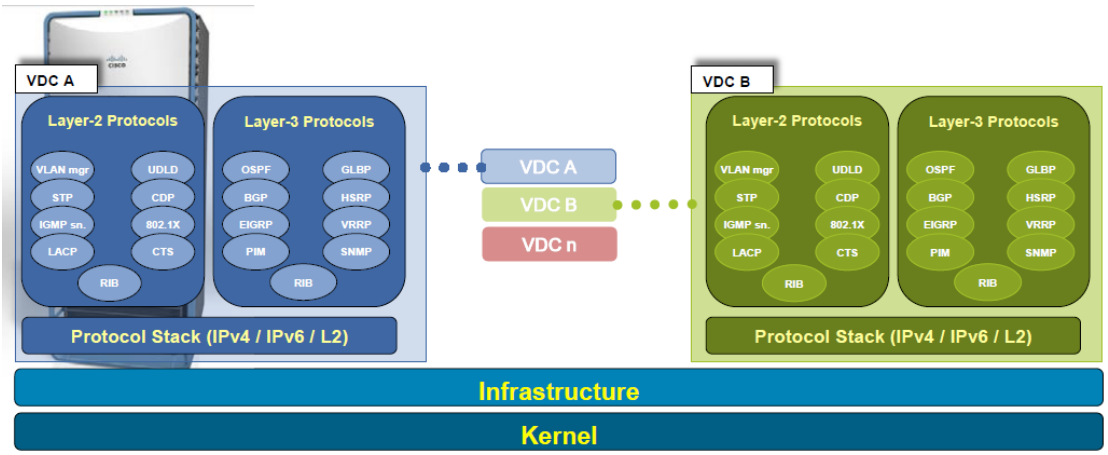
理睬的，可以参考VMware ESX/XEN/HyperV之间的关系。

VDC（Virtual Device Contexts）是Cisco基于操作系统级别的一虚多网络虚拟化技术。下面列表用于展示几项主要网络一虚多技术的区别。

技术	逻辑虚拟化对象
VLAN	数据平面
VRF	数据平面+少部分控制平面（路由协议）
虚拟防火墙	数据平面+管理平面
VDC	数据平面+控制平面+管理平面+系统资源

从下图的NXOS模型和VDC模型，可以看出VDC是建立在底层OS之上的，因此推测其采用的应该是前文中提到的OS-Level虚拟化技术。





现阶段Cisco公布的软件规格为每物理设备最多虚拟4个VDC，并支持每VDC 4k VLANs

和200 VRFs进行分层次虚拟化部署。按照云计算的需求来看，一般给每个用户分配的都是以带宽为度量的网络资源，不会以交换机为单位进行虚拟网络资源非配。即使是给的话，一台N7000只能虚拟4个VDC也不够用户分的。如果纯粹的流量隔离需求，最多使用到VRF也就够了，再多层次的虚拟化目前还看不清使用场景需求。

举个类似的例子，分层QoS一段时间以来也很火爆，但个人认为分2层也好，分10层也好，业务用不上都是白搭，当然在不考虑具体应用情况，纯粹拼指标时还是很有用的。又回到了前面的老话，正确的网络设计原则永远是自顶向下的。

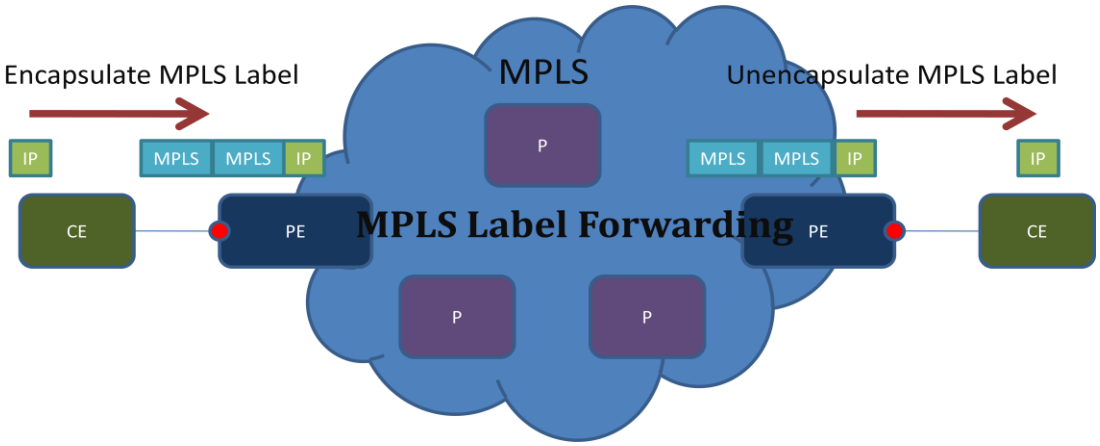
VDC可说的东西就这么多了，更多的就涉及到NXOS的核心设计了，Cisco也不太可能会向外公布。从Cisco的行业地位来看，未来的1-2年左右，其他各个设备厂商相应的私有技术都会随之跟进。也许实现手段和市场宣传各有春秋，但就技术使用和用户体验来说不会有多大差别。

再简单聊两句VLAN和VPN这两个熟透了的一虚多技术。VLAN好理解，就是在Ethernet报头中加个Tag字段，多了个层次区分，将Ethernet层面数据报文划分从一维转成二维，瞬间规模就大了N倍，随之也使Ethernet的复杂度大大增加。VPN就稍微复杂些了，谁让当初IP设计时候没有考虑预留其他Tag标识字段这块呢。

VPN（Virtual Private Network）分为本地有效的VRF（Virtual Routing and Forwarding，也有说是VPN Routing and Forwarding）和用于跨设备的MPLS VPN两个部分。VRF就是将本地的路由表、转发表和三层接口都给个VPN Tag，统一做IP层面的路由转发处理。例如从属于VPN A的接口进入设备的报文，只能查VPN A的路由表和转发表，从VPN A的其他接口转发出设备。当然后来又根据需求设计了一些跨VPN互访的技术，就不多说了。由于每个VPN都要维护自己的路由转发表，因此需要将各个路由协议RIP/OSPF/ISIS/BGP等都通过VPN标识隔离出多个数据库进程用于分开构造各自的路由表。这个也没啥难的，不管是数据报文还是协议报文都是基于接口出入设备的，因此只要将不同接口绑定到不同的VPN中，就可以做到IP路由层面的隔离了。而且这个VRF都是本地有效，各个厂家做的小有区别也不会相互影响。

跨设备转发时就麻烦了。首先得设计个Tag来让所有设备统一VPN Tag，于是有了MPLS Label；再就得让数据报文传输过程中带着这个Label四处游走，于是有了MPLS报头；还得让全体设备能够统一MPLS报头中Label对应本地VPN及IP路由的关系，于是有了LDP/BGP

VPNv4等专用和扩展协议用于传递Label。继续通过万能图解释VPN跨设备转发。



大的体系有了，还得补充细节。MPLS报头在IP报头外面，而且字段又少，只能多裹层头，分层标识不同PE设备和PE设备上的不同VPN（公网Label与私网Label）；规模大了，不同区域的Label无法互相识别，于是要想办法VPN跨域；三层IP报文搞定了又想在VPN里面传二层Ethernet报文，于是有了VLL/VPLS。。。

有兴趣的同学可以去IETF统计下VPN跨设备转发相关的RFC，个人是实在数不过来了，到今天都还有各种相关Draft不断地推陈出新，排队审核呢。

5.6.4 小结

数据中心内部的服务器互访技术介绍到这里就告一段落了，无论是前面的服务器跨接入层互访还是后面的跨核心层互访模型，都对网络虚拟化提出了严峻的要求。可以说在后面的云计算数据中心建设中，非虚拟化的网络将很快的被挤出市场舞台。未来10年的大型数据中心网络是属于网络虚拟化的，从技术层面，目前只能看到一个独领风骚的前行者。

PS：声明一下，作者不是唯Cisco派的，是唯技术派的。

5.7 数据中心跨站点二层网络

本章节重点技术名词：RPR/ VLL/VPLS/A-VPLS/GRE/L2TPv3/OTV

前面说了，数据中心跨站点二层网络的需求来源主要是集中云时的多站点服务器集群计算和分散云时的虚拟机vMotion迁移变更。搭建二层网络时，依据中间网络的承载方式不同，主要分为光纤直连、MPLS核心和IP核心三类。这里的多站点一般指3个及3个以上，只搞两个站点的云计算喊出去会比较掉价。

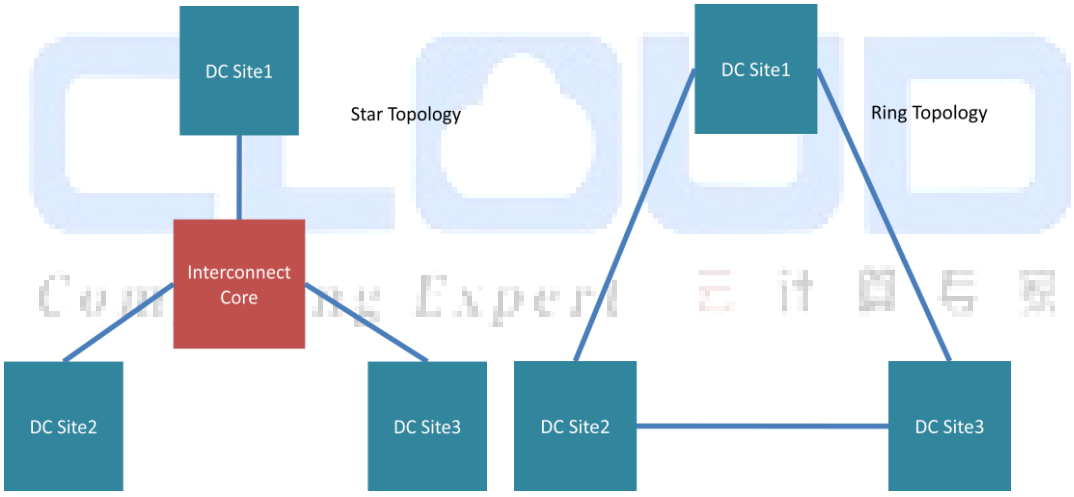
5.7.1 光纤直连

两个站点就不多说了，直接在两个站点的核心或汇聚设备之间拉两根光纤就OK了，也用不到什么特别的技术。唯一需要注意的是在两个站点之间的链路上做些报文控制，对广播和STP等报文限制一下发送速率和发送范围，避免一个站点的广播风暴或拓扑收敛影响到其他站点的转发。

