E3-vnet fabric white paper

Items	description
Docs creating date	2017.5.16
Docs finishing date	2017.6.11
Initial author	jzheng.bjtu@hotmail.com
Description	This document description why should we do newly design our virtual network and how, and in what way we can utilize our infrastructure. actually it the blueprint description.

Table of Contents

1.what challenges we are facing.

At presence OpenStack powered virtual network is not that regulated for its tenant ,because most of mechanisms are utilizing traditional Linux network function ,for example Linux bridge,Linux veth and even ovs are options, it enforce no infrastructure requirement outside of common servers ,we are not say it's mis-designed or something ,because OpenStack is more like a collective orchestrator which launches virtual compute ,network and storage ,hope it could run in heterogeneous environments ,so this is so hard for OpenStack orchestrated NFV appliances which need to guarantee High performance and High availability .

Which should we design one virtual infrastructure network to satisfy these demands :highly available and highly performed, and is this possible without specified hardware supported ,we shall say YES because of so many x86 based solution available ,DPDK is that good an choice in data plane .

So we are making innovative changes to virtual network to accommodate NFV appliances ,not incremental ones .

2.what service we are providing.

We make it clear we are providing services ,and services are layered in three:NFVIaaS,NFVPaaS and NFVSaaS.next let's go through these aspects .

2.1 NFV Infrastructure as a service

What can it provide actually? virtual network, yes it is. through collecting distributed physical network resources , virtualize them and make them a pool ,tenant can request these virtual network resources .it seems simple and clean ,but we should offer some operational interfaces to ease virtual network to be managed ,as NFV specific [1]states :it provide interface to VIM &VNFO in NFV MANO.

The following figure should describe the virtual network model:

[1] NFV specification: infrastructure overview.

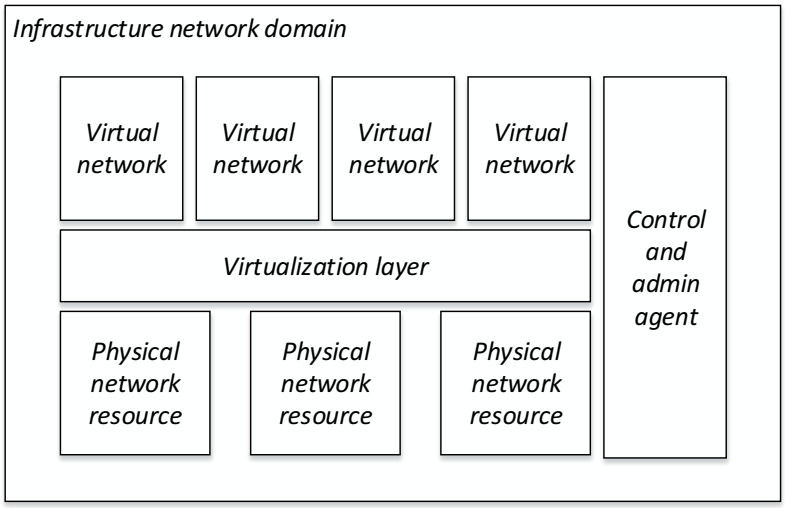

Figure 1 Infrastructure network domain model

It's very clear that what specific service the virtual network should provides in more detail:**E-LINE,E-LAN and E-TREE**[2].and subsequent requirement will be reiterated later .as for control and admin function ,control plane will be hybrid :distributed topology discovery and central management. Another admin function is OAM function, in more specific way: north-south OAM report function and east-west OAM function.

how external network can be led into our e3 vnet fabric? in three fashions:

- Tunnel access: edge vSwitch should decode tunnel and map the tunnel id into FCE and encapsulate them and deliver them into core provider network. typical tunnel types are VXLAN and GRE.
- Vlan access: similar to tunnel access, vlan id is mapped into FCE.
- IP based access: special for IPv4 in case we need to handle traffic from ISP, for scaling reason, it supports OSPF or BGP ECMP pathing from uplink routers.

Next we present several use cases.

2.1.1 NFVIaaS use case-OpenStack neutron access

We clearly know OpenStack neutron is truly virtual network, it offers its users L2 connectivity and L3 connectivity inside and outside neutron network, why should we still provide another kind of L2/L3 connectivity for its tenants then ? we address network service in NFV manner which scales well enough and is highly performed and highly available ,think about cisco's and bigswitch's OpenStack solution ,they employ their proprietary hardware to accelerate neutron network and even offer additive services.

Integrated with OpenStack Neutron ,we basically provides L2 connectivity :E-LAN and E-Line ,actually E-LAN is mostly wanted. Figure 2 will illustrate how:

[2] NFV specification – network domain overview.

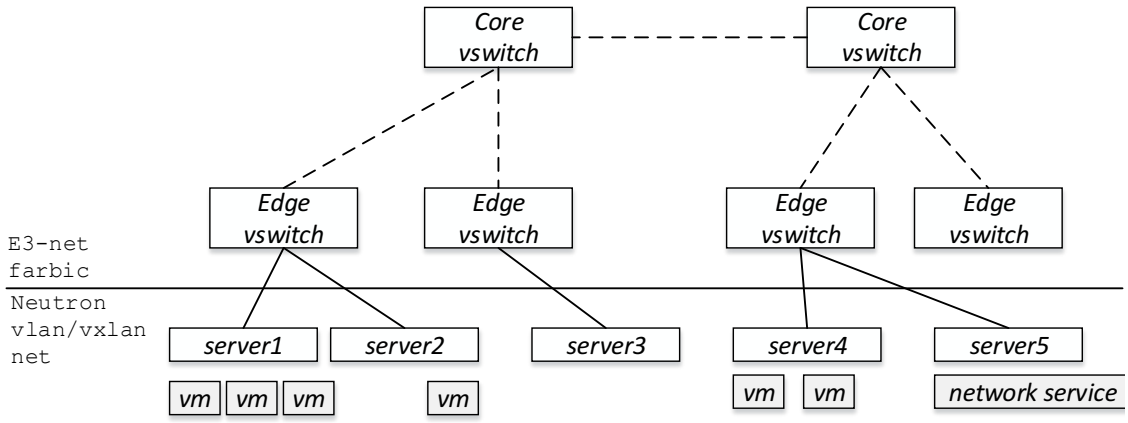

Figure 2 OpenStack neutorn powered by E3-net fabric

Note that E3-net fabric only offers l2 connectivity, it means that given a neutron network with network segmentation ID ,let's decide which edges servers should join the E-LAN:

The simplest way is every servers with every networks join the E-LAN ,this means every edges gets configured once new network is created ,this is terrible when BUM traffic floods within the E-LAN.

