Linux 上虚拟网络与真实网络的映射

使用 Linux 上的网络设备模拟真实网络

随着云计算技术的发展，如何以类似物理网络的方式分割虚拟网络成为热点，物理网络也引入了更多支持虚拟化的网络技术，使得问题更加复杂。本文将阐述在 Linux 上如何模拟出传统网络及支持虚拟化技术的网络，并介绍其原理。

夏文超，软件工程师，现从事OpenVirtualiztion 方面的工作。您可以通过 developerWorks 社区与夏文超进行交流。

2013 年 12 月 16 日

虚拟化环境中的网络问题

在提供 IaaS 服务的云计算环境中，每个用户都能得到一个虚拟的计算机，而这些虚拟机器以密集的方式运行在后台服务器集群中。虚拟机的一个特点是提供给用户类似于物理机器的体验，而现实世界中的物理机器能通过各种网络拓扑结构组网。如何在虚拟环境中方便快捷地创建和现实中一样的网络，成为一个新的挑战。

图 1.物理网络映射问题例子

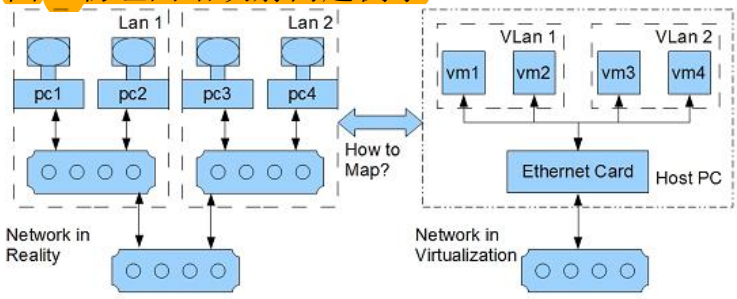


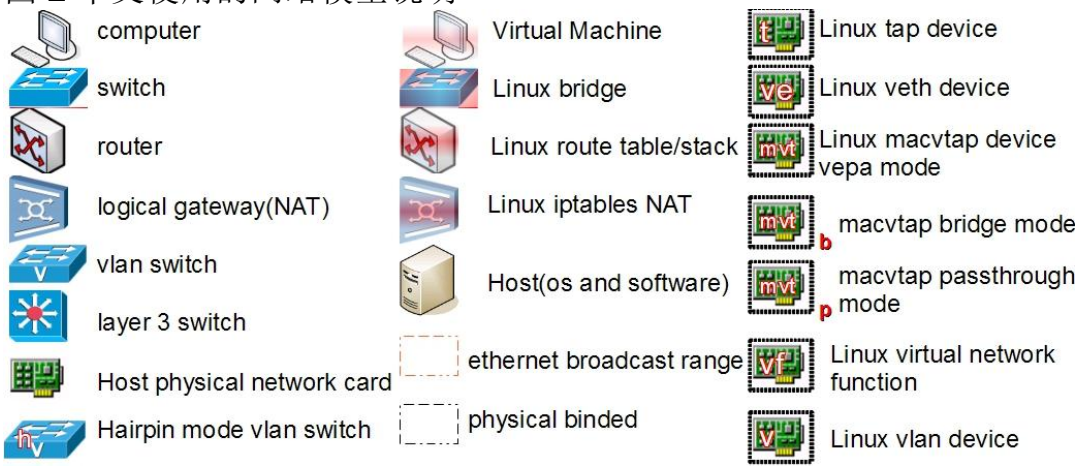
图 1 为一个网络映射问题的例子。图中左边是现实世界中一个常见的网络环境：四台 PC 通过各自的物理网卡组成了两个子网，两个子网中的 PC 默认情况下是不能通讯的，也就是说他们被物理隔离了。图 1 的右边显示了虚拟化环境下的情景，四个虚拟机同时运行在一个物理主机上，并且需要象图 1 左边的真实环境一样划分出两个子网并隔离。如何才能做到这一点，或者说如何简单方便的创建出和图 1 左边部分类似的网络环境，成为虚拟化里必须要解决的一个问题。

