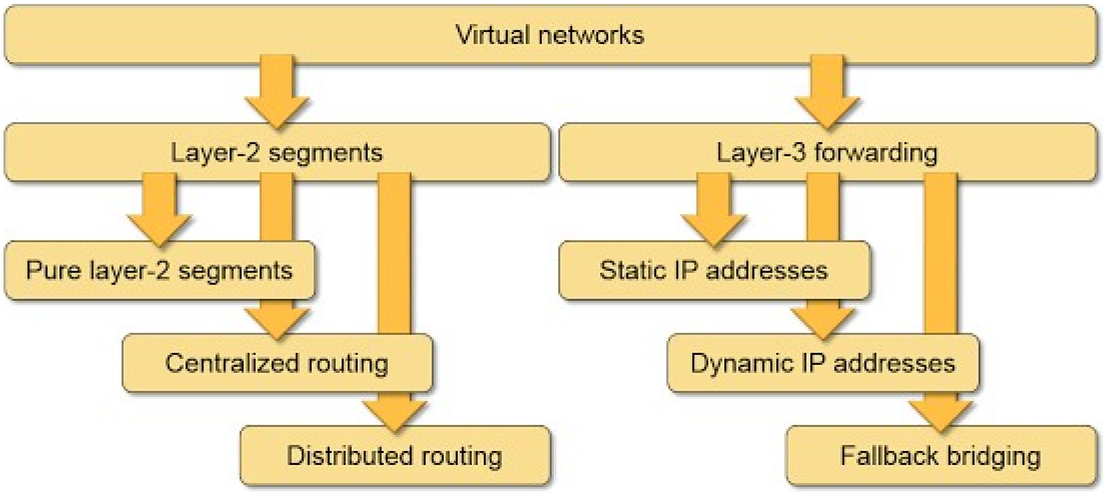
# 多租户的vpc实现方案



1.vlan
2.vpn
3.physical underlay network and virtual overlay network

# 简单的分类和例子

Userspace solutions: from Cisco Cloud Services Router and CohesiveFT VNS3 to Cisco Nexus 1000V Interlink and CloudSwitch;
Layer-2 solutions (VLANs, Metro Ethernet, Edge Virtual Bridging – 802.1Qbg)
MAC-over-IP solutions (VXLAN, Hyper-V Network Virtualization, VMware NSX)
IP-over-IP solutions (Juniper Contrail, Amazon EC2)

# 更加详细的分类和例子

# Layer-2 or layer-3 networks?

Some virtual networking solutions emulate thick coax cable (more precisely, layer-2 switch), giving their users the impression of having regular VLAN-like layer-2 segments.

**Examples**: traditional VLANs, VXLAN on Nexus 1000v, VXLAN on VMware vCNS, VMware NSX, Nuage Networks Virtual Services Platform, OpenStack Open vSwitch Neutron plugin.

Other solutions perform layer-3 forwarding at the first hop (vNIC-to-vSwitch boundary), implementing a pure layer-3 network.

**Examples**: Hyper-V Network Virtualization, Juniper Contrail, Amazon VPC.

## Layer-2 networks with layer-3 forwarding

Every layer-2 virtual networking solution allows you to implement layer-3 forwarding on top of *pure layer-2 segments* with a multi-NIC VM.

Some virtual networking solutions provide *centralized built-in layer-3 gateways* (routers) that you can use to connect layer-2 segments.

**Examples**: inter-VLAN routing, VMware NSX, OpenStack

Other layer-2 solutions provide *distributed routing* – the same default gateway IP and MAC address are present in every first-hop switch, resulting in optimal end-to-end traffic flow.

**Examples**: Cisco DFA, Arista VARP, Juniper QFabric, VMware NSX, Nuage VSP, Distributed layer-3 forwarding in OpenStack Icehouse release.

## Layer-3 networks and dynamic IP addresses

Some layer-3 virtual networking solutions assign *static IP addresses* to end hosts. The end-to-end layer-3 forwarding is determined by the orchestration system.

**Example**: Amazon VPC

Other layer-3 virtual networking solutions allow *dynamic IP addresses* (example: customer DHCP server) or IP address migration between cluster members.

**Examples**: Hyper-V network virtualization in Windows Server 2012 R2, Juniper Contrail

Finally, there are layer-3 solutions that *fall back to layer-2 forwarding* when they cannot route the packet (example: non-IP protocols).

Example: Juniper Contrail

# Why does it matter?

In a nutshell: the further away from bridging a solution is, the more scalable it is from the architectural perspective (there’s always an odd chance of having clumsy implementation of a great architecture). No wonder Amazon VPC and Hyper-V network virtualization (also used within the Azure cloud) lean so far toward pure layer-3 forwarding.