It must be confined ,but how ? maybe we can dynamically get the information about compute nodes and network nodes mapping ,**the alternative is we manually configure them**, which seems very efficient.

2.1.2 NFVIaaS use case-genuine NFVIaaS

NFVIaaS is the lower layer of NFVPaaS,similar to Case 1 in 2.1.1,the infrastructure only provides connectivity ,at this layer it never offer any other service .**with this ,we can schedule network function basing on the currently network resource layout ,in another word,upper layer is network oriented.**

We must emphasize the *control and admin* function of NFVIaaS with which we later orchestrate upper layer network function.

2.1.3 the benefit of our NFVIaaS

As we know ,more than 50% [3] workloads of the Data Center around the world has be virtualized ,virtualization must be the trend. With our E3-net fabric which applies to virtual and physical environment, we can deliver better experience to our potential customers because of our smart and reliable virtual infrastructure network ,these infrastructures consist of commodity servers ,thus reducing the overall expense and preventing vendor lock.

It conforms to NFV specification and principle to build the network function ,we hope it could be revolutionary change to this very little area.

[3] According to VMWARE statistics

2.2 NFV Platform as a service

Platform as a service,PaaS layer provides developers the necessary infrastructure resources which aids rapid innovation for our developers .what our PaaS offers is listed below:

- Network topology customization.
- VNF execution environment (VM/container).
- SDK for rapid development:P4 and C based toolchain.

As its name illustrates, NFVPaaS is platform as a service ,but with out NFVI ,what kind of platform we are going to providers ? and who are our potential customers of NFVPaaS and with our platform what they can do ?

What's our NFV platform?

Sure as you imagine ,basing on our NFVIaaS,we build a platform which provides enough virtual infrastructure (virtual network ,virtual compute and even virtual storage) interfaces ,with these we can rapidly develop our network function and verify them. once after that ,we even allow these functions to be published to the network function store ,then end customers can order them.

Through this platform, we are try to build our ecosystem ,even though we know it's possible to make it popular ,but we are going to use it and develop our own network function.

Who are our customers?

Any registered developers are our customers, and partners, and we insist **developers as partners** .

What interfaces should we provide?

Anything for our customer to complete and verify their network functions is provided.

We summarize them here:

- The network topology[4] to interconnect network functions.
- The metadata about the virtual network resource.
- The execution environment which may be VM or container .
- The network function framework SDK is recommended.
- … …

We must define a framework which is scalable and pluggable for newly developed VNFs to be scaled to satisfied our customers ' needs.

2.3 NFV Software as a service

Finally it comes to our end customer facing service layer,SaaS can directly serve our end customer in whatever manners ,no matter it's virtual tenants or physical campus network or enterprise network or even carrier-grade network facing customers .

[4] Which needs the fabric to be location agnostic and always be network oriented orchestration

The ways with which our end customers (as tenant) access our fabric network ease them to be well served by the network functions which reside inside the net fabric .

Our SaaS customer will be almost any network entities ,the following figure show a lot about the way SaaS goes :

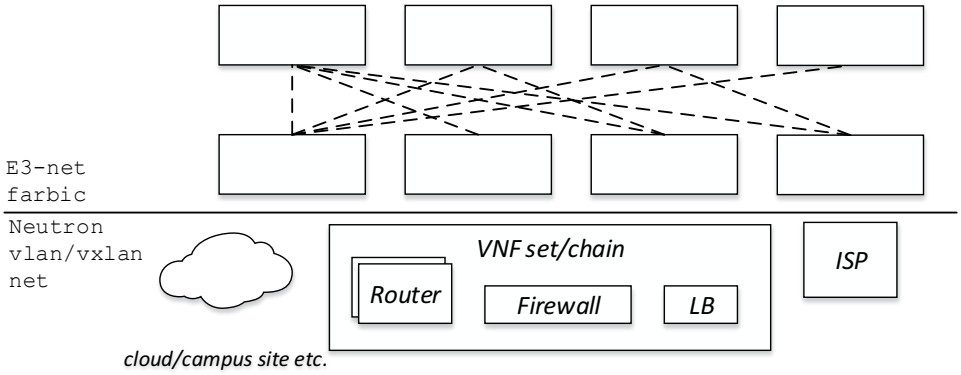

Figure 3 NFVSaaS use case

If one sentence is need to summarize what NFVSaaS is doing, I would say :with our infrastructure fabric connectivity is allocated to delivery end-user network service on end-user basis.

Typical network functions include vCPE[5],vRouter ,vFirewall and vLoadbalancer,most of them are pre-deployed ,in the docs we call them **builtin VNFs** .

2.4 relationship of these service layers

NFVIaaS relies on E3-net fabric which offers basic connectivity to make sure Infrastructure outside of fabric is pooled and can be serviced .both NFVSaaS and NFVPaaS rely on NFVIaaS which acts as their execution environment ,but what about the relationship between NFVaaS and NFVPaaS ,they have same execution environment.NFVPaaS should be regarded as PoC(proof of concept) environment while NFVSaaS is kind of product environment ,furthermore ,NFVPaaS can reinforce NFVSaaS with continuous VNF iteration ,update and even innovation .