当站点较多时，理论上有两种结构可用：

星形结构：专门找几台设备作为交换核心，所有站点都通过光纤直连到此组交换核心设备上，缺点是可靠性较低，核心挂掉就都连不通了，而且交换核心放置的位置也不易规划。这种结构不是值得推荐的模型。

环形结构：推荐模型，尤其在云计算这种多站点等同地位互联的大型数据中心组网下，环形结构既省设备省钱，又能提供故障保护，以后肯定会成为建设趋势。



从技术上讲星形拓扑不需要额外的二层互联技术，只部署一些报文过滤即可，可以通过链路捆绑增强站点到核心间链路故障保护和链路带宽扩展。而环形拓扑必须增加专门的协议用于防止环路风暴，同样可以部署链路捆绑以增加带宽冗余。

环形拓扑的公共标准控制协议主要是STP和RPR（Resilient Packet Ring IEEE802.17），STP的缺点前面说了很多，RPR更适合数据中心多站点连接的环形拓扑。另外很多厂商开发了私有协议用于环路拓扑的控制，如EAPS（Ethernet Automatic Protection Switching，IETF RFC 3619，Extreme Networks），RRPP（Rapid Ring Protection Protocol，H3C），MRP（Metro Ring Protocol，Foundry Networks），MMRP（Multi Mater Ring Protocol，Hitachi Cable），ERP（Ethernet Ring Protection，Siemens AG）等。

这里简单介绍一下RPR。从控制平面看，环路拓扑组网相对简单，控制协议交互规则制定也比较前面的TRILL/SPB更加简化，了解了全网各节点位置后，确定内外环两条通路即可。在数据平面上，RPR通过MACinMAC方式在环上封装外层节点MAC信息方式确认已知单播传递节点对象，非目标节点会将数据报文直接转给环上的下一跳，只有当目标节点收到此报文后根据外层目的MAC信息确认本地为终点，将报文下环转发。环上每个节点都会对未知单播/组播/广播报文着做下环复制和逐跳转发处理，直到转了一圈后，源节点再次收到此报文丢弃终止转发。

由于RPR在环路传输数据报文封装时增加了1个Byte的基本环控制和1个Byte的扩展环控制用于环路信息识别，因此也必须使用专用硬件处理环路接口的报文收发封装工作。RPR虽然很早就确立了标准内容，但由于其初始应用针对运营商城域以太网，且只能支持环路拓扑，因此各个厂商并没有花太大力气去开发产品进行支撑推广，当前使用不多。

就作者看来，未来几年的云计算数据中心建设时，除非在所有站点采用相同厂家的设备还有可能使用一些私有协议组环（可能性比较低），前文提到预测会以站点为单位选择不同厂家进行建设，这时就需要公共标准用于多站点互联了。在光纤直连方式下成熟技术中最好的选择就是RPR，但如果TRILL能够将多拓扑这块内容定义好，未来是能够将其取而代之的。

5.7.2 MPLS 核心网

一些大型的行业企业（如政府军工）自建内部网络时，会使用MPLS技术搭建各个地方的互联核心网。此时可以将各地的数据中心站点复用MPLS核心网进行跨地域连接，省钱才是王道。在自建的MPLS核心网中，需要在各个站点的PE设备间搭建VPLS隧道用于传输Ethernet报文。如果是租用运营商的VPLS隧道则不需要考虑这么多，那时PE是由运营商提供的，对用户来说组网部署和前面的光纤直连没有区别。

VLL

如果是只有两个站点互联的情况，可以使用VLL（Virtual Leased Line）。VLL是一种点到点的虚拟逻辑链路技术，数据报文从隧道入口入，只能从定义好的另外一端出口出，不存在多个隧道终点一说。数据平面没啥可说的，A点收到的二层报文进隧道直接封装上MPLS报头发给B点就OK了，整个过程框架可参考前面的MPLS转发图。控制平面由于隧道都是点到点连接方式，不需要复杂寻址，只要在数据流量传输时，给VPN分配外层封装的对应Label

即可。分配方式有以下四种：

CCC（Circuit Cross Connect）：全网静态为VPN分配一个Label，包括所有路径的PE和P设备都需要手工配置。此Draft已经处于Dead状态，目前基本也没人用了。

SVC（Static Virtual Circuit）：只在PE上静态配置私网VPN的Label，公网标签不管。有用的但也不多，静态配置这种方式对故障处理总是心有余而力不足的。

Martini：RFC4762，使用LDP协议在PE间建立连接，为VPN动态分配Label，省事好用。

Kompella：RFC4761，使用BGP协议在PE间建立连接，使用BGP VPNv4扩展字段携带VPN对应Label信息进行传递，这个实现起来比Martini复杂一点点，用得也就少了一些些。

后面两种Martini和Kompella方式在MPLS L3 VPN和VPLS里面也都有应用，都是作为控制协议来为VPN分配和传递Label用的。

VPLS

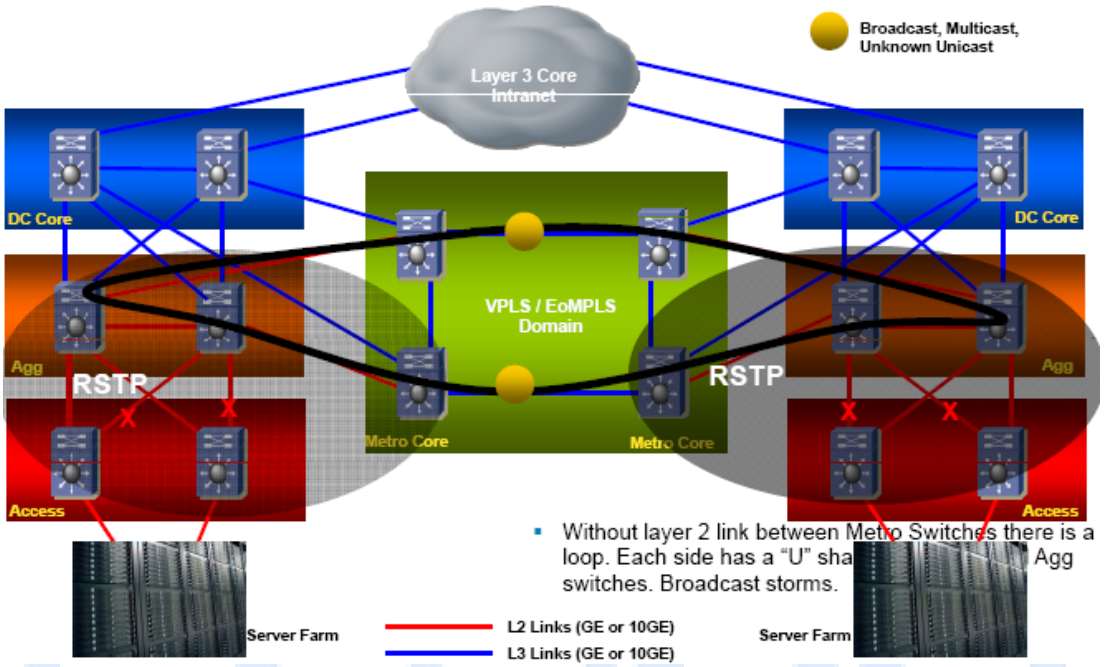
当存在多个站点时，A站点收到的二层报文就有个选B还是选C进行转发的问题。于是有了VPLS（Virtual Private Lan Service）。VPLS是支持点到多点的虚拟链路技术，从隧道入口进入后，可以根据VPLS MAC地址表从多个隧道出口中去选择正确的出口，或者广播给所有出口。控制平面还是通过Martini和Kompella两种方式分配与传递VPN对应的Label。数据平面则要多维护一张VPN的MAC对应VC（Virtual Circuit）转发表，既前面提到的VPLS MAC地址表，本地接口收到的报文，MAC地址学习方式还是和传统Ethernet一样；只有当报文从远端PE过来时，记录的源MAC需对应远端PE的VC ID。

由于VPLS透传的是二层Ethernet报文，就涉及到VLAN标识处理的问题。VPLS可以配合QinQ技术，将用户侧发来的带VLAN标签报文打上外层VLAN标签，以扩展VLAN数量规模。当然现在的交换机一般都是最大支持4k的VLAN，大部分场景都是够用的了，还没有听说谁家的数据中心VLAN部署超过4k。但云计算服务器节点数量规模成倍增加以后就不好说了，留出冗余总是好的。

为了防止广播风暴，VPLS做了水平分割特性，PE设备从远端PE收到的广播/未知单播报文只能发给本地的CE，不能转发给其他PE。还有其他的分层PE和Hub-Spoke等技术在数据中心多站点互联环境中一时还应用不上，这里也就不过多介绍了。

