E3Fabric Internals

Items	internals
Docs creating date	Feb 12,2018
Docs finishing date	
Initial author	jzheng.bjtu@hotmail.com
Description	This article mainly introduces internals of the e3fabric

E3fabric provides a highly available and highly performed virtual network for NFV workloads with the support of the COTS hardware by migrating virtual network into infrastructure network while retaining scalability. we achieve this by eliminating the overhead of virtual networking processing in compute entity[1], thus preserving all the processor resource to VNF business logics[2]. While in order to scale datacenter-widely, we use dedicated servers to do virtual switching in the infrastructure network[3]. To construct the virtual network with maximized throughput and minimized latency and jitter, we take the centralized routing as the forwarding policy, generalized speaking, in a SDN[4] way. Note this will be working together with customized container[5] technology to realize a IaaS for NFV[6].

This paper is focusing on networking virtualization. the subsequent chapters will cover the specific parts.

Infrastructure Network Resource Introduction

As mentioned, E3fabric is a link level network virtualization solution, it provides virtual layer 2 network in datacenter in a very efficient way with little overhead. Let's clarify the infrastructure network resource first.

This section includes compute node network resource, infrastructure switch network resource and network node network resource.

Compute nodes host virtualized network function, the connectivity to virtual network is obtained through x86 based PCIe NIC[7].the NIC MAY support SR-IOV based

---

[1] The universe compute wrapper often includes bare metal, container, virtual machine, we call the hosting server **compute node** all through the paper.

[2] the compute entity employs hardware assisted virtual switching to connect to e3fabric network, i.e. SR-IOV NIC switch

[3] this is viable to do switching in a large scale infrastructure with COTS large volume server and switch, by employing DPDK as the data plane acceleration, VPP shows us a good example of how to scale. It can handle traffic at the rate of potential hundreds of Gigabit per second. we call these hosting server **network node**.

[4] quite clearly, we do not use OpenFlow or other standard control plane protocol, it's proprietary one.

[5] We plan to employ containerd(https://containerd.io), since it's been a project of CNCF, and open sourced, it's long term developed and supported by the community.

[6] Usually we call it the NFVIaaS, this is a platform that deliver cloud native VNF services.

[7] Network Interface Card

physical function(PF) and virtual function(VF) partition in order to able to support multiple networks. That's to say, **the traffic originated from VNF is mapped to virtual network by selecting an 802.1q VLAN tag for it.** and SR_IOV switch is responsible for multiplexing or de-multiplexing traffic on the same NIC(link). The model is depicted below:

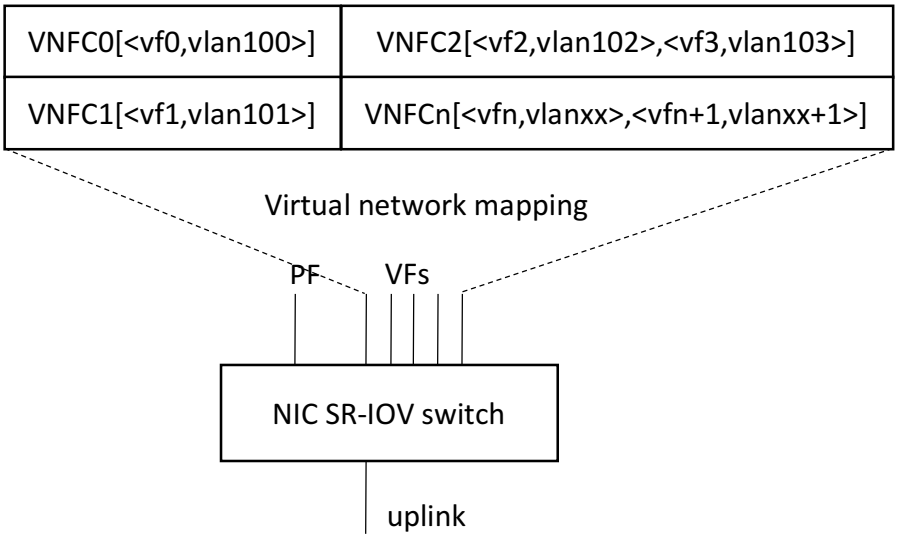

Figure 1 Host virtual network mapping

As Figure 1 illustrates, different VNFC[8]s map to separated VNFs, often every VF has chosen a VLAN ID for it. cases are common when VNFs need to attach more than one **virtual interface**[9] to the same virtual network, in such circumstance, these VFs share same VLAN ID[10].