[5] Which means customer premises equipment.

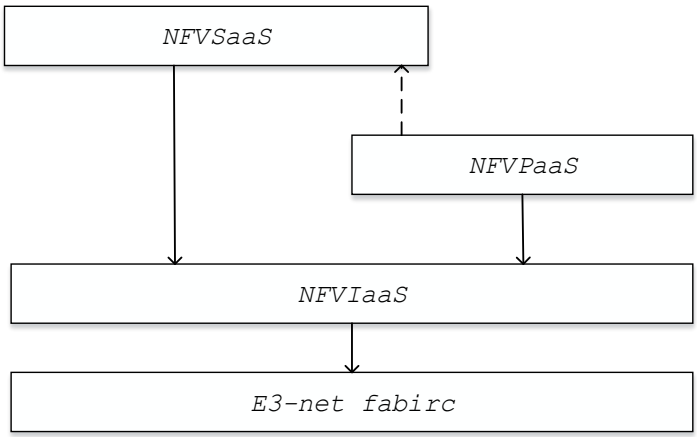

Figure 4 service layers dependencies

3.E3net Fabric introduction

the core idea behind E3nbet fabric is deliver users' packets through cross-connected network structure by employing smart routing in switching, *first we must make it clear why routing technology is necessary in our fabric structure*.

imagine we have a flat network ,a lot of end systems are connected to our flat ,although we can segregate users using vlan or something else, but it's still only one tier network which impose link layer addresses recording pressure on the switches ,this will not scale either ,in such circumstance ,networking virtualization must be conducted by users themselves ,a typical case is OpenStack neutron network which implements L2 connectivity with ML2 within compute and network nodes ,I do not mean this is not good since this framework is highly modularized and vendor neutral, but less adaptive to NFV workloads which intrinsically consolidates network scalability and availability.

In order to better regulate network resource ,we do network virtualization at infrastructure layer[6].what the core fabric looks like then ? what kind of services it provides ?sure as we stated before ,the fabric provides L2 connectivity[7] :E-LAN and E-Line. Once users' packets enter core fabric network , they will be routed through dedicated path. the core is smart enough to know how to generate the path ,this is where smart routing technology always the major control plane method.

now we can give this definition: **e3-net fabric is virtual core network which provides L2 connectivity while employs smart L3 routing technique as control plane**.

One principle is we do not introduce new protocol suites , we use existing ones except we may take hybrid control plane method since Data Center is always clustered and not that geographically distributed. distributed control plane is for topology auto discovery and central control is for others .

[6] In NFV's infrastructure reference, infrastructure network domain (IND) is individually a block .
[7] Large L2 network is the trend since it allows users to benefit from location independency.

3.1 protocol suites

what we are gonna do is implement L2 switching on top of L3 network infrastructure ,the obvious solution as you may know is VXLAN, but we do not use it as our transport protocol for two reasons:

- vxlan is too heavy and tedious, the vxlan mesh structure is not efficient either .
- L4 protocol stack is needed in VXLAN. For our fabric ,it's not agile at all.

The Ethernet over UDP solution definitely the transport efficiency .

How about MAC in MAC ?SPB[8] or TRILL[9]? Both of them offers multi-pathing , and we want the core function is simple and controller does most of work except forwarding.

SPB(PBB) is not our option since it does not contain any explicit fields to indicate the time to live,and it does not have explicit L3 routing headers either .TRILL is great except for its lack of support for multi-tenancy.

Finally what comes into our minds is MPLS L3 routing , consider we still need to keep Link layer information when the packet is on the fly in the core network ,Ethernet over MPLS is the encapsulation frame format we choose .

3.2 overview of the fabric.

The fabric is spine-leaf like, yes we have spine switches and leaf switches in our fabric network , but they are not cross-connected as the typical ways as traditional spine-leaf network does ,they are partial x-connected .

The following figure illustrates how the fabric is structured.

[8]Shortest path bridging, IEEE specification which utilize PBB encapsulation and IS-IS to do smart pathing and routing ,it naturally support multi-tenancy.

[9] Transparent interconnection of Lots of Links,IETF specification as opponent of SPB,TRILL is more like IP L3 routing ,because it explicitly contains src/dst nicknames(act as IP address) and TTL in TRILL header, it utilize IS-IS as dynamical routing protocol. However ,TRILL does not offer any multi-tenancy .