VMware NSX 和 Hyper-V 对网络虚拟化中 Packet forwarding 的行为作出了详细文档描述，但是亚马逊没有任何相关文档来说明 AWS VPC 中的 Packet forwarding 机制，尽管亚马逊采用的是私有的解决方案（深度定制的 Xen Hypervisor 和自行定制的虚拟交换机），但是通过分析 AWS 的网络特征和用户文档搞清楚它的一些细节也不是一件很困难的事情。

Chiradeep Vittal 通过运行一系列测试写了一篇博客来分享他的心得，总结如下：

1. 亚马逊 VPC 的虚拟交换机只做 L3 unicast IPV4 forwarding（类似于最新的 Hyper-V 网络形态），所有的非 IPv4 流量和 IPv4 多播和广播会被 drop 掉。
2. 在 Hypervisor 层虚拟交换机中 L3 的转发并不减 TTL，看起来就像所有的虚拟机都处于同一个子网中。（猜测：毫无疑问，此功能需要有类似避免回路出现的功能辅助，否则网络可能会被阻塞。）
3. Hypervisor 代理了所有的 ARP 请求并且回复目标虚拟机的 MAC 地址或第一跳的网关地址（早期的 AWS VPC 实现使用相同的目的 MAC 地址回复所有的 ARP 请求）
4. 虚拟交换机实现了类似路由的功能，例如，如果 ping 的是默认网关会响应，如果 ping 的是其它子网的网关，数据包会被 drop 掉。

这种实现看起来平淡无奇， 但是，慢着，这还不是全部，Amazon VPC 转发模型绝妙的地方在于，他是 multi-VRF（multiple routing tables）机制，用户可以在 VPC 中创建 multiple routing tables 并把他们当中的某个分配给其中一个子网。

你可以，举例来说，使用默认路由来路由 internet 发起的请求，并把这些请求路由到 web server 所在的子网，把访问数据库的请求路由到你自己的数据中心，对应用服务器之间的网络请求（本地链接）使用非默认路由等等，如果你是一个 MPLS/VPN geek 用这一特性来分拆路由表这是一个很 cool 的特性，但是，同时对于那些想要把已经存在的 L2 网络 migrate 到云中的用户来说也是一个挑战。