VPLS技术已经很完善了，喜欢细节的同学可以去查下RFC相关文档。这里再说下其在数据中心多站点互联应用中的不足之处。数据中心要求的是全冗余，无单点故障点或单点故

障链路，而VPLS在双PE冗余方面没有专门的定义，因此造成技术上的使用不便，一是会形成如下图所示的跨站点二层环路，二是本端CE无法感知对端CE-PE间链路状态情况，故障时导致流量黑洞问题。

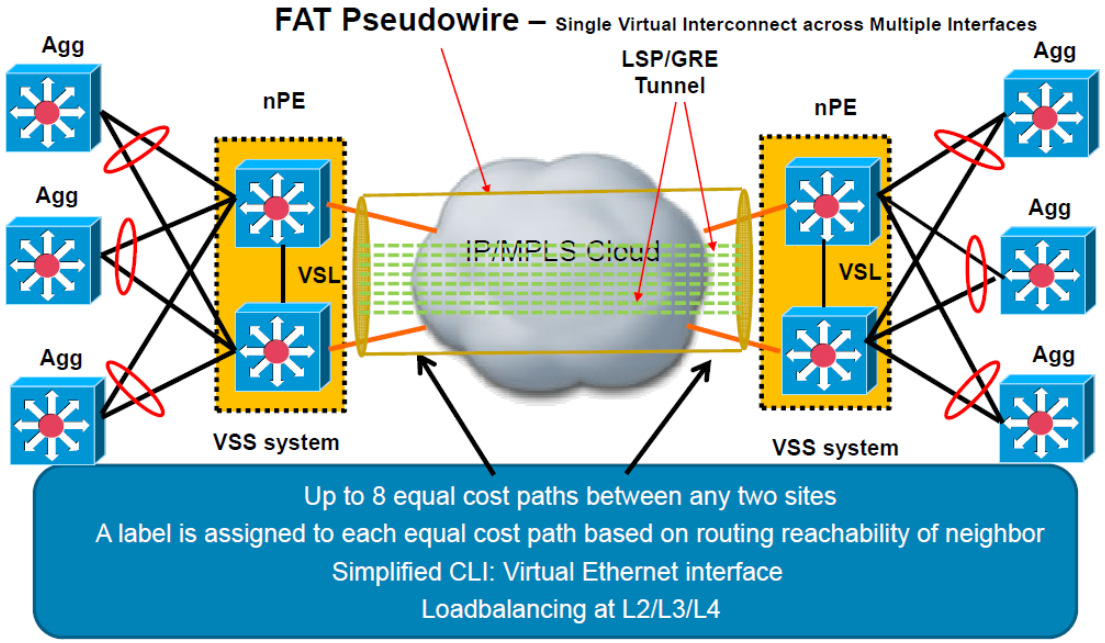


解决上述问题有以下两个思路：

首先是使用万能的STP构建出整网拓扑，即可避免环路，还可检测故障切换。缺点和前面在其他地方使用时一样，浪费带宽与收敛速度慢，另外就是要让STP跨站点组网，会导致一个站点出现问题，其他站点全部受影响。此方案的好处就是公共标准大家都能做，而且不存在互通问题。

其次是使用控制平面多虚一技术，如VSS/IRF和vPC，使多个物理节点变为唯一的逻辑节点将整个拓扑由环状变为链状，以避免环路。同时通过链路检测监控联动路径切换动作以避免流量黑洞问题，如Cisco的EEM。这些小技术组合起来可以解决上述问题，但缺点是都是私有技术，没有统一标准，无法支持不同厂商产品混合组网。

另外可以一提的是Cisco的私有技术A-VPLS（Advanced VPLS），此技术配合其VSS，可以将多条VPLS的PW（Pseudo Wire，可理解等同于VC）虚拟化为一条逻辑的Fat PW，达到多PW路径负载分担的效果，和链路聚合很类似。如下图所示。



此技术由于需要往MPLS报头中添加一个Flow Label的标签字段，用于处理多PW的流量路径Hash，因此别的厂家设备肯定无法识别，只能在全Cisco设备环境下部署。其他厂商也有开发出类似的技术，对多PW进行流量负载分担，但也都是私有的小特性，无法互通组网。估计再有1-2年IETF可以搞出个多PW负载分担的标准来，到时大家就好做了。

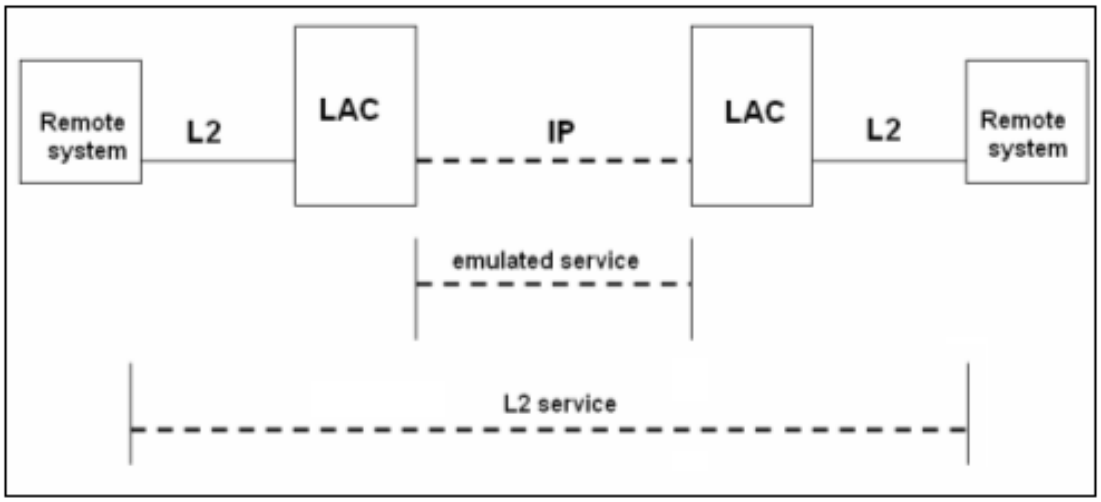
5.7.3 IP 核心网

全球最大的公共IP核心网就是Internet了，只要解决了报文加密的安全问题，且Internet出口带宽足够大，谁说以后的数据中心站点间二层互联不能走Internet呢。另外也有很多大企业的核心网采用IP建网，国内的如金融电力，国外则遍地开花。

从技术上来看，公共的技术标准主要有VLLoGRE/VPLSoGRE和L2TPv3，私有技术就是Cisco的OTV了。VLLoGRE/VPLSoGRE没啥好说的，就是在IP层打通个GRE隧道，再把Ethernet报文扔到隧道里面传。下面主要说说L2TPv3和OTV。

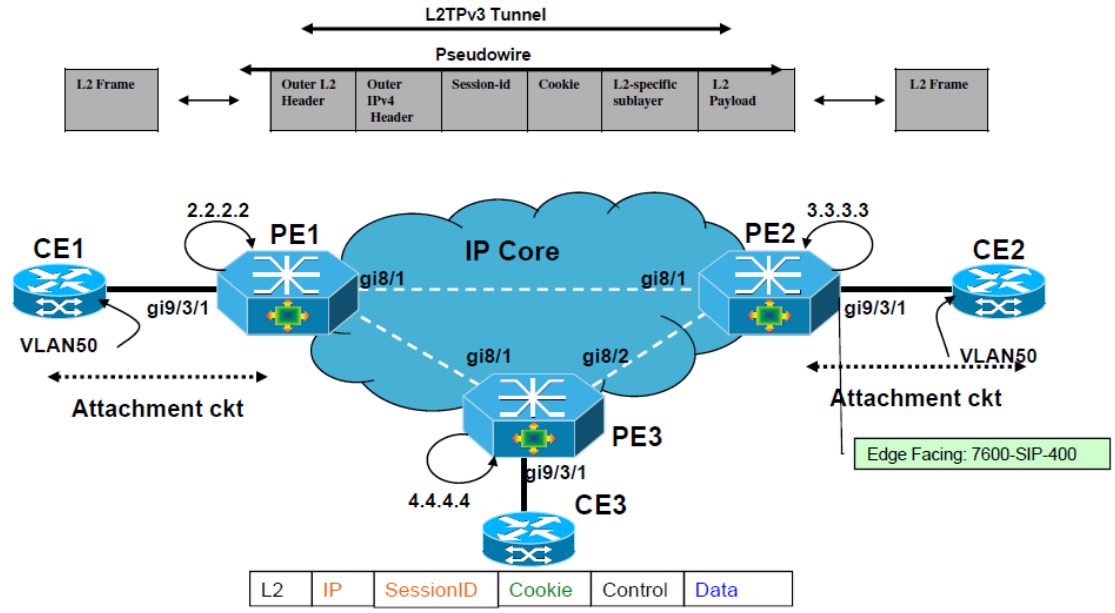
L2TPv3

L2TP（Layer 2 Tunneling Protocol）是IETF RFC2661（L2TPv2）定义的，已经有不少年头了，主要应用于移动办公通过Internet进行VPN接入等场景。L2TPv2的标准封装是基于PPP格式的，后来为了进行应用扩展，推出了L2TPv3，RFC3931。L2TPv3可以封装在IP/UDP层之上，摆脱了PPP的束缚。



LAC（L2TP Access Concentrator）是L2TP的角色名称，作为IP隧道的起终点，另外还有个LNS（L2TP Network Server）角色，但在数据中心多站点互联场景中应用不到，这里就不做介绍了。可以简单理解LAC等同于VPLS的PE。