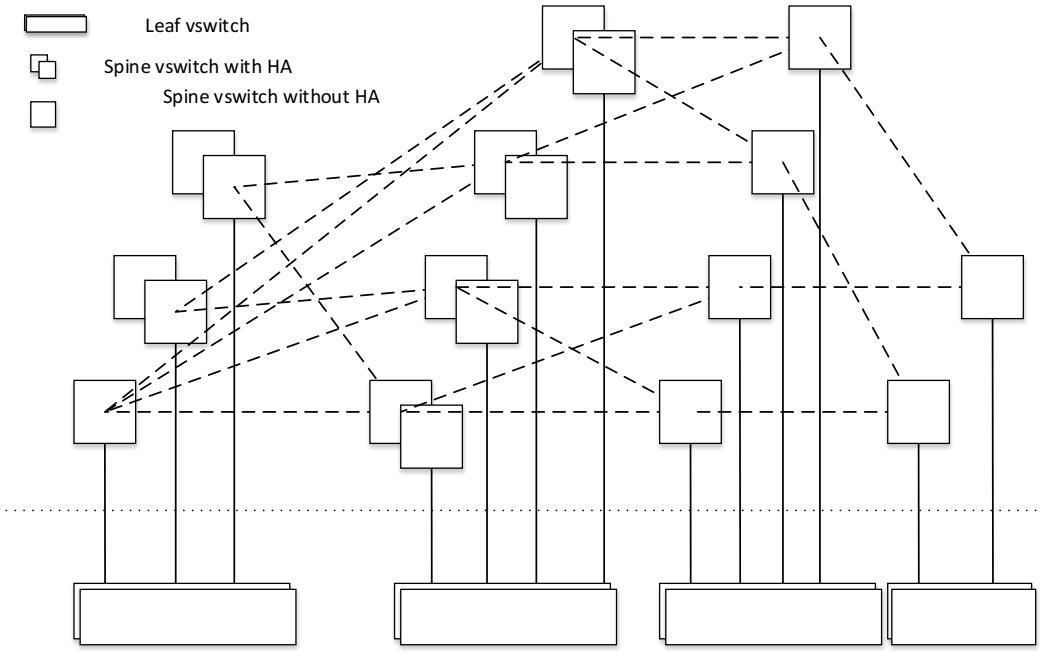

Figure 5 e3-net fabric structure general model

Let's introduce a little about the features of fabric, not all the features are covered.

- there are two layers in the fabric : spine and leaf. Leaf vSwitches are responsible for accessing user's traffic into fabric network , while spine vSwitches are responsible for forwarding them with the defined behaviors .
- spine vswitches act as backbone vswitches ,it's optional to make it highly available ,we can see in the figure the overlapped square shape stands for HA vSwitch ,the detail behind it will be considered later.
- virtual paths between leaf vswitches and spine vswitches can exist Equal Cost multiple Path(ECMP) which means traffic led into spine backbone can be along different paths ,doing so we can migrate potential bandwidth bottleneck or simply increase available bandwidth or migrate single point failure using multipath.
- Virtual paths within spine vswitches still can exist ECMP.
- Multipath is everywhere in our fabric, through topology-agnostic control ,we can better utilize them at the same time, not block part of them .

We must learn the topology through distributed control method, and let any these vswitches join and leave at any time . this should be one of our virtual Data Center network greatest requirements.

3.3 structure of spine network

spine network in our fabric is the backbone network , users' traffic should already be labeled by leaf network ,along with leaf vswitches, spine network install pre-computed label switching path(LSP) label entries .

spine network relies on physical LAN segment[10] to do basic per-hop forwarding, usually interfaces attached into the same LAN segment are configured L3 IP addresses within the same subnet, thus simplifying our forwarding procedure by directly addressing next hops.

Addresses can overlap, **BUT do not ever make two LAN segments address within overlapped address space.**

In our abstraction of our fabric structure, **spines vSwitches can be grouped together to be an island (site) as long as any of them has same direct connection(s) to the next hop vSwitches within all neighbor LAN segments**. Given a site, this topology can expand round by round ,but often we transform our topology in a pipelined way.

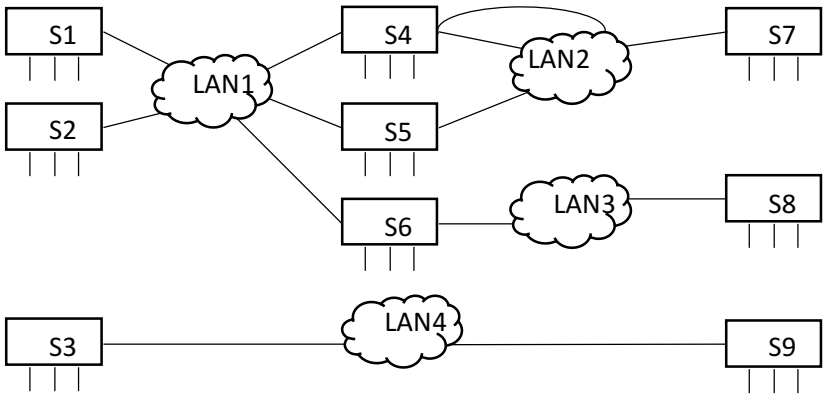

Figure 6 demo of pipelined spine vswitches structure

From Figure 6 ,obviously we can know S1 and S2 belong to the same site ,while S4 ,S5 and S6 do not group to be a site because S6 is attached to LAN3 which has no any direct connection to S4 and S5. S4 and S5 belongs to the same site even though S4 has more than one connection into LAN2,this is allowed since we have multiple physical paths ,and our fabric encourages to add redundant links physically to make it robust .

3.4 structure of leaf network

leaf network vswitches interfere with spine network and access network simultaneously, this section briefly tells how leaf vswitches are connected to spine vswitches .

given a spine site[11] ,the leaf vswitches are connected to the spine site in three ways : *direct access ,shared access and cross access .*

direct access is the simplest way ,as Figure 7 shows ,the leaf vswitch can only be accessed by one spine vswitch in spine vswitch in spine site.

[10] LAN segment is an area where broadcast behavior is confined.
[11] Note that spine site is the spine set, any vswitch in the set has the same connection to neighbor spine LAN segments.

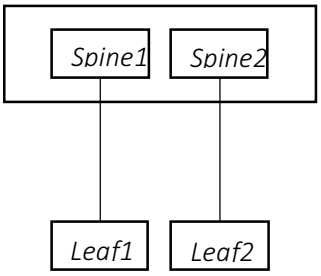

Figure 7 direct access model

In direct access model ,leaf1 is directly connected spine1,and leaf2 is directly connected spine2,leaf1 and leaf2 can not communicate directly ,they must rely on spine network. In shared access ,leaf vswitches are directly accessed via a LAN segment ,Figure 8 shows the model.

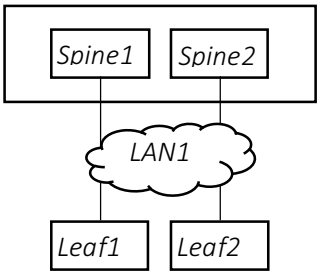

Figure 8 shared access model

The LAN1 is the media which connects spine and leaf vswitches ,the way is recommended since it reduces the expense of leaf vswitches communication and any leaf vswitch is accessible by spine vswitches ,vice versa .