由于 Amazon 不开源，我们看看 opencontrail 的实现方案，vxlan 方案，以及 noc 方案，最后做一个比较

# opencontrail 方案

## 数据封装格式 IP

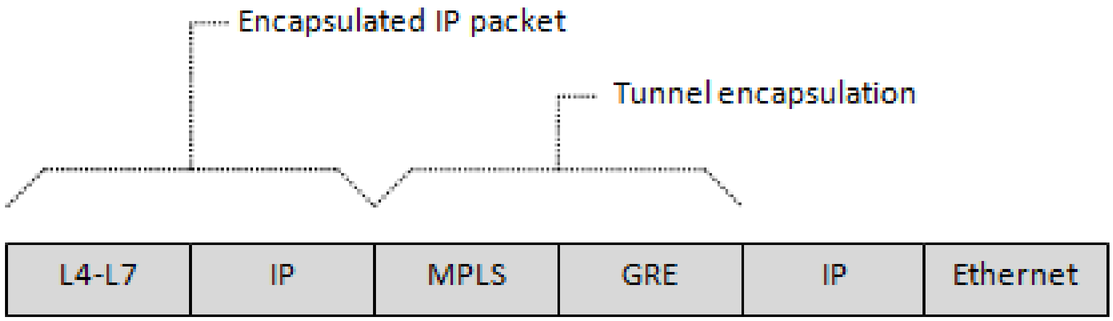


Figure 8: IP over MPLS over GRE Packet Format

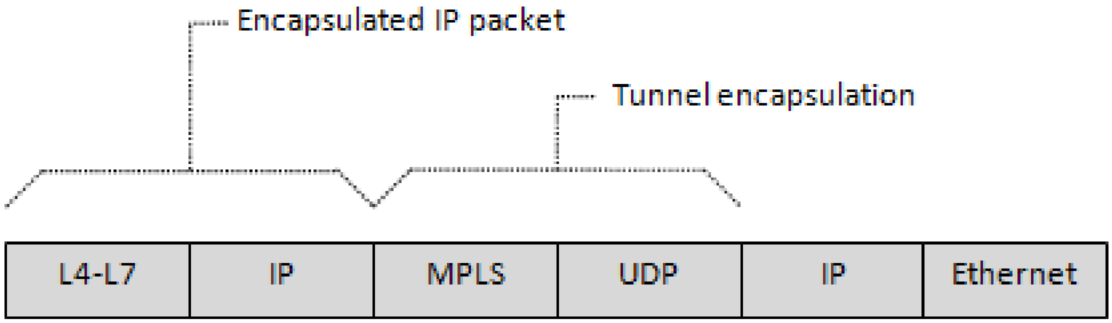


Figure 11: IP over MPLS over UDP Packet Format

# 数据封装格式 ethernet

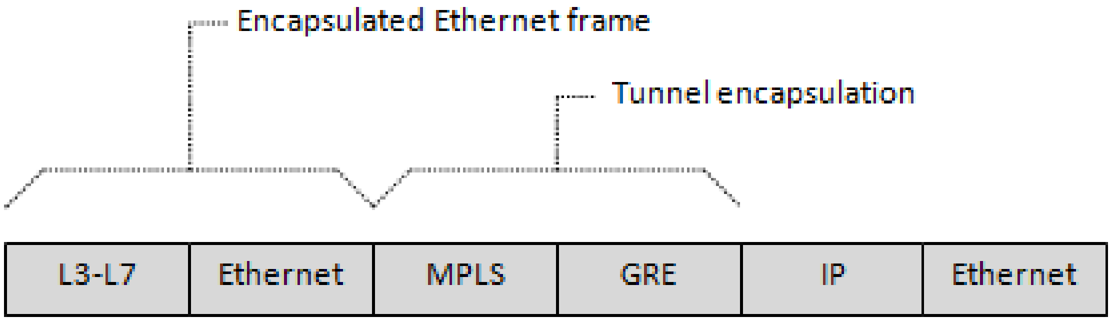


Figure 9: Ethernet over MPLS over GRE Packet Format

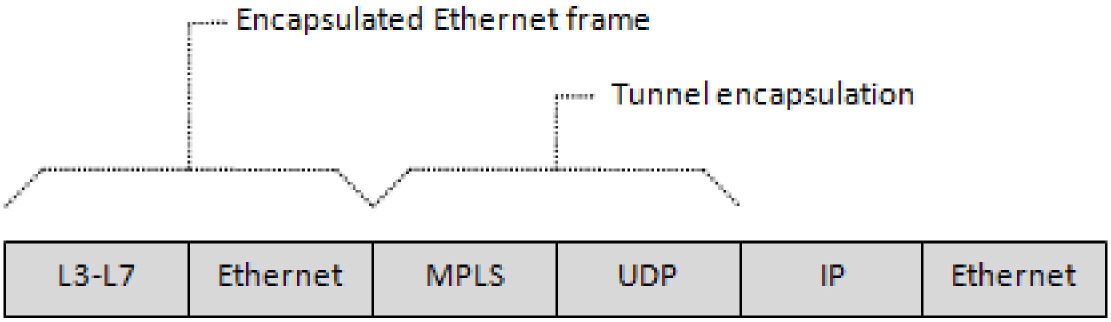


Figure 12: Ethernet over MPLS over UDP Packet Format

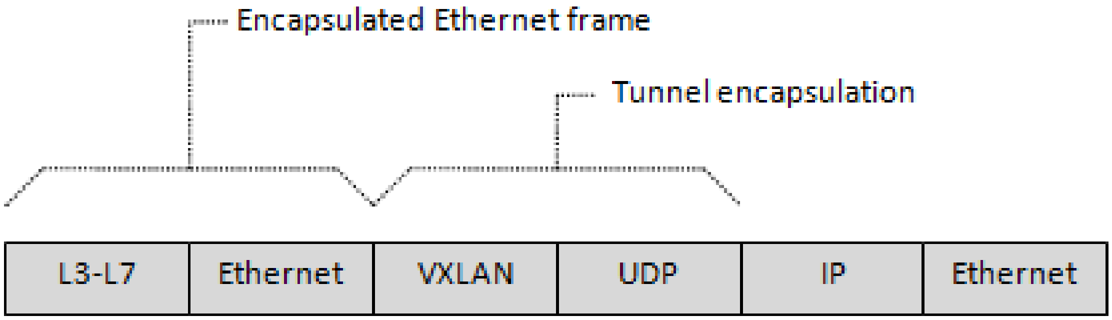


Figure 10: Ethernet over VXLAN Packet Format