虚拟化环境中模拟网络的主要方法

为方便理解，本文把虚拟化环境中模拟现实网络的方法分为两种：使用传统网络技术，或使用虚拟化网络扩展技术。传统网络技术主要指在虚拟化技术流行以前，现实世界中已经存在的以太网络，包括传统 IP 网络、802.1Q VLAN 网络，对它们 Linux 已经有良好支持，用户可以配置这些 Linux 设备以完成对现实网络的模拟。虚拟化网络扩展技术主要指为应对云计算与虚拟化环境带来的挑战而新出现的网络技术，包括 802.1Qbg 和 802.1Qbh 网络。

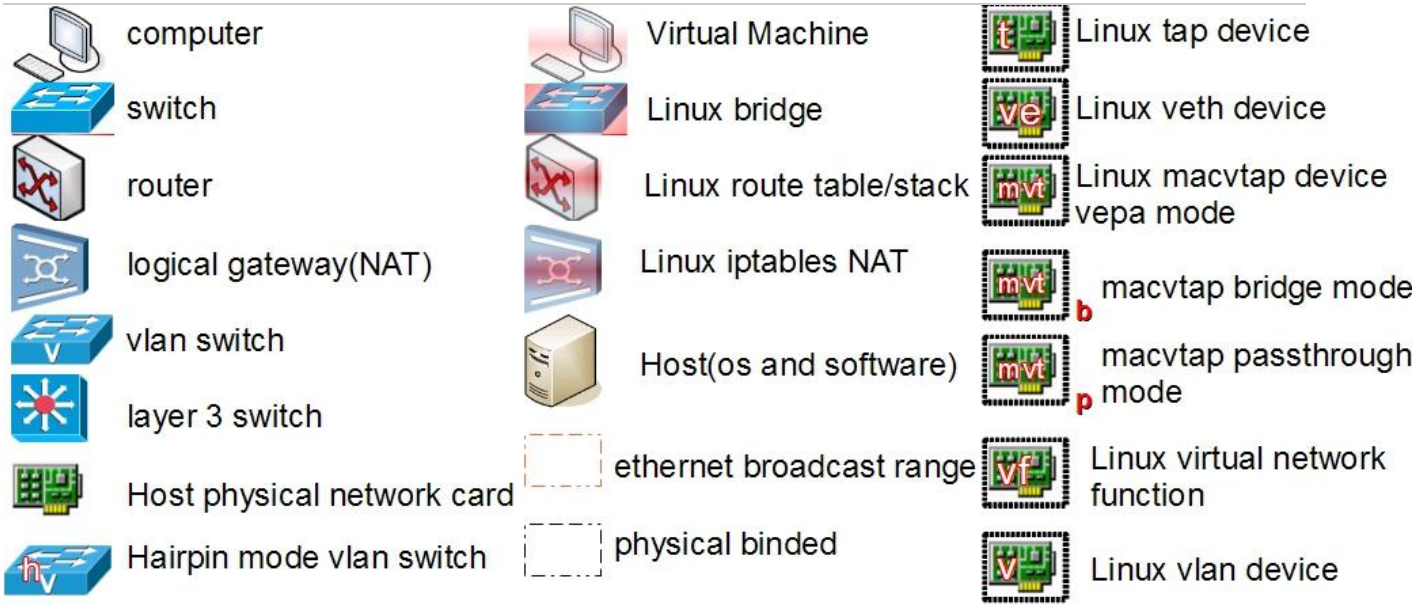
虚拟化环境中网络模型说明

图 **2** 本文使用的网络模型说明



[点击查看大图]