Direct access and shared access are single homed,this means one leaf vswitch is only connected to one spine site ,our fabric allows one leaf vswitch is connected to more than one spine site at the same time .

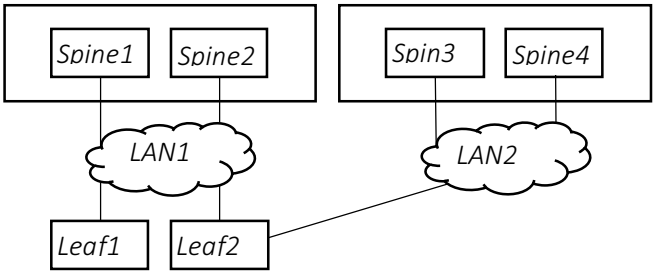

Figure 9 multi-homed cross-connected access

Leaf2 is connected to two spine site via two LAN segments. the benefit is leaf2 can be routed by multiple site, thus enabling BGP line like routing capability.

3.5 structure of access network

access network is the network where users are connected into fabric, there are three kinds of access network:vlan access network ,vxlan access network and IP based access network .

3.5.1 VLAN access network

VLAN is the common way to segregate user's traffic at Layer 2,in this case ,users' packets leave end-system with a VLAN tag which identifies which tenant network It belongs to in the confined network ,here we does not support port based access network manner since it does not scale that well. the limitation is VLAN access network only supports 4095 tenant networks at the access layer. it's ok because the VLAN range is meaningful locally. **VLAN id plus MAC from users' end-system is mapped to virtual network instance in the fabric.**

3.5.2 VXLAN access network

similar to VLAN access network, the access network support many more virtual networks than VLAN ,but VXLAN needs more overhead since VXLAN is mac over UDP where UDP is the transportation underlay protocol.in VXLAN access network ,leaf vswitches contain tunnel endpoint (TEP) where **VXLAN id plus MAC is mapped to virtual network instance in the fabric**.

3.5.3 IP based access network

Layer 3 network is the core access network from ISP where IP addresses bundles are assigned fabric, in such case VLAN is available for accessing ISP traffic. For reasons including ECMP load balancing or multi-path HA ,we introduced IP based access network .

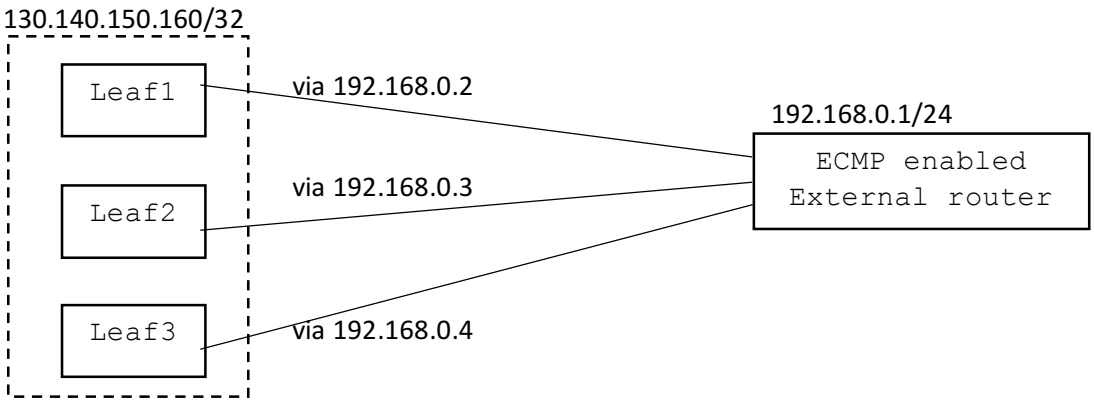

Figure 10 IP based access network architecture

Imagine we have a IP address 130.140.150.160/32 which is routable by an ECMP enabled external router, we configure private address ranging 192.168.0.0/24 on these interfaces, through configuring statically or dynamically external router,we have three ECMP routes to virtual network instances.

IP based access network scales with ECMP routes which can be established by dynamic routing protocol such as BGP and OSPF.

4.Ethernet Over MPLS routing and forwarding

we employ EoMPLS as our data encapsulation method, first we introduce what's EoMPLS.

4.1 introduction of EoMPLS

MPLS is originally designed to accelerate IP routing with connection oriented technology, though MPLS is originally for IP protocol, it still applies to other protocols, and this is why it's called multi-protocol label switch.

In our fabric we need to provide L2 connectivity, of course we need to keep L2 information of users' packets. next we present EoMPLS frame format.

In RFC 4448[12], in order sequentially deliver packets across MPLS based pseudo wire ,a 4-byte PW control word is introduced as bellows :

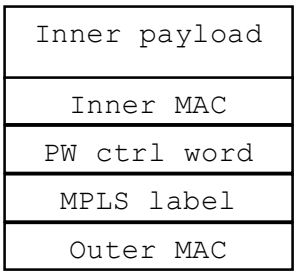

Figure 11 MPLS frame format with PW ctrl word

However, pseudo wire control word is not necessary since we do not sequentially deliver our packets, **we DROP this field in our fabric**.