从控制平面看，L2TPv3使用自己的控制协议报文建立IP隧道，由于隧道都是点到点的，也不存在什么拓扑学习的问题，这点和VLL相同。数据平面就是进隧道封包转发了，封装时只需在原始L2报文和外层IP头之间插入一个L2TPv3报头即可，里面包含SessionID和Cookie字段，其中SessionID字段32Bit ，Cookie字段可选，最长64Bit，整个L2TPv3报头最大12Byte，要大于前面那两个oGRE的了。整体结构可参考下面Cisco胶片的截图。但是注意里面虽然画了3个PE，但是实际上L2TPv3和VLL一样只能支持点到点的传输，如果CE3上也有VLAN50想和CE1/CE2一起组成二层网络L2TPv3是搞不定的。这个图有那么点儿混淆概念的意思。其中的Control和L2-specialfic sublayer字段在数据中心互联场景中就是指Ethernet报头，而在其他场景中也可以使用PPP等其他二层链路协议报头，毕竟L2TP定义的是要承载所有L2数据报文。

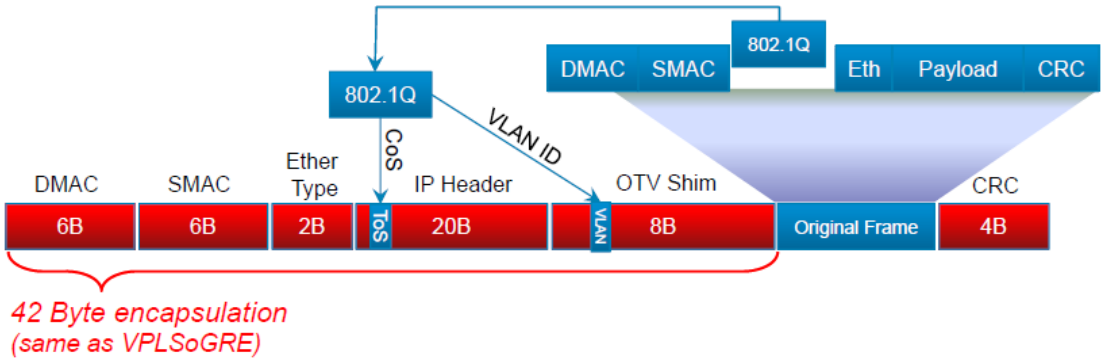


L2TPv3是RFC标准，谁家都能用，但只能像VLL一样支持两站点互通的场景，因此使用并不广泛，当前也就看到Cisco有在小范围推广。VPLSoGRE还是大部分厂商在多中心互联中主推的技术。顺便一提，L2TPv3只解决跨IP封装传输的问题，对于前面提到的多PE和流量黑洞的问题并没有对应方案，还是得配合VSS/IRF和探测处理等私有技术才能适用于数据中心。

OTV

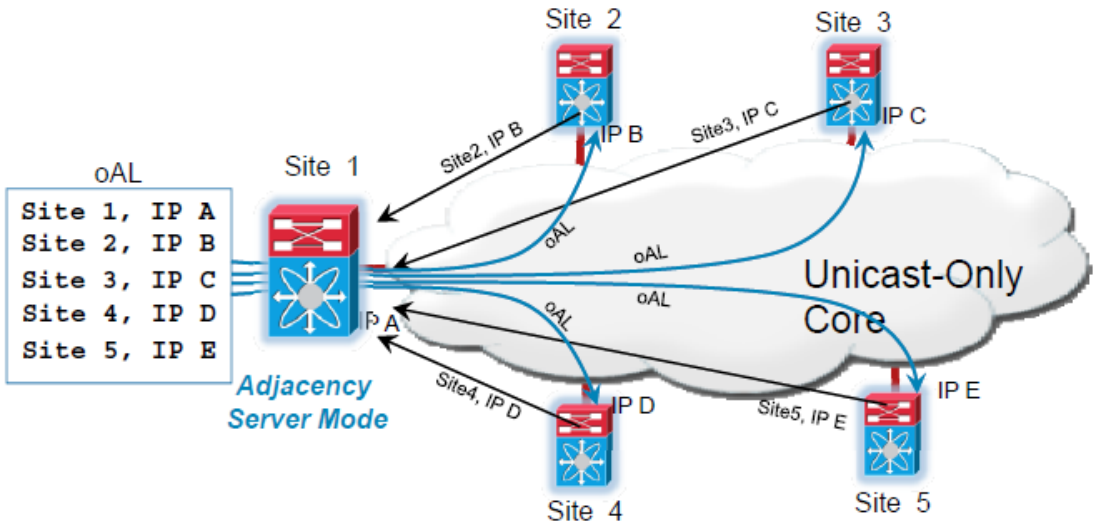
前面说了，在IP核心网情况下，公共标准只有使用VPLSoGRE才能支持多站点的二层互联，而VPLSoGRE同样存在前面VPLS组网中的多PE连接和流量黑洞问题，需要配合其他一系列的私有技术才能一并解决，部署起来相当繁琐，而且真出了问题定位也很困难。公共标准在这种场景下不是不能用，而是不好用，只能看私有技术了。

Cisco在其新一代数据中心交换机Nexus 7000中推出了OTV（Overlay Transport Virtualization）私有技术来专门处理数据中心多站点二层互联使用场景。数据平面OTV以MACinIP方式封装原始Ethernet报文，报文结构如下：



可以看到对比VPLSoGRE只是将8字节GRE头替换成OTV标识，长度没有变化。因此转发效率估计和VPLSoGRE相当。

控制平面上，OTV有组播与单播两种方式建立邻接拓扑，组播方式适用于支持组播的IP核心网，个人觉得还是很少见的。单播方式需要设置一台AS（Adjacency Server），保存所有的邻接设备信息列表oAL（overlay Adjacency List）。所有OTV节点需要手工设定此AS地址，上线时去取得其他邻居节点信息以建立邻接。当然还可以配置备份AS节点避免单点故障。OTV单播邻接建立方式如下截图所示。



针对数据中心多站点互联的场景，OTV设计了一系列的机制处理。

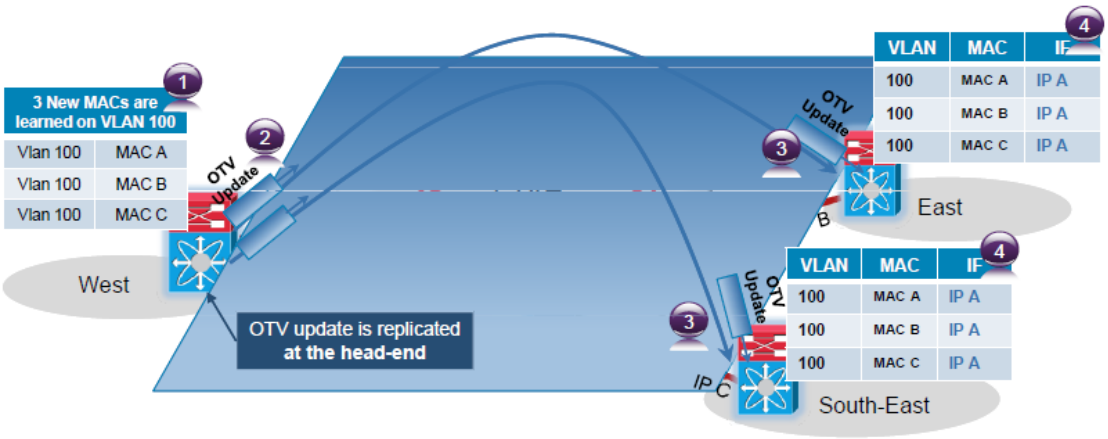
1、STP隔离。将STP BPDU报文在OTV边缘设备ED（Edge Device）上进行阻塞，禁止其跨站点传播。

2、未知单播隔离。将未知单播数据报文在ED上进行阻塞，禁止跨站点广播。同时可以手工配置静态MAC地址对应远端OTV接口的表项，以应对部分静默主机应用场景。

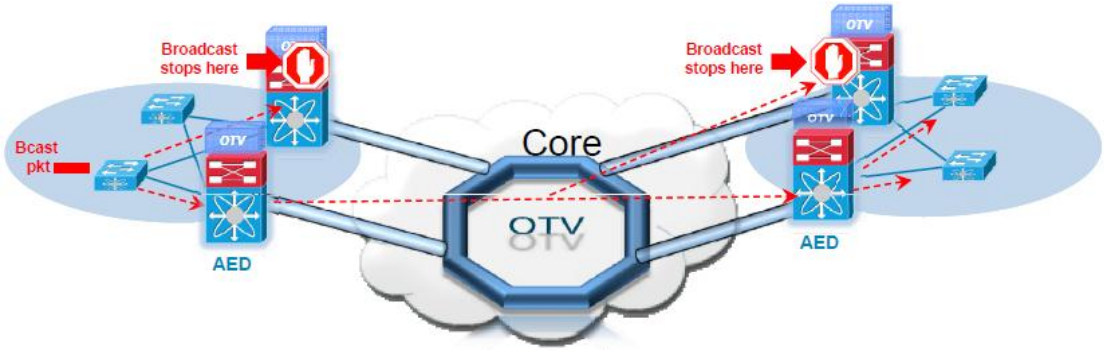
3、ARP控制。对远端站点返回的ARP Reply报文进行Snoop和Cache，当再收到本地查询同样目的IP的ARP Request时直接代答，不向其他OTV站点扩散，减少跨站点的ARP