## OpenContrail Architecture

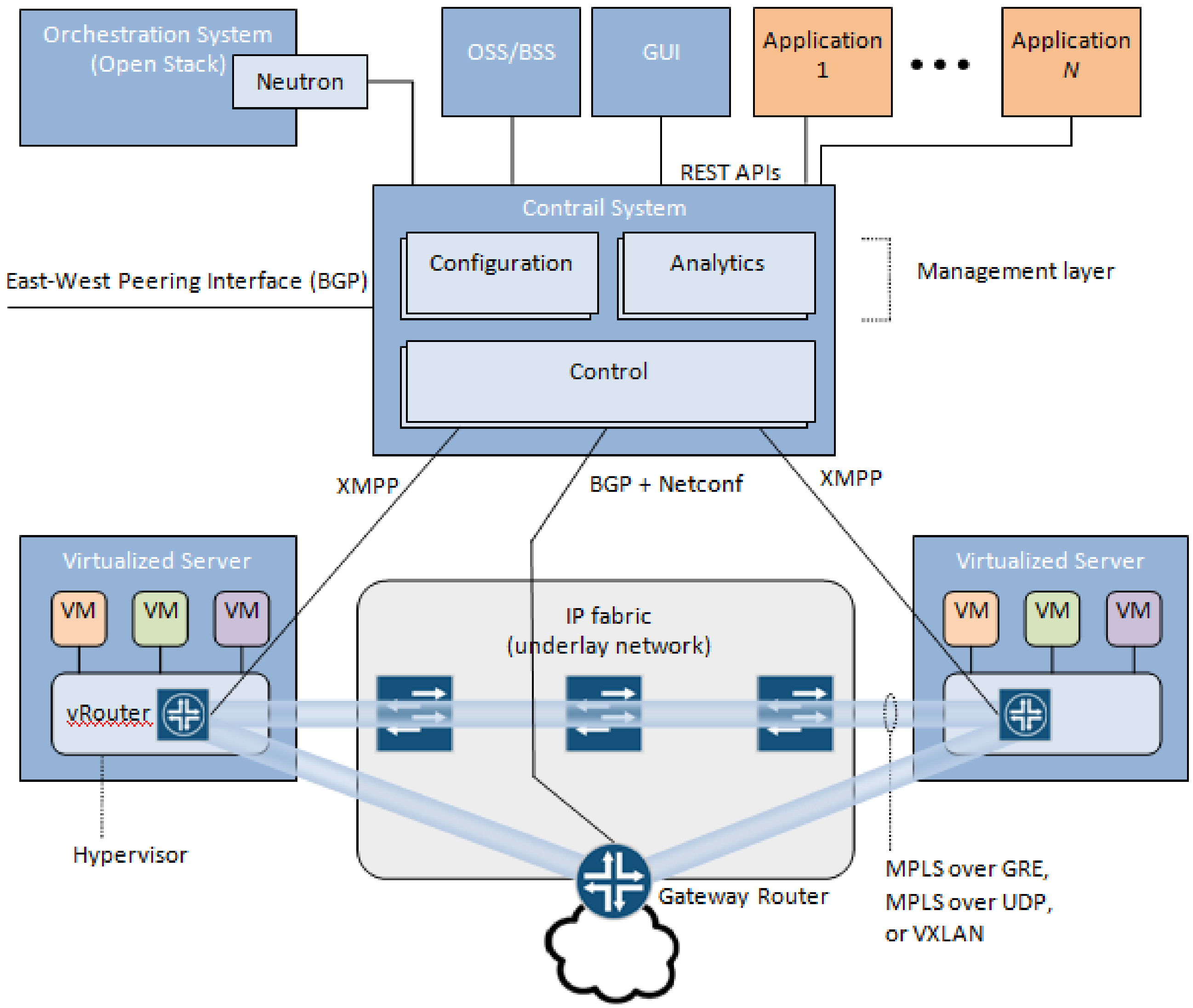


Figure 1: OpenContrail System Overview

## 重点关注 vrouter

The vRouters should be thought of as network elements implemented entirely in software. They are responsible for forwarding packets from one virtual machine to other virtual machines via a set of server-to-server tunnels. The tunnels form an overlay network sitting on top of a physical IP-over-Ethernet network.  Each vRouter consists of two parts: a user space agent that implements the control plane and a kernel module that implements the forwarding engine.