The MPLS label occupies 32-bit width, the first 20 bit is the label field ,MPLS ranges from 0 to 2^20-1 ,often the segregation space is large enough, the following three bits are traffic class indicator (QoS and Explicit congestion Notification),earlier before 2009 ,this field is called EXP(for experimental use),in our fabric ,these field has another meaning special for our fabric .we define two classes as follows :

- **E3_NET_UNI_PATH**:the path is determined ,and the destination is clear . usually the MPLS label is swapped every hop.
- **E3_NET_MULTI_PATH**:for BUM(broadcast ,unknown destination and multicast traffic), we generate a TREE across the fabric ,and the packet gets replicated on the fly, thus minimizing the total traffic compared to VPLS full mesh like structure .usually the MPLS label will not change and it identifies E-LAN instance .

the 1 bit next to Traffic Class is an indicator which indicates whether it's the bottom of the stack ,usually we only push one layer label into the label stack ,we set it to 1 .

the last 8-bit the Time to Live ,similar to TTL in IP header ,it decrements per hop ,once the vSwitch receives a packet with TTL equal to 0, discard it.

[12] RFC 4448 - Encapsulation Methods for Transport of Ethernet over MPLS Networks

4.2 EoMPLS routing model

Respectively we introduces this section in two more detailed ways: E-LINE routing and E-LAN routing.

4.2.1 E-LINE routing model

E-LINE is an Ethernet service, the virtual topology in our fabric is line like shape ,whenever the packets to presented to endpoint of E-LINE instance ,it's delivered along it's pre-determined path, the fabric network is not aware of packet detail (whether know unicast or multicast, etc.), instead the TC field will always be E3_NET_UNI_PATH .

Imagine we have the E-LINE instance that spans several vSwitches :

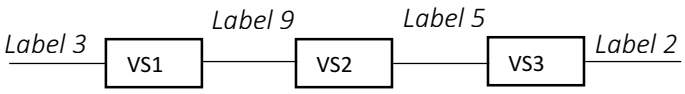

Figure 12 E-LINE label distribution model

as Figure 12 illustrates[13], the label is changed when it's sent out of a vswitch, note that the labels of these consecutive virtual links does not need to be always different as long as the label themselves can be distinguishable locally. we have a routing table in which packets are forwarded based on the label ,not the real destination addresses .

Table 1 demo of E-LINE routing table

vSwitch ID	In iface	In label	Out label	Next-hop
VS1	1	3	9	(192.168.1.2,2)
VS1	2	9	3	(leaf1,1)
VS2	1	9	5	(192.168.1.3,2)
VS2	2	5	9	(192.168.1.1,1)
VS3	1	5	2	(leaf2,2)
VS3	2	2	5	(192.168.1.2,1)

The first thing we must remember is the virtual link is bidirectional, then we must install symmetric routing table on the same virtual switch. the advantage is that the things to get a packet forwarded is getting simpler, it's possible for us to maintain a per-port index table, then we know the next-hop information according to the information contained in the packet MLPS header, the next-hop information is maintained by a address resolution process which could be ARP(standard Linux function).

4.2.2 E-LAN unicast routing model

Originally we imagined it's possible to use a label sourced from a virtual link to route to any destination, however the outer most label itself only presents information locally about what virtual link it belongs to, it does not specify which virtual link to go when there are more than

[13] assume these three virtual switches' spine interfaces are attached to the subnet:192.168.1.0/24, the addresses are respectively .1,.2 and .3 for simplicity.

forks in the vswitch node (i.e. in E-LAN tree topology). As VPLS does, the label topology must be full mesh like and symmetric. Even though an asymmetric label topology may reduce the number of labels, it does not make sense of significance, not sure whether complexity is introduced.

Image we have the following vswitches that comprise the spine framework for a E-LAN instance:

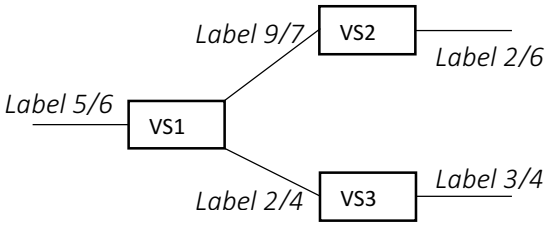

Figure 13 E-LAN unicast routing label model

Imagine we have N sites which are going to join E-LAN instance, there are (N-1) labels bound to every local virtual link, fortunately labels are only meaningful locally, labels from different virtual links may overlap and the overhead imposed on label space is not that large.

We have three paths with labels here:

- 5(1)<->9(2)<->2
- 6(1)<->2(3)<->3
- 6(2)<->7(1)<->4(3)<->4

Next we enumerate the possible routing table:

Table 2 demo of E-LAN uni-pathing routing table

vSwitch ID	In iface	In label	Out label	Next-hop
VS1	1	5	9	(192.168.1.2,2)
VS1	1	6	2	(192.168.1.3,3)
VS1	2	9	5	(leaf1,1)
VS1	2	7	4	(192.168.1.3,3)
VS1	3	2	6	(leaf1,1)
VS1	3	4	7	(192.168.1.2,2)
VS2	1	9	2	(leaf2,2)
VS2	1	7	6	(leaf2,2)
VS2	2	2	9	(192.168.1.1,1)
VS2	2	6	7	(192.168.1.1,1)
VS3	1	2	3	(leaf3,2)
VS3	1	4	4	(leaf3,2)
VS3	2	3	2	(192.168.1.1,1)
VS3	2	4	4	(192.168.1.1,1)

By install these label entries in respective virtual switches, whenever a virtual switch receives a labeled packet, it clearly knows what's the next hop is.

4.2.2 E-LAN multicast routing model

one thing which is crucial to reduce BUM traffic overhead is to replicate BUM packets when A packet flows through the multicast tree, this means virtual switch must know whether the packet is led to multiple sites. Also the when the packet arrives at Leaf virtual switch, it must know which site the packet is originated.

First we build a **group-shared tree** with which no Reserve Path Forwarding(RPF) check is essential any more. But how to carry the sender's identity?

Suppose we are going to deliver sender 's label which varies hop by hop then when packet reaches leaf site, it knows where the packet is from. we simply aggregate next-hop and deliver a copy to it, however there maybe more than one virtual links through the next hop, we can not distinguish the unicast path at all, we may need another layer label to carry sender's information.