Request广播。

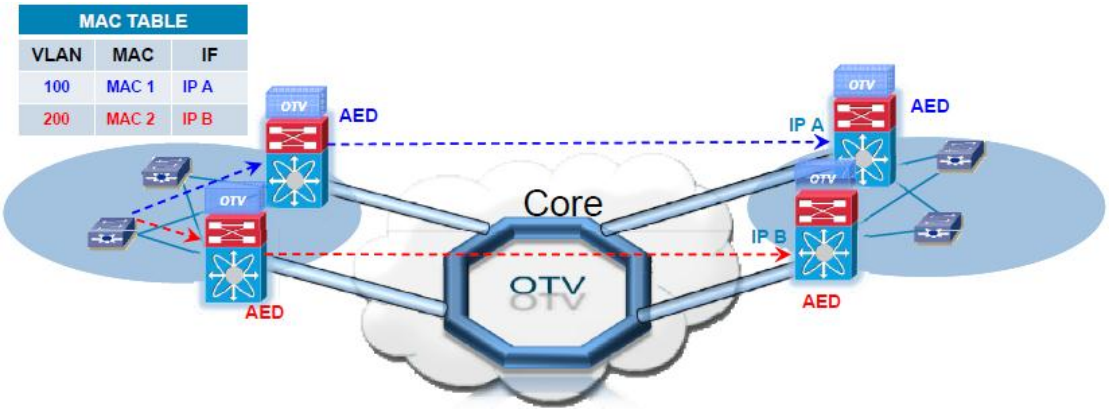
4、MAC地址学习控制。由于未知单播报文被隔离了，因此需要通过OTV协议报文进行站点间的MAC地址学习同步。过程如下图所示。跨站点的广播报文的MAC地址学习规则仍然与传统Ethernet相同，而且OTV不会对其做特殊控制。广播报文限速这种功能现在基本是个交换机就能支持了，算不上特色技术。



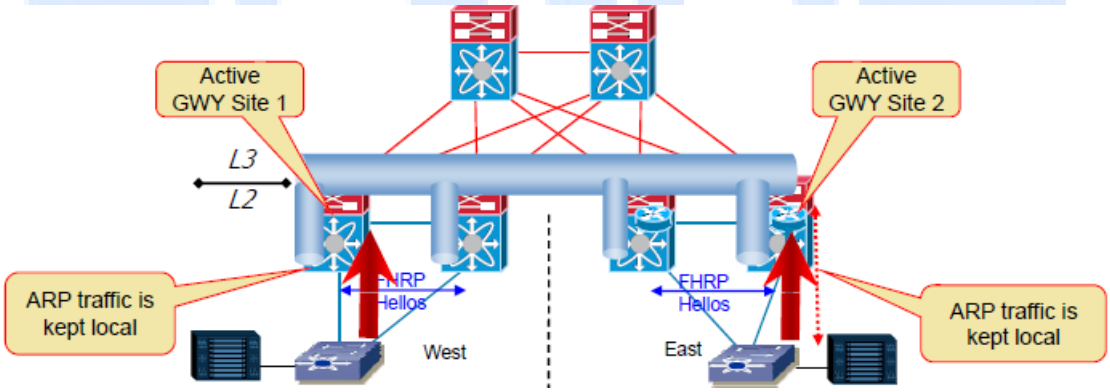
5、站点ED双机冗余。可以在一个站点使用多台ED接入OTV核心网，前提是要保证多ED在站点内部可以二层互通。运行控制协议进行AED（Authoritative Edge Device）设备的选举，只有AED可以转发和接收广播/组播报文，以避免环路风暴。如下截图所示。



另外可以在多个ED间基于VLAN进行不同的AED选举，达到全局意义上的流量转发负载均衡。如下截图所示。



6、HSRP隔离。不管二层技术再怎么搞，除非采用VSS/IRF等私有技术去做控制平面整体虚拟化，否则网关冗余在数据中心里面都是必不可少的。而当多个站点二层通信时，也必然存在网关部署位置的选择问题。如果站点A的主机每次和外界通信都走站点B的网关，则会导致大量的非必要流量途径站点间二层互联链路，浪费带宽且路途绕远。Cisco提出了更好的处理方式，OTV在ED上通过过滤HSRP HELLO报文，将HSRP进行站点间隔离，这样各个站点的网关相同，但各自为政，上下行流量路径最优。



OTV的主要内容差不多就这些了。从技术上来讲，其细节考虑最全面，而且瞄准的市场是所有站点间IP可达的应用场景，既OTV同样可以部署在光纤直连和MPLS核心网的场景中，远远领先于其他的数据中心多站点互联技术方案，但由于其私有协议的地位和可能的性能瓶颈，在市场上最终能占到多大地盘还有待观察。这里有两点猜测：

1、OTV如果今年仍在米国拿不下专利，估计明年就该去IETF提Draft了。米国的专利还是很严格的，这种继承性居多的技术审查通过不易。再想想国内的一些相关领域技术专利，都是神马的浮云。

2、未来的一两年间，其他各个瞄准数据中心的设备厂商类似（用仿字不好听）OTV的私有技术将会如雨后春笋般发布出来。

5.7.4 小结

数据中心跨站点二层互联目前还是个比较新的需求，实际上马的项目并不多，以前大多是满足存储需求的光纤直通。但大潮已经涨起来了，国内的各大运营商纷纷启动测试，国外估计也有一些项目应用。各个厂商最好尽快做好弄潮的准备，不然很快就会被淹没的。

从各个角度分析，个人认为上述三种互联方式中还是光纤直连最靠谱。

首先从省钱角度来看，建得起数据中心多站点的就不会差那几个钱拉不起光纤，如果使用CWDM/DWDM等波分复用设备可以在一对光纤上建多个通道供存储和不同业务共享使用，总带宽最高可以上T，性价比不见得比租用和复用MPLS/IP核心网络要低。

其次从重要性角度来说，需要跨站点运算的应用程序肯定是以企业关键业务为主，给关键业务拉根专线，不去和其他业务搅合，也是可靠性设计的必要需求。

最后从当前可使用技术的角度考虑，VLL/VPLS的数据报文查表封装处理工作，新一些的转发芯片可以搞定，但L2TPv3/GRE/OTV一般的转发芯片肯定是搞不定的，这样就需要额外的CPU或NP去处理数据报文，性能上自然也不会有什么太好的期待，万兆线速都是奢求。而且不管是怎么封装，传输的时候外面一堆报头都很损耗带宽，列举个极限的例子，OTV和VPLSoGRE外层报头要封42Byte，对64Byte载荷的数据小包，报头带宽损耗达到了42/(42+64)=40%，一条10G链路，跑满了才能用6G传数据，效率那是相当的低啊。

而光纤直连方式目前主要技术瓶颈在多站点互联时，缺少专业高效的公共标准技术，RPR如果想在数据中心有所作为，还需要进行一些协议上的标准改进，多考虑一些如未知单播/广播数据报文和STP/ARP/VRRP等协议报文的数据中心场景专门处理，以及站点多边缘设备冗余接入应用场景处理。另外各个厂商对相关产品的高调发布和大力市场推广也是必不可少的。个人觉得在数据中心多站点互联这块，私有技术早期也许能忽悠住一些用户，但最终肯定是站不住脚的。

5.8 数据中心多站点选择

本章节重点技术名词：SLB/GSLB/LISP

数据中心多站点建设时考虑应用服务冗余，则必然面临着以下问题：1）Client访问Server的时候选A站点还是B站点；2）A站点的Server故障或服务迁移到B站点后，Client访问如何能随之快速切换。

前面已经提到，在选路技术中主要有两个解决方案思路，一是DNS，二是路由。

DNS方案的缺点首先是应用扩展性不强。DNS协议以处理HTTP/HTTPS的应用为主，其他类型应用较少。不过话说回来，目前的应用中基于WEB的BS（Browser-Server）结构快打遍天下无敌手了，这个问题倒也影响不大。还有个问题就是DNS自身协议设计时，没有考虑多IP选择问题，所以此类解决方案都要和主机探测、迁移同步等私有技术配合起来使用，一般都得好几个盒子联动起来才行。DNS的好处是简单，使用的技术都是成熟技术，准备好接口写两行代码谁都能分上一杯羹，只要操心优化处理性能方面即可。目前的典型技术代表就是GSLB+SLB（可选）+vMotion通知（可选），下面会进行详细介绍。

路由的方案是完全基于IP技术出发，网络设备厂商自己就能搞定，不需要找DNS或vMotion去联动。其中的主备中心的路由掩码比明细和主机路由发布两种方案都不是啥好招，属于拆了东墙补西墙，会造成其他的问题。而稍微完整一点儿的LISP目前也还是试验探索阶段，vMotion后主动路由刷新这个最主要需求，也没能得到太好的解决。下文会简单介绍LISP，并探讨更优的处理方案。