## Implement

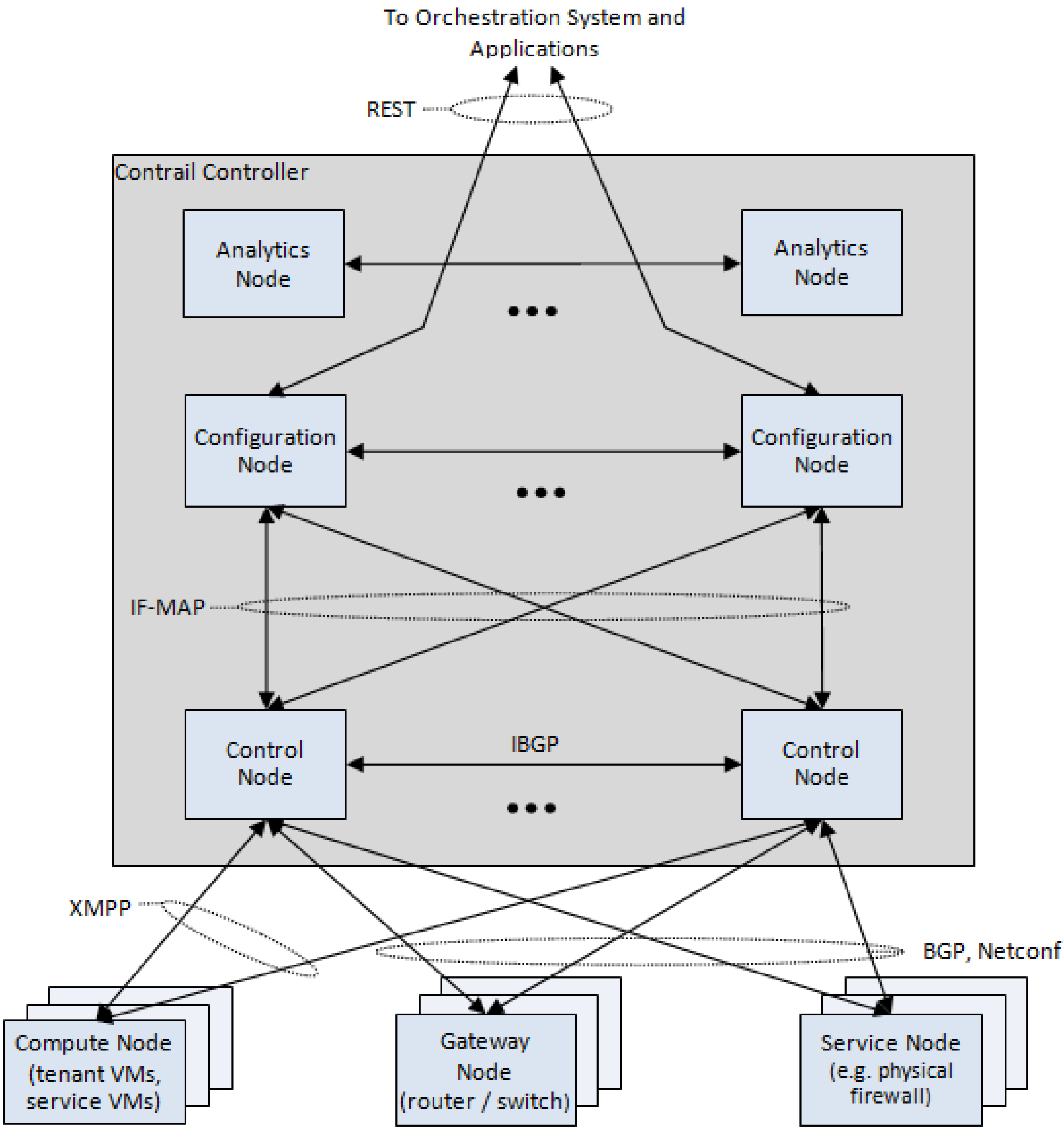

To Orchestration System and Applications
REST
Contrail Controller
Analytics Node
Analytics Node
Configuration Node
Configuration Node
IF-MAP
Control Node
IBGP
Control Node
XMPP
BGP, Netconf
Compute Node (tenant VMs, service VMs)
Gateway Node (router / switch)
Service Node (e.g. physical firewall)

Figure 2: OpenContrail System Implementation

Computer Node

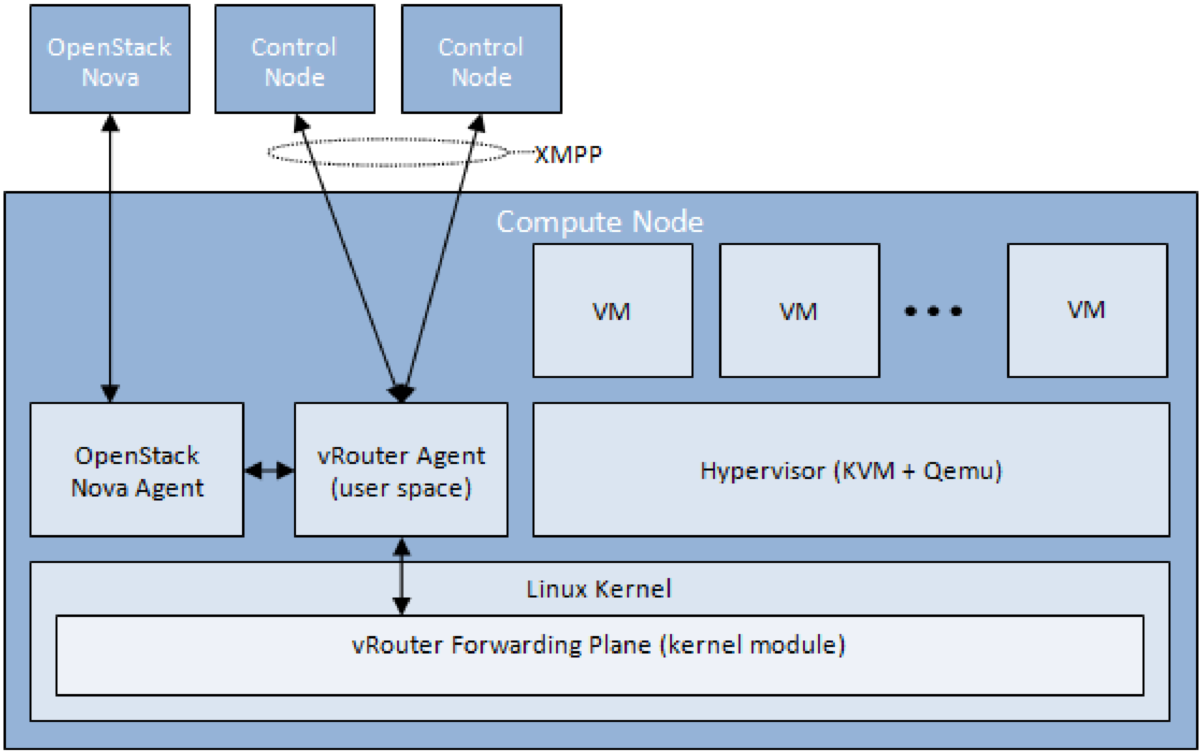


Figure 3: Internal Structure of a Compute node

重点关注 vrouter agent 和 vrouter forwarding plane

The vRouter agent is a user space process running inside Linux. It acts as the local, light-weight control plane and is responsible for the following functions:

- Exchanging control state such as routes with the Control nodes using XMPP.
- Receiving low-level configuration state such as routing instances and forwarding policy from the Control nodes using XMPP.
- Reporting analytics state such as logs, statistics, and events to the analytics nodes.
- Installing forwarding state into the forwarding plane.
- Discovering the existence and attributes of VMs in cooperation with the Nova agent.
- Applying forwarding policy for the first packet of each new flow and installing a flow entry in the flow table of the forwarding plane.
- Proxying DHCP, ARP, DNS, and MDNS.  Additional proxies may be added in the future.

## vRouter forwarding plane

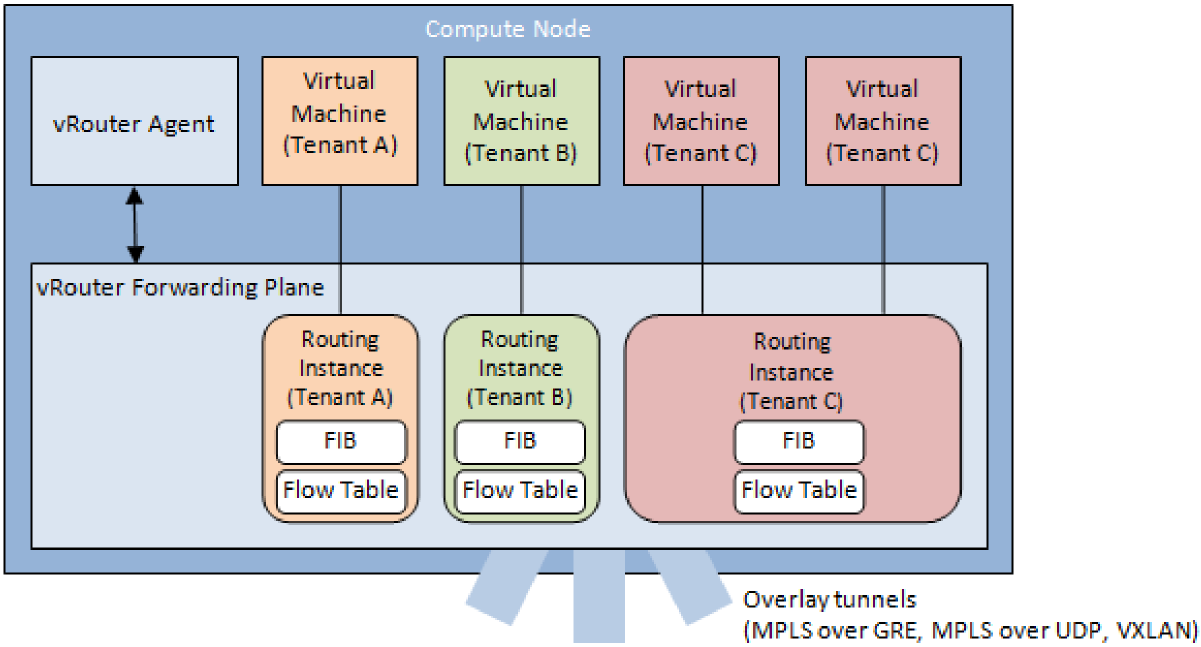

Compute Node
vRouter Agent
Virtual Machine (Tenant A)
Virtual Machine (Tenant B)
Virtual Machine (Tenant C)
Virtual Machine (Tenant C)
vRouter Forwarding Plane
Routing Instance (Tenant A)
FIB
Flow Table
Routing Instance (Tenant B)
FIB
Flow Table
Routing Instance (Tenant C)
FIB
Flow Table
Overlay tunnels
(MPLS over GRE, MPLS over UDP, VXLAN)

**Figure 4: vRouter Forwarding Plane**

The vRouter forwarding plane runs as a kernel loadable module in Linux and is responsible for the following functions:

- Encapsulating packets sent to the overlay network and decapsulating packets received from the overlay network.
- Assigning packets to a routing instance:
  - Packets received from the overlay network are assigned to a routing instance based on the MPLS label or Virtual Network Identifier (VNI).
  - Virtual interfaces to local virtual machines are bound to routing instances.
  - Doing a lookup of the destination address of the in the Forwarding Information Base (FIB) and forwarding the packet to the correct destination.  The routes may be layer-3 IP prefixes or layer-2 MAC addresses.
  - Optionally, applying forwarding policy using a flow table:
    - Match packets against the flow table and apply the flow actions.
    - Optionally, punt the packets for which no flow rule is found (i.e. the first packet of every flow) to the vRouter agent which then installs a rule in the flow table.
    - Punting certain packets such as DHCP, ARP, MDNS to the vRouter agent for proxying.