For administrative purpose, compute node should report these SR-IOV backed PFs and VFs to controller.

This solution requires no special functions for infrastructure switch except that it MAY support 802.1q based forwarding mechanism to optimize underlay traffic. as TRILL[11] emphasizes, the underlay infrastructure switch provides 802.1D(802.1Q) Layer2 connectivity, these switches can employ their own loop free technologies, So does E3Fabric.

As default deployment, all ports of a switch should work in TRUNK mode to allow all configured VLAN traffic to pass. Since these VLAN is locally scoped, we can configure them[12] in management plane.

Network node use PCIe NIC as with compute node does, it still employs SR-IOV to partition virtual functions in order to balance load to different CPUs in case NIC is of 25Gbps/40Gbps/100Gbps rate when a single CPU can not handle all the traffic in a burst traffic pattern. Unlike RSS[13] or Flow Director, EoMPLS[14] is not supported to share traffic among CPUs for most high volume NICs, SR-IOV partitioning works it around.

---

[8] VNFC: virtual network function component.
[9] From VNF's view, they would rather call the networking attach point virtual interface.
[10] Control Plane guarantees it.
[11] TRILL: Transparent Interconnect of Lots of Links
[12] allowed VLAN range is associated to LAN ZONE object.
[13] RSS: Receive Side Scaling.
[14] EoMPLS: Ethernet over MPLS, RFC 4448

Virtual Network Resource introduction

For Cloud Native user/developer, what the VNF wants to know is what virtual network(s) it's accessing, it knows the exact virtual interface. NFVIaaS MUST expose these virtual interfaces along with their metadata to VNFs.

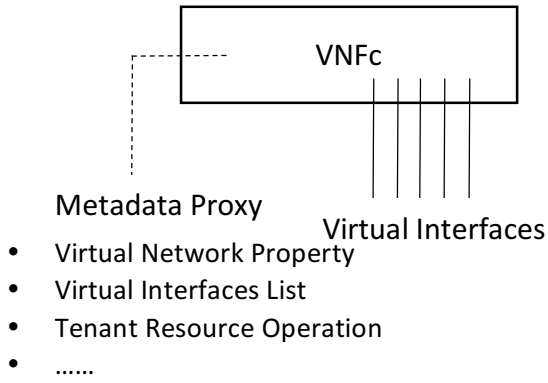

Figure 2 virtual network resource overview

More detail about metadata will be described in NFVIaaS documents. these virtual interfaces will be delivered to VNF container with the requested format.

Virtual Network Orchestration Scenario

This chapter includes the several cases of Virtual Network Service Orchestration. Here we define two terms: **LANZONE affinity** and **anti-LANZONE affinity**.

- **LANZONE affinity**: the policy that compute nodes which connected to the Ethernet service must reside in the same LANZONE, the Ethernet Service involves no network nodes to do forwarding, instead, they share the same VLAN ID.
- **Anti-LANZONE affinity**: the policy that compute nodes MUST span across more than one LANZONE, and network nodes act as intermediate proxy to forward traffic from one LANZONE to another.

In-Host Ethernet service orchestration

In-Host Ethernet Service is the service case where two or more VNFCs reside in the same compute node, multiple virtual interfaces share the same VLAN ID and exchange packets via SR-IOV NIC switch[15].each single virtual interface in the host can join and leave the Ethernet service without involving network nodes.

In-LANZONE Ethernet service orchestration

In-LANZONE Ethernet service is the service case where all the virtual interfaces of VNFCs are connected to the same LANZONE, comparing with In-Host Ethernet Service, the difference is these interfaces MAY span more than one compute node, but they are still in

[15] this is less optimized because packet flows through PCIe bus to NIC switch, then echoes back, it's still hairpin pattern.to improve performance, **we MAY use in-memory channel to replace it if there is no other VNFCs joining the virtual Ethernet service**.

the same LANZONE, even no Ethernet service instance is created at network nodes. In-Host Ethernet service is the special case of In-LANZONE Ethernet service.

Cross-LANZONE Ethernet service orchestration

Cross-LANZONE Ethernet service is the service case where the virtual interfaces MAY be connected to more than one LANZONE, the virtual interfaces which are connected to different LANZONEs MAY have different VLAN IDs[16], each of the virtual interfaces can join and leave the Ethernet service at any time. Once a LANZONE joins an Ethernet service, the VLAN ID is determined, subsequent virtual interfaces will share the same VLAN ID. the VLAN ID must be within the defined VLAN range.

[16] The VLAN ID is locally determined, scaling is not a problem.