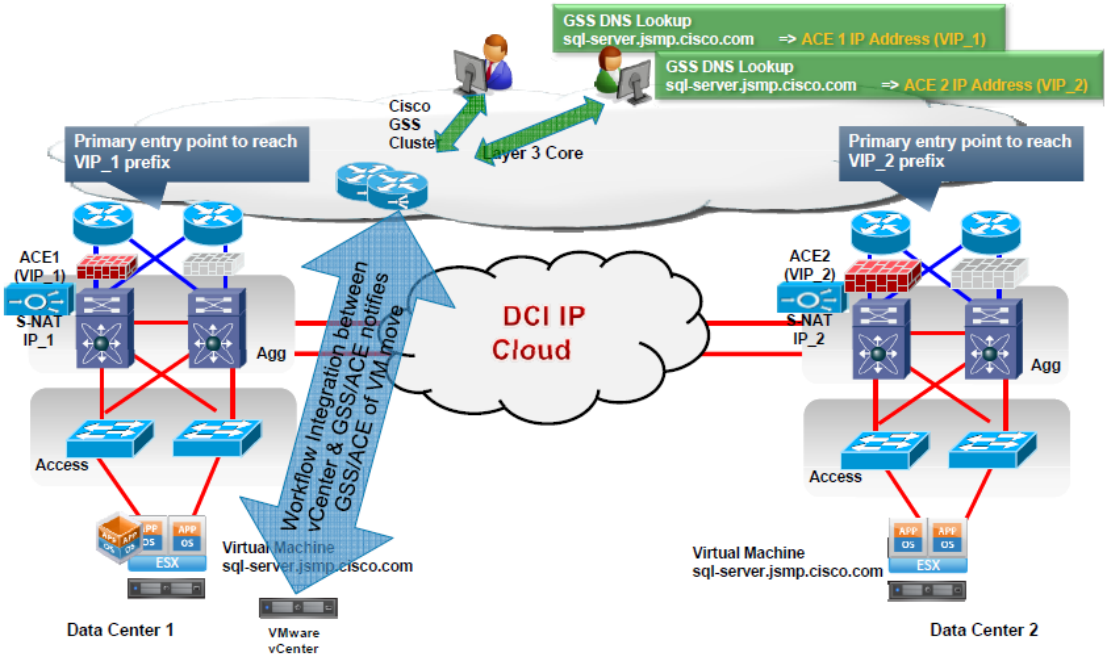
5.8.1 GSLB

GSLB（Global Server Load Balance）全局负载均衡技术，这个貌似是Redware先提出来的叫法，其他各个厂家叫法都有区别，如F5的3DNS和Cisco的GSS等。作者觉得这个词更贴切而且技术性普适一些，就按这个名词进行技术介绍，从原理上讲和3DNS/GSS什么的没有啥大的区别。

GSLB就是一台DNS解析服务器。最主要功能就是对不同Client发往相同域名服务的请求，以一定算法规则进行Hash，回应不同Server IP地址。常见的算法包括轮询（90%都用的）、最少连接数和服务器最快响应速度等。增强功能是可以对Server IP进行探测，如果探测到某台Server故障，则会使用其他正常的Server IP进行Client的DNS响应。常见的探测方式有ICMP（90%都用的）、TCP以及上层应用如FTP/HTTP等。当然也可以再搞些HTTP重定向等特性，将GSLB设备放在数据中心站点的入口便于故障快速切换。

在vMotion应用场景中，由于VM在迁移前后的IP地址不变，因此两个数据中心站点的VM对外提供服务的Server IP地址相同，GSLB此时就需要服务器前面的SLB（Server Load Balance）设备进行配合了。SLB是个NAT（Network Address Translation）服务器，主要作用

就是将后端真实服务器的IP和TCP/UDP Port等映射为对外提供服务的虚拟IP和TCP/UDP Port，然后将不同Client访问虚拟IP的流量修改目的IP后，分别发到后端的不同真实服务器上，以达到对后端多台真实服务器的流量负载均担效果。SLB同样需要对真实服务器进行探测，以及根据不同的算法规则将Client流量均匀Hash到不同的真实服务器上。使用不同的SLB可以将后端相同的真实服务器IP映射为对外的不同虚IP地址，此时配合GSLB就可以解决vMotion前后VM服务IP相同的迁移切换问题。另外如VMware的VM管理控制平台vCenter可以将vMotion动作通知GSLB设备，达到快速切换效果。下面以Cisco的技术结构截图举例，其中的GSS就是GSLB，ACE为SLB。其他厂家的方案在技术结构上也没有啥根本区别。



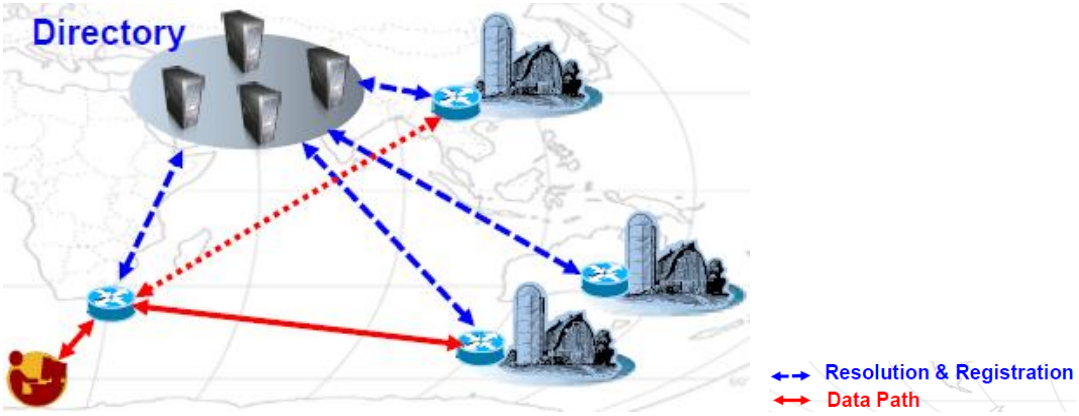
说实话个人觉得GSLB技术没有啥难度可言，DNS解析和调度算法都是现成的东东，稍微有点儿技术实力的厂商都能做个盒子出来。这个东西关键在于性能，由于DNS解析等上述功能行为都得靠CPU实现，没有啥公用芯片，那么就看谁家的算法实现效率高，谁家支持的新建并发规格大。如果性能规格差不多，都能满足需求，再要考虑的就是可靠性和性价比了。关于衡量数据中心性能和可靠性的问题，会在本文的相关外篇中再深入讨论。

5.8.2 LISP

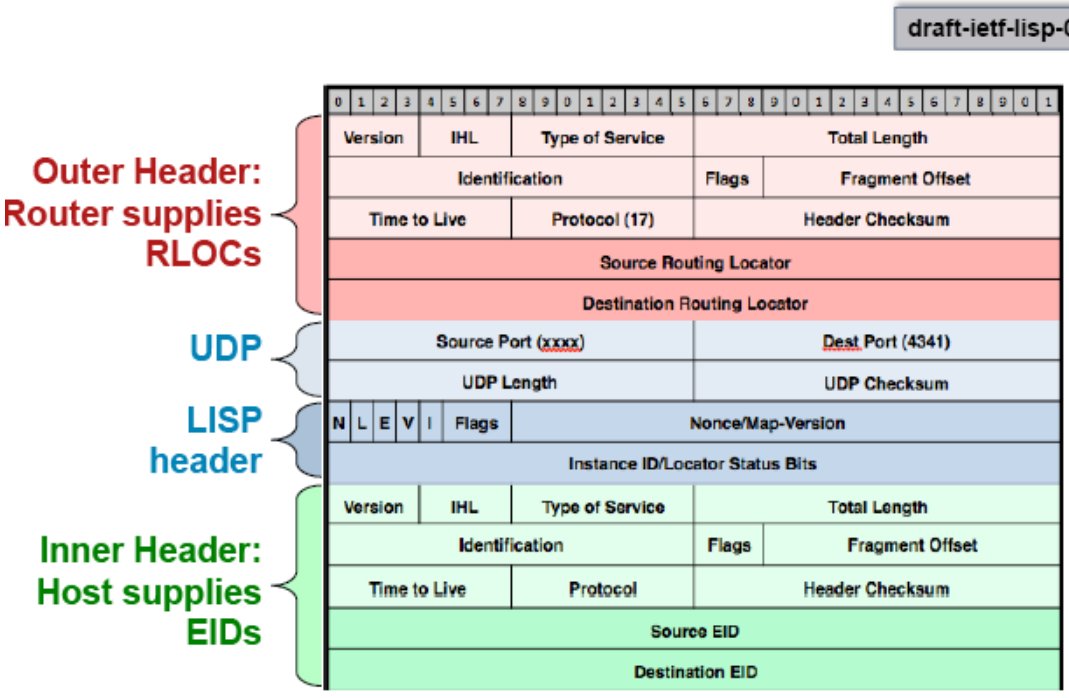
LISP（Locator/ID Separation Protocol）实质是个IPinIP的协议，其主要思想早在15年前就已经被人提出来进行研究，然而一直没有太具体的东西产出。直到2006年，Cisco重新开

始投入资源进行研究，目前已经提交了很多的IETF Draft，最新版是今年4月份的Draft Version12。但就应用来说。Cisco的LISP目前也只处于试验阶段，距离能够推广商用还有不短的时间，很多技术细节方面问题需要解决。

LISP的应用结构如下截图所示：



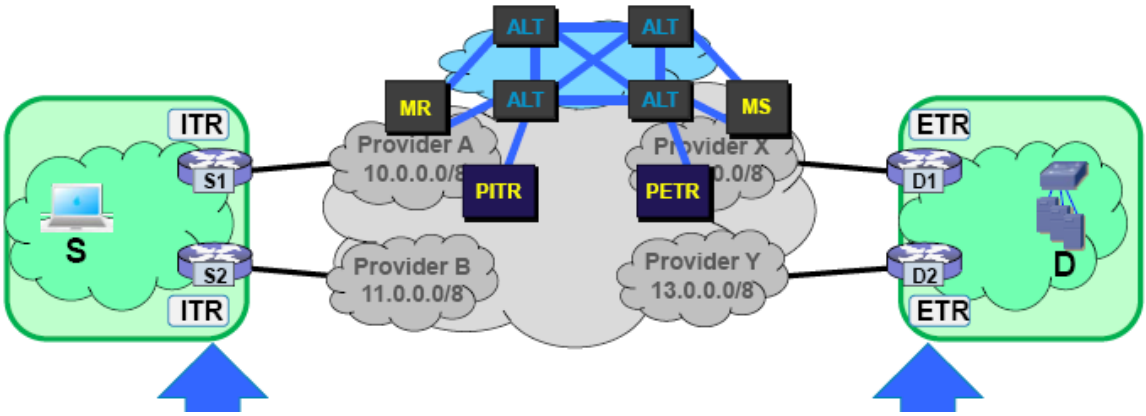
LISP提出将标识Locator的IP（RLOC）和标识目的节点ID的IP（EID）进行区分和叠加封装，在公网传输时只根据Locator IP转发，只有到达站点边缘时才会剥离外层IP，使用内层标识EID的IP进行转发。从下面的报文封装截图就可以看到其IPinIP的思想。