# Packets Forwarding detail

1.Layer 3 Unicast

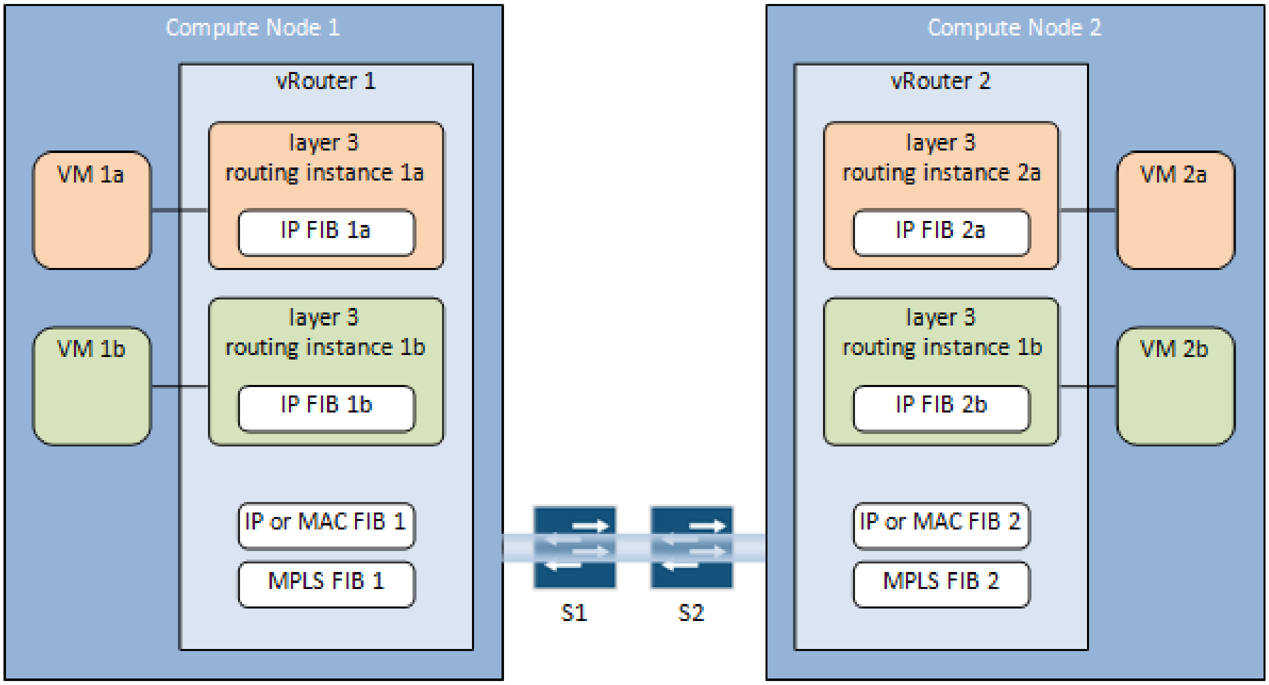


Figure 13: Data Plane: Layer 3 Unicast Forwarding Plane

# An application in VM 1a sends an IP packet with destination IP address VM 2a.

1. VM 1a has a default route pointing to a 169.254.x.x link-local address in routing instance 1a.
2. VM 1a sends an ARP request for that link local address. The ARP proxy in routing instance 1a responds to it.
3. VM 1a sends the IP packet to routing instance 1a.
4. IP FIB 1a on routing instance 1a contains a /32 route to each of the other VMs in the same virtual network including VM 2a. This route was installed using by the control node using XMPP. The next-hop of the route does the following:
   a. Imposes an MPLS label which was allocated by vRouter 2 for routing instance 2a.
   b. Impose a GRE header with the destination IP address of compute node 2.
5. vRouter 1 does a lookup of the new destination IP address of the encapsulated packet (which compute node 2) in global IP FIB 1.
6. vRouter 1 sends the encapsulated packet to compute node 2. How this happens exactly depends on whether the underlay network is a layer 2 switched network or a layer 3 routed network. This is described in detail below. For now we skip this part and assume the encapsulated packet makes it to compute node 2.
7. Compute node 2 receives the encapsulated packet and does an IP lookup in global IP FIB 2.

Since the outer destination IP address is local, it decapsulates the packet i.e. it removes the GRE header which exposes the MPLS header.

8. Compute node 2 does a lookup of the MPLS label in the global MPLS FIB 2 and find an entry which points to routing instance 2a. It decapsulates the packet i.e. it removes the MPLS header and injects the exposed IP packet into routing instance 2a.
9. Compute node 2 does a lookup of the exposed inner destination IP address in IP FIB 2a. It finds a route that points to the virtual interface connected to VM 2a.
10. Compute node 2 sends the packet to VM 2a.