Here we declare the E-LAN frame format first, we employ two layers in label stack:

- the bottom layer label carries the ID of sender site, control plane determines and encode the label.
- The top layer label is taken as ID to select next hops route, though it's call multicast route.

Let's take the following figure as an example how E-LAN multicast routing operates with the help of top layer label:

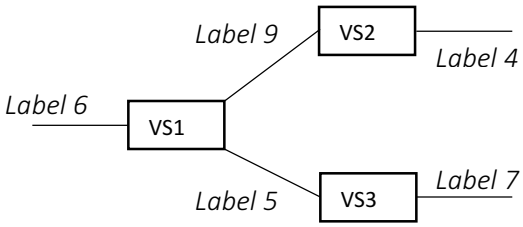

Figure 14 E-LAN multicast routing model

In Figure 14, multicast routing still uses one label to select routes in a virtual switch, however the installed routing table entry in a virtual switch should be a set of next-hop. Table 3 will show what multicast routing table looks like:

Table 3 demo of multicast routing table

vSwitch ID	In iface	In label	Next hop set
VS1	1	6	(9,192.168.1.2,2),(5,192.168.1.3,3)
VS1	2	9	(6,Leaf1,1),(5,192.168.1.3,3)
VS1	3	5	(6,Leaf1,1),(9,192.168.1.1,2)
VS2	1	9	(4,Leaf2,2)
VS2	2	4	(9,192.168.1.1,1)
VS3	1	5	(7,Leaf7,2)
VS3	2	7	(5,192.168.1.1,1)

4.3 EoMPLS forwarding process

This section describes how virtual switches forward packets according to previously introduced routing mechanism, basically divided into two categories: spine switches and leaf switches. Here we reference the definitions described in 802.1ah PBB specification:

- **BCB**: backbone core bridge, the bridge that does label switching as backbone only, does not senses what the customer's traffic looks like.
- **BEB**: backbone edge bridge, besides label switching, BEB also decode and encapsulates customers 's traffic.
- **B-component**: backbone component, a functional block that selects route and does label switching.
- **I-component**: instance component, a functional block that bridging customer's traffic with backbone traffic.
- **CNP**: customer network port, user facing port of I-component.
- **PIP**: provider instance port, backbone network facing port of I-component.
- **CBP**: customer backbone port, user facing port of B-component.
- **PNP**: provider network port, backbone network facing port of B-component.

4.3.1 forwarding process of spine vswitch

Spine virtual switch acts as a BCB, only contains B-component.The procedures are stepped as follows:

1. Upon receiving a packet, inspect the packet header and determine whether it contains a MPLS header, if not, deliver the packet to upper layer stack for further standard interface processing.
2. Search the routing table with tuple (in_port, label), usually the routing tables are on a per-port basis.
3. If the route exists and it's an unicast route, first we change the label to the one that appears at the next hop, the next hop link layer information as well.
4. Else if the route exists and it's a multicast route, walk through the multicast next-hop list, find the entries except the one the packet is received from, respectively change the label and next hop link layer addresses, send them out.

4.3.2 forwarding process of leaf vswitch

Leaf virtual switch is much more complex than spine virtual switch, it contains both components as figure 15 shows:

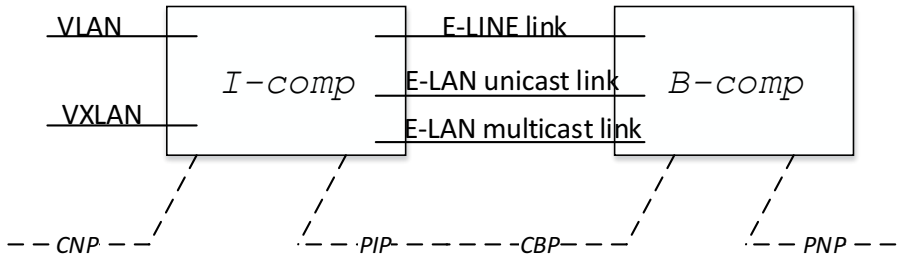

Figure 15 components in Leaf(BEB) switch

Once I-component receives a packet from CNP, it decode the packet (with VLAN or VXLAN encoding) and decide which VSI (virtual switch instance) it belongs to, next according to the VSI type (E-LINE or E-LAN), we determine what label it's encapsulated with.

- For E-LINE virtual switch instance, learn the mac-port mapping relationship on a per-vsi basis. There is only one fabric virtual link as PIP, so we push the related MPLS header to the packet, and then send it out of PIP.
- For E-LAN virtual switch instance, do mac snooping as E-LINE VSI does. However, there are more than one PIPs within a VSI, each of them is allocated with a label which identifies which site(s) to go. According to the destination link layer address, search the mac-port mapping entry, if found as a unicast path, push the associated label send it out of PIP. Otherwise, find the VSI associated multicast fabric virtual link. First we push the unique site MPLS which identified which way to go, then the next MPLS header is the multicast fabric virtual link associated label which is used to select next hop.

Usually PIP and CBP are paired and the channel maybe a queue.

B-component functionality includes the one that is implemented in B-component of BCB, what is different is the forwarding process is completely symmetric, while B-component of BEB is not.

- When packet comes from PNP, it searches the label-VSI mapping table to decide which VSI it belongs to, once is down, send it out to CBP.
- The packet that comes from CBP will be prepended with link layer addresses, and then sent out to PNP.

As for multicast packet, I-component could learn MAC and fabric virtual link mapping through bottom label, and it's easy to main multiple VSI instance at BEB.