LISP有两个主要的目标：一是公网设备不需要学习站点内部明细IP路由项，可以有效减少公网的路由数目；二是当访问的目标服务在站点间迁移时，可以只变更Locator的外层IP，不需要对服务节点的内部IP地址进行变更，可以避免TCP等上层应用的中断重建，此点主要

是应用于数据中心vMotion和手机上网漫游的场景。

LISP的技术结构如下截图所示：



上图的名词很多，通过简单描述整个数据转发过程来帮助大家进行理解：

1、由源主机发往目的主机的数据报文第一次到达客户区域的LISP边界设备ITR。

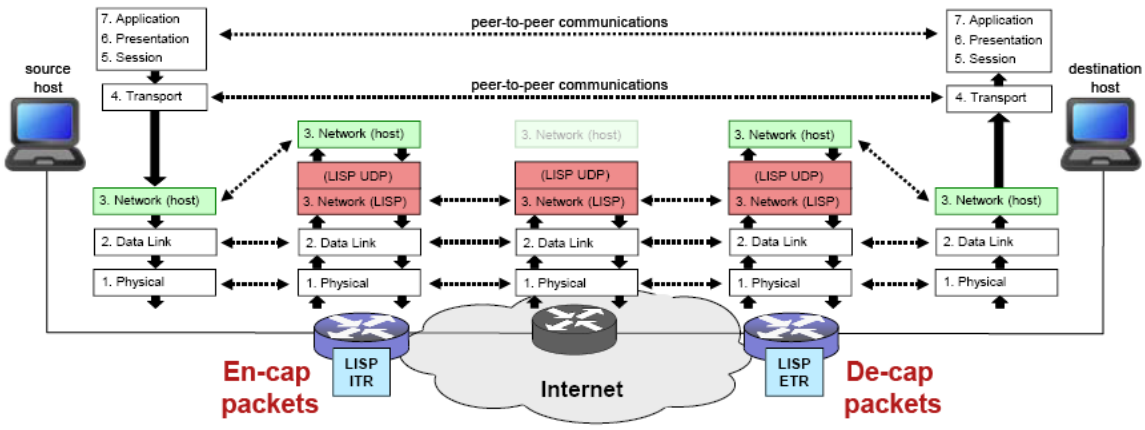
2、ITR会根据报文的目的IP地址EID，向Directory区域的本地查询服务器MR请求EID对应的目的站点边缘设备公网Locator IP。

3、MR会根据本地EID所属路由网段与MS的对应表项将查询请求提交给数据服务器MS。

4、MS上拥有EID对应目的站点边缘设备ETR的对应表项信息，MS会查表将此请求转发给ETR。

5、ETR会根据自身设置的规则（如优先级等）选择站点的某个公网IP作为Locator IP反馈会ITR。

6、ITR会根据ETR回应报文中的Locator IP封装外层报头将数据报文发到公网上，同时记录此EID与Locator IP的临时对应表项，当再有去往此EID的数据报文流经时直接封装转发。封装转发的过程如下截图所示，也有些眼熟吧。

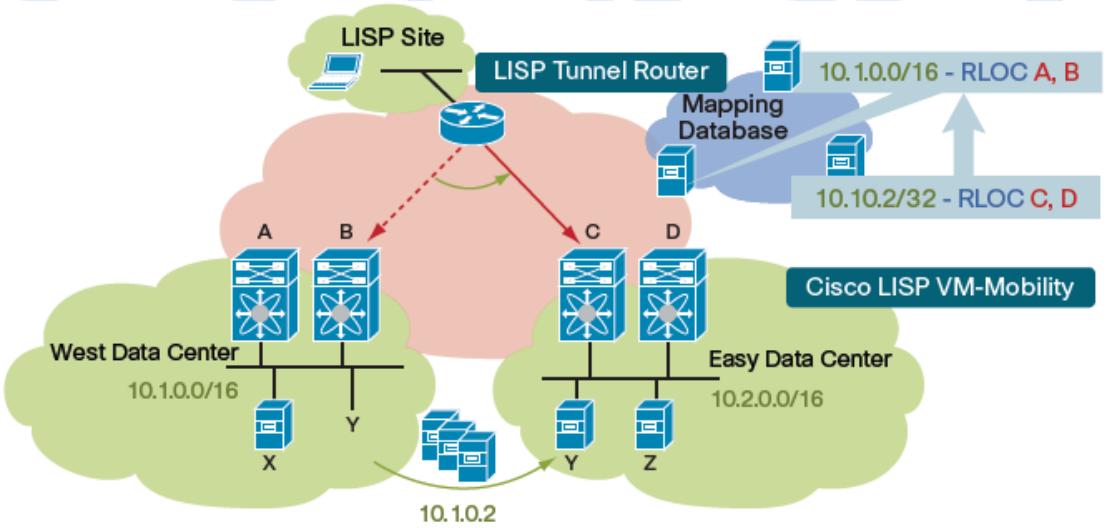


如果觉得1-5步的过程复杂不易理解，请回想一下DNS整套的域名解析过程，都是相通的。注意上述名称都是技术角色，一台设备可以实现多个LISP的角色功能，如同时实现ITR/ETR功能，或同时实现MR/MS功能等。

另外ALT是用于搭建MR和MS之间Directory区域互联用的中间角色，通过BGP扩展报文为MR和MS之间传递路由信息；PITR和PETR是用于LISP与不支持LISP的网络对接时做ITR和ETR代理用的角色。由于目前LISP也还没有定稿，此部分设备功能没有完整定义，有兴趣的同学请自行深入研究。

LISP中各角色之间大都通过手工指定的方式建立连接关系，如ITR上需要指定MR地址，ETR上需要指定MS地址，只有MR和MS之间可通过BGP来建立邻接关系并通过扩展报文传递EID表项信息，但目前实现出来的还是以手工指定方式为主。而且ETR上要将哪些EID信息发送到MS上，也同样需要通过配置网段掩码的方式手工进行指定。

LISP并不是专门为数据中心开发的技术，因而Cisco如果想将其在数据中心场景进行研究推广，估计会进行一些协议改造使其更加适用于数据中心的场景需求。目前Cisco给出的LISP数据中心实现vMotion过程如下截图所示：



上图是能找到的里面相对描述最清楚的了，但说实话感觉还是很糙。个人理解如果希望LISP应用于数据中心多站点选路，还需要解决以下一些技术问题：（下面这几段读起来可能会有些费劲，珍惜脑细胞的同学慎入）

1、迁移后服务器EID在新站点的ETR注册问题。既VM迁移后，新站点的ETR如何知道它此时需要向外发布对应EID。Cisco的当前做法是使用IP报文侦听，先配置个侦听范围，既可能会迁移过来的IP地址段，当监听到本地出现此地址段为源IP地址的IP报文时，会激活此

EID表项并发给MS进行注册。个人感觉侦听免费ARP会更方便一些，vMotion后VM肯定会发免费ARP报文的，但发IP报文就得看服务器的应用层协议设置了。不过此方案需要在站点间过滤免费ARP，不能使其跨站点传输，否则二层隧道会将免费ARP扩散到所有站点，侦听就没意义了，而过滤后会不会有其他问题还需细琢磨。这里只随口提个思路，什么方案都是有利就有弊的，需权衡清楚再实现。另外也可以让vCenter等管理平台去通知ETR，类似于前面的GSLB方案，这样由必须和VMware等虚拟机厂商做强联动，有些违背使用LISP的初衷。

2、迁移后通知ITR快速切换新的Locator IP问题。这个就更加复杂了，VM刚由站点A迁移到站点B，ITR不知道啊，还是在用旧的Locator IP封包发送，此时应用业务肯定就断了，直到ITR获取到新的Locator IP后才能再建立起连接恢复业务。如果是时间敏感型的业务，中断个几分钟，上下几百万就没了不是。当前LISP提了很多解决方向出来，但还没有什么确定的技术方案。个人思路如下，此问题需分解为两个小问题各自解决：