Now we return to the part that we glossed over in step 7: how is the encapsulated packet forwarded across the underlay network.

If the underlay network is a layer 2 network then:

1. The outer source IP address (compute node 1) and the destination IP address (compute node 2) of the encapsulated packet are on the same subnet.
2. Compute node 1 sends an ARP request for IP address compute node 2. Compute node 2 sends an ARP reply with MAC address compute node 2. Note that there is typically no ARP proxying in the underlay.
3. The encapsulated packet is layer 2 switched from compute node 1 to compute node 2 based on the destination MAC address.

If the underlay network is a layer 3 network then:

1. The outer source IP address (compute node 1) and the destination IP address (compute node 2) of the encapsulated packet are on the different subnets.
2. All routers in the underlay network both the physical router (S1 and S2) and the virtual routers (vRouter 1 and vRouter 2) participate in some routing protocol such as OSPF.
3. The encapsulated packet is layer 3 routed from compute node 1 to compute node 2 based on the destination IP address. Equal Cost Multi Path (ECMP) allows multiple parallel paths to be used. For this reason the VXLAN encapsulation include entropy in the source port of the UDP packet.

# 2.Layer 2 Unicast

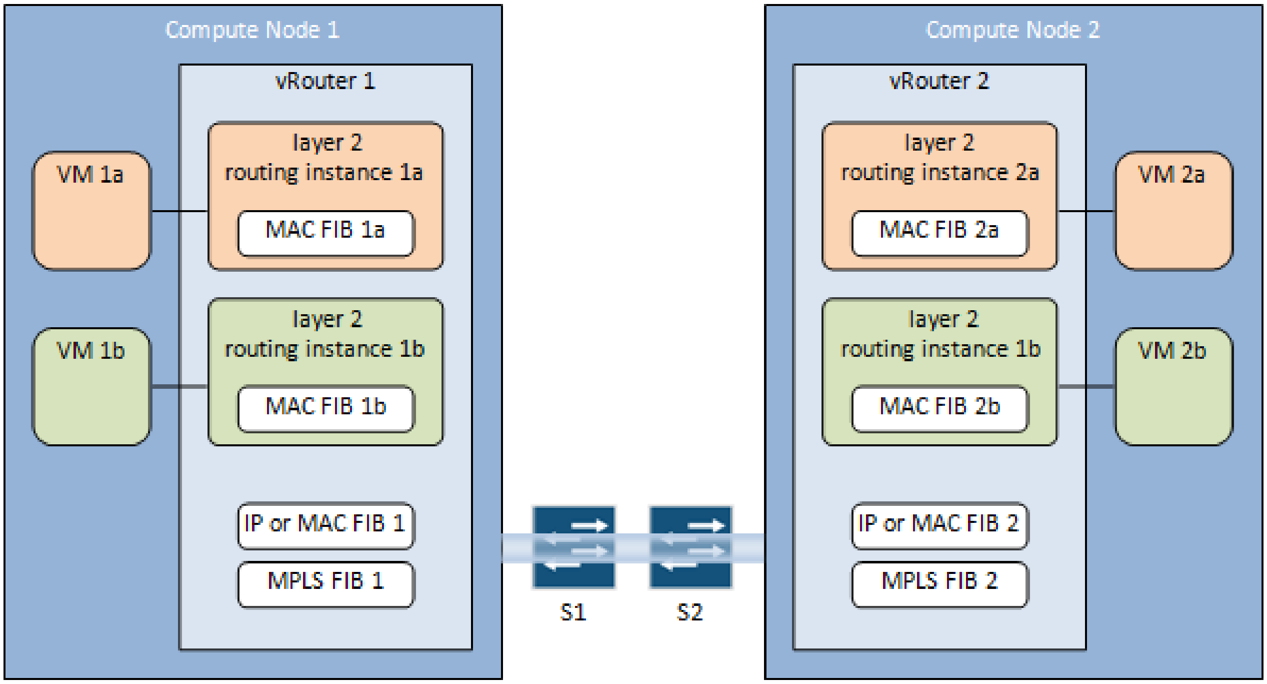


Figure 14: Data Plane: Layer 2 Unicast

Forwarding for L2 overlays works exactly the same as forwarding for L3 overlays as described in the previous section, except that:

- The forwarding tables in the routing instances contain MAC addresses instead of IP prefixes.
- ARP is not used in the overlay (but it is used in the underlay).

## 3.Fallback Switching

OpenContrail supports a hybrid mode where a virtual network is both a L2 and a L3 overlay simultaneously. In this case the routing instances on the vRouters have both an IP FIB and a MAC FIB. For every packet, the vRouter first does a lookup in the IP FIB. If the IP FIB contains a matching route, it is used for forwarding the packet. If the IP FIB does not contain a matching route, the vRouter does a lookup in the MAC FIB – hence the name fallback switching.

## 3.Layer 3 Multicast

# NOC 方案