4.4 ECMP consideration in fabric

Equal Cost Multiple Path in our E3-vnet fabric provides multiple paths for the same site set, it could apply to both E-LINE and E-LAN Ethernet services. In our fabric, given a set of sites, ECMP can be accomplished by establishing more than one Ethernet Service between them and bind them to the one VSI in I-component of BEB, thus we could balance the traffic across all the active Ethernet service instance as the following figure illustrates:

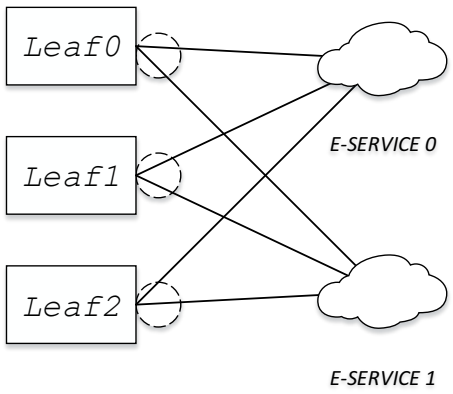

Figure 16 ECMP model in E3Fabric

Still remember how Ethernet link bonding works? The ECMP does the same way as these virtual links are in active mode, however, this does not sense what's going on outside at the core? That mean the leaf nodes does not know whether the Ethernet service works right outside there.

But with Connectivity Fault Management (802.1ag) Ethernet OAM protocol, it's possible for us to detect link failure and Ethernet service failure, thus deactivating E-Service in ECMP.

Note with ECMP forwarding, it's all right to forwarding packets asymmetrically, because it packets will finally go to the target site.

5.OAM considerations

BigSwitch[14] has two major products which they call Big Cloud Fabric(BCF) and Big Monitoring Fabric, the main idea behind it is to aggregate all the underlying leaf and spine switches into one logical switches, then through central controlling, it provides uniform RESTFUL API. Then through rapid development and integration, the BCF can satisfy almost any scenarios including OpenStack, VMware, customized data center.

What's emphasized here is its BMF product, from this we know Data Center visibility matters a lot during operations, it sensors network traffic and presents to customers, moreover, it provides automatic troubleshooting capability all through the ONE API entrance.

But what's OAM? You may remember ping and traceroute in Linux box, it's dedicated to L3 connectivity, ping is for end-to-end connectivity and traceroute is for hop-by-hop connectivity.

Since our fabric offers L2 connectivity, we shall employ some kind OAM protocol for Ethernet. Lots of protocols and standards are available, here we choose IEEE 802.1ag Ethernet Service[15] OAM standard as our OAM mechanism in our fabric, 802.1ag is also called CFM(Connectivity Fault Management) protocol.

There are two kinds of endpoints in CFM, they are MEP (Maintenance End Point) and MIP(Maintenance Immediate Point). Since it has hierarchical structure in CFM, they are customer OAM, provider OAM, and even operator OAM[16].

Our fabric offers end-to-end E-LINE Ethernet service and Site-to-Sites E-LAN Ethernet service, considering the complexity we may meet, here we only Implement customer OAM. This means the fabric spine vswitch will not sensor OAM traffic at all and regard them as normal traffic.

Note: though we implement customer layer OAM function, the OAM traffic is never generated by customers, instead, the fabric leaf vswitch generates it. and only loopback (unicast and multicast) is realized. Strictly speaking, the control agent on leaf generates the traffic to one of

[14] A company which employs SDN as its technology towards Data Center networking. http://www.bigswitch.com/
[15] note it's not Ethernet Link OAM.
[16] There may be more than three kinds listed above.

the attached path(since ECMP exits in a VSI, specify the path explicitly) in the VSI, doing so will greatly alleviate the burden of data plane.[17]

6.control plane functions

we have the philosophy in control plane: **one logical switch abstraction and uniform restful API along with its high availability.** In this article we call the controller **e3-orchestrator**.

6.1 south bound interfaces

6.1.1 virtual switches topology learning

the first thing to care about is how to setup the fabric global topology map, statically or dynamically. The considerations should be taken into is whether dynamical topology is stable when virtual switches come and go at any time, and this may render service disruption (the defined Ethernet services may migrate when topology changes).

Given a graphic which describes the topology logically, denote the expression: <N, E> where N is the node set and E is the edge set. When a virtual switch joins the fabric network, the new <N', E'> impose no change on current Ethernet services. However, when a virtual switch leaves the fabric network, the Ethernet connectivity services which rely on the virtual switch are influenced. In such a case, the e3-orchestrator should sensor the underlying change in infrastructure.

In what way the topology is setup? in a virtual switch, whenever a physical port is attached to virtual switch, there is a corresponding tap device which is special for slow protocol processing with Linux standard protocol stack.

By doing so, we could customize our topology discovery using application layer library, the simplest way is to broadcast datagram which includes chassis ID and link ID, then any directly attached interfaces will receive that advertisement and report neighborship to central controller which will update database.

6.1.2 virtual switch configuration channel

controller must preserve a channel to push commands or fetch profiles, the controller initiates the session and it's bidirectional. the would be any RPC variant and encodings, for instance, if we choose python as the programming language, we may choose JSONRPC as the transport middleware.

6.1.3 virtual switch report channel

Usually the the another control session which virtual switch initiates does work like reporting status of the virtual switch, it does not require controller to respond and acknowledge, in another word, it does advertisement only.

6.2 north bound interfaces

[17] If DPDK is used as our data plane kit, we must choose the DPDK with version >= 17.02 where tap device PMD is introduced and still evolves in DPDK 17.05.

North bound interface is the interface which is exposed to external entity, usually we design the interface to be RESTful like. the provided managed objects include (but not limited to):

- virtual switches.
- Ethernet services.
- visualized elements.

Note that in order to integrate with 3rd part application, API SDK for a programming language (Python, Java) is also necessary.

7.summary

In order to better deliver customers' traffic across our fabric network and maximize infrastructure utilization, the controller must quantize the network resources, orchestrate Ethernet service based on users provided policies (shortest path, weighted shared, exclusive... …). It must be smart enough to meeting changing requirements to virtualize network.