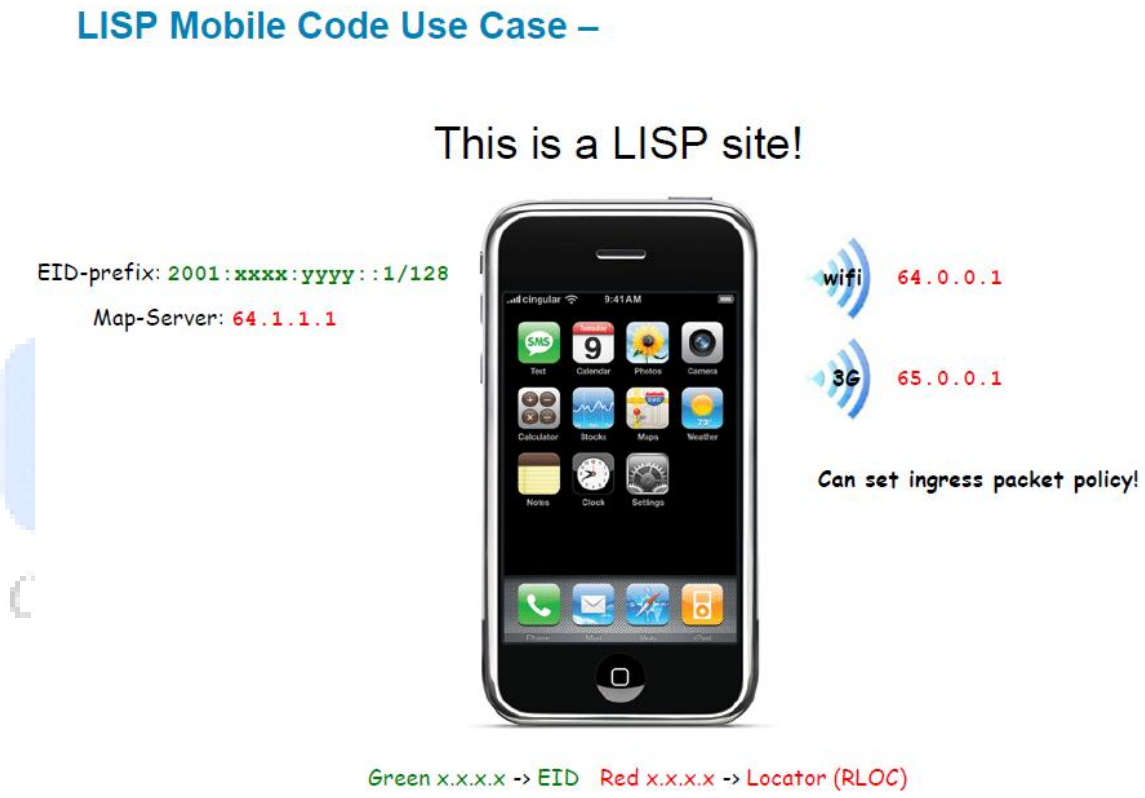
首先是要解决ITR如何感知迁移发生的问题：1）管理平台通知，需要联动，而且ITR那么多，也不会都注册到管理平台上去。不太靠谱。2）由ITR探测EID存活状态，探测范围太大，EID可能以主机路由居多，对ITR负担太重，可行但不是很好。3）由原始站点ETR探测EID状态，当迁移后探测到EID不在本地站点，则通知ITR删除临时表项。ETR上是保存了所有ITR表项的，可以很方便知道都要通知谁。但由于各站点间服务器前端网络三层互通，因此还要想办法将此探测报文在站点间隔离，否则迁移前后始终都是通的，和前面的ARP侦听方案存在相同的问题。4）在问题一已经解决的情况下，当服务器EID在新站点的ETR注册完成后，由新站点ETR向所有原始站点ETR发通知，再由原始站点ETR通知ITR删除临时表项。这个方案感觉相对更靠谱一些，可以将所有可能运行同组业务的数据中心站点ETR都设定到一个组里面，大家有事没事互通有无一下。

再有要解决ITR感知迁移后如何切换EID对应的Locator IP问题：1）使用上面几种感知迁移发生的方法后，都可以将ITR的EID对应Locator IP临时表项删除，由ITR重新发起一套EID的寻址流程获取新的Locator IP，缺点是稍慢。2）在上面第4种解决方案中，原始ETR收到新ETR的通知后，向ITR发个EID变更信息告知新的ETR地址，由ITR向新的ETR直接发请求获取新的Locator IP，不经MR/MS倒腾一遍手了，这样需要在LISP中多定义一个EID变更报文和相关处理流程。切换速度能提升些，但也稍微复杂了些。

提问题->找多个解决方案->比较不同方案的利弊，协议设计就是这么个过程。

LISP即使能够成事，至少也得2年以后了，所以大家可以先看个热闹，等RFC标准立起来再介入也不迟。也许Cisco回头觉着整这个太费劲，说不定哪天就偃旗息鼓了。别的厂商又真不见得有这么大号召力能把LISP忽悠起来。全当学着玩了，目前别拿LISP太当真。

另外多说一句LISP在Mobile里面的应用场景，Cisco已经今年5月份已经向IETF提出了draft-meyer-lisp-mn-05，其中mn就是mobile node缩写。简单来说就是在手机上支持ITR/ETR功能，可参考下图：



技术上很有想法，市场发展上不咋看好。感觉Cisco把手伸到手机里面，还不如前文提到的伸进服务器虚拟化里面有搞头。

小结

小结一下，数据中心多站点选路目前网络厂商单从路由角度来看没有什么好的方案，还是用DNS搞定更靠谱一些，只要应用程序的BS结构始终领先前行，暂时就还不用考虑其他的解决方案。让那些有钱有势的大厂商去试验吧，大家在后面跟进就是了。搞预研是有一定风险的行为，资源消耗了，万一路没选对，大厂商还能壮士断腕，小厂商就得折腰而终了，须慎研慎行。

凡事都有两面，换而言之，混乱的局面也是崛起的机会，如果谁能想到个啥招，搞个盒

子出来，从路由或者其他层面独立解决多站点选路的问题，还是可以一试的。大赚不好说，但搞到像F5/Redware/Citrix这种规模并不是没得奔头。

5.9 技术总结

对数据中心网络而言，当前的技术发展正处于一个关键时期，虽然Cisco暂时占先，但只是通过技术的先进性增大了其市场话语引导能力，为其在今后10年的数据中心市场的竞争中增加了一些砝码，绝不是说就一定会所向披靡。未来充满变数，哪怕是什么时候Google或Tencent推出了应用于云计算数据中心的网络设备或技术，作者都完全不会感到吃惊，一切皆有可能。就像Arista这两年横空出世，彻底的取代了Force10在作者心中“傻快”的霸者地位，预计其在今后几年的数据中心市场中势必会有所斩获，天下武功，唯快不破。

新技术是层出不穷的，前文介绍的这些技术只是作者知道的内容，眼界有限，相信还有更多更先进的技术在此章介绍之外。期望通过本文能够对读者学习其他技术也有所帮助，万事万物有果必有因，透过纷乱的报头字段、状态机变化和报文交互等协议机制设计表象，去了解清楚技术产生的背景原因和要解决的问题，势必在学习时可以达到事倍功半的效果。

用以下言语与技术同好共勉：在技术之路上，了解得越多，敬畏之心越重，但仍需不断前行，即使无法成为引领者，也必将超越原地踏步者。

6 终章

完了就是完了，其实没啥好多说的，想说的要说的该说的前面都已经说过了。文中做了不少预测，根据作者的恶趣味，最后在这里对那些神棍内容再总结一下凑凑字数。

在未来5-10年间作者认为：

市场方面

1、云计算市场出现不了如Microsoft于操作系统、Google于网络搜索或Cisco于数据通信的一家独大局面，但对多虚一的集中云与一虚多的分散云，市场划分会更加清晰，客户抗忽悠能力也将得到大幅度提升。

2、解决了安全问题后，基于服务的SaaS会占据更多的业务租用市场，中小企业自身IT资源消耗进而降低，业务能力反而提升。例如同时租用Google云的数据管理，Amazon云的人力资源，Microsoft云的ERP，Cisco云的统一通信和作者云的客户关系管理等系统来综合

搭建企业IT平台，会成为很时髦很常见的思路。（想创业的抓紧，SaaS机会贼多的，而且初始投入规模并不需要太大）

3、提供云服务（以SaaS为主）的产业将如雨后春笋般出现，这些服务提供商将会搭建大量的数据中心为客户提供云租用服务，也会成为网络设备厂商们的衣食父母。需求较小的企业都去直接租用服务了，因此数据中心步入了大型与巨型为主的时代，动辄成千上万的服务器节点绝对是小Case。数据中心产品的销售也将随之进入规模化采购阶段，搞定几个大客户，厂商一年下来就吃穿不愁了。

技术方面

1、VM之间互通技术之争会以硬件交换机进入服务器内部为最终结局，有可能是在网卡上实现，也有可能直接在主板上加转发芯片。毕竟从现在发展情况来看，芯片价格会越来越便宜，集成度会越来越高。

2、存储方面FCoE基于Ethernet带宽发展方面的优势，必将取代FC，当然过程会是比较漫长的，估计10年之后FC也还能占有一定的空间。

3、数据中心站点内部TRILL将会一统天下。巨型数据中心内，基于IP层面的交互会导致传输效率降低和部署复杂度提升，因此仍然会以Ethernet技术为主，而TRILL是目前看得到的最有希望胜出的公共标准。各个厂商的私有技术会将在规模稍小一些的大中型数据中心内有所应用，比如前面说的SaaS创业企业，其可能会更看重网络的高可靠性、高性能、易管理和易维护等私有技术强项的地方。而且网络规模较小，搞一家厂商的设备就差不多了，不需要考虑互通。

4、数据中心跨站点二层互联方面，RPR由于是公共标准可以成为种子选手，但其成长空间目前并不充分，还要看技术发展演进和各个厂家的态度。当然如果有哪家厂商愿意把自己的私有技术拿出来推成标准，也还是很有希望在市场上占据高点的。

5、在多站点选路方面，应该会有些新的技术标准出来，DNS方案一统天下的局面不会长久。这块谁都有机会，就看投入与机遇了。

7 感言

沥沥拉拉写了小两个月，长度和时间都远远超出了最初的计划，也耗费了不少的热情和精力，以后是不敢随便写这种大文章了。但整个写作过程对作者来说受益匪浅，不断总结是

自我提升的重要动力。后面休息休息还会再整理一些关于云计算数据中心安全、存储、性能和可靠性等方面的外篇，先在这里立个目标好做自我督促。

套用一些书中常看到的话，谨以此文献给我的家人朋友和同事，并纪念作者步入而立之年。顺便感谢每一位能从头读到这里的读者，你们的存在是我写作乐趣的源泉。