图 **2** 本文使用的网络模型说明



为方便阅读，上图列出了本文将使用的网络元素。图中左列表示现实世界中存在的网络元素，分别为电脑终端、二层交换机、路由器、网关、支持 802.1Q VLAN 的交换机、三层交换机、物理网卡、支持 Hairpin 模式的交换机。图中中列为虚拟化环境中的元素，分别为虚拟机，Linux Bridge、Linux 路由表、Linux iptables、Host 主机。棕色虚线框表示以太网广播域，黑色虚线框表示物理捆绑关系。图中右列为 Linux 系统里的网络设备模型，分别为 TAP 设备、VETH 设备、工作在 VEPA 模式的 MACVLAN 设备、工作在 Bridge 模式的 MACVLAN 设备、工作在 Passthrough 模式的 MACVLAN 设备、SRIOV 的虚拟 VF 设备、VLAN 设备，下文将有对它们的简介。

使用传统网络技术模拟现实网络

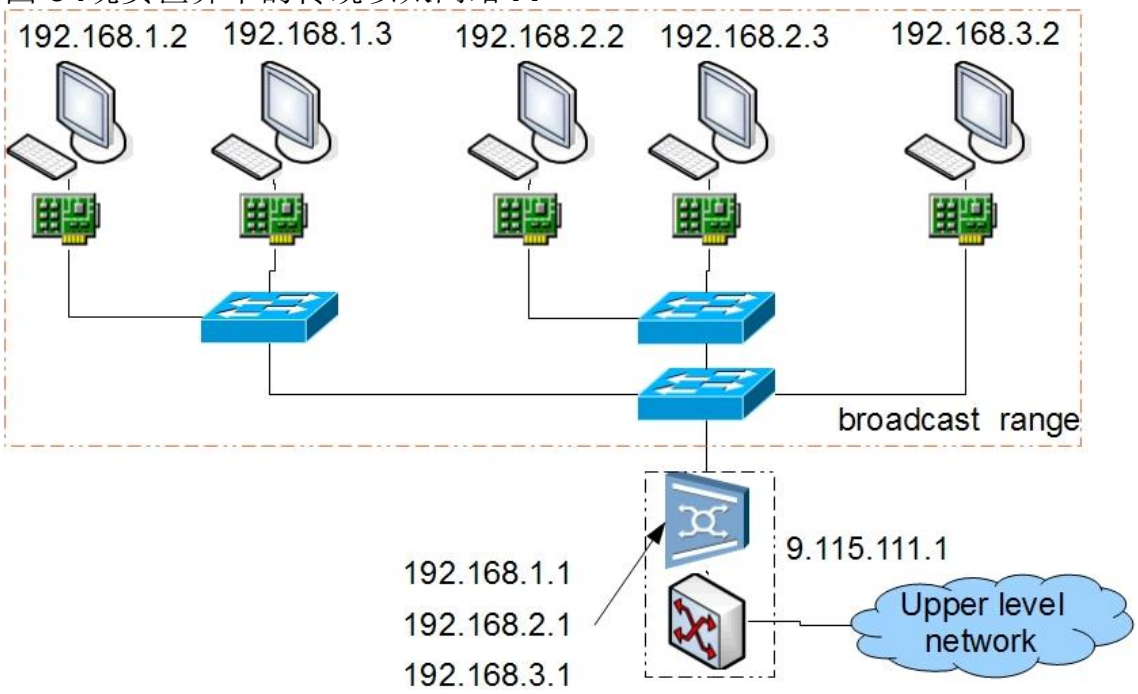
Linux Host 侧使用的网络元素简介

Linux 主要使用以下三种设备模型：Bridge、TAP、VETH、VLAN。Bridge 设备是基于内核实现的二层数据交换设备，其作用类似于现实世界中的二级交换机。TAP 设备

是一种工作在二层协议的点对点网络设备，每一个 TAP 设备都有一个对应的 Linux 字符设备，用户程序可以通过对字符设备的读写操作，完成与 Linux 内核网络协议栈的数据交换工作，在虚拟化环境中经常被模拟器使用。VETH 设备是一种成对出现的点对点网络设备，从一段输入的数据会从另一端改变方向输出，通常用于改变数据方向，或连接其它网络设备。VLAN 设备是以母子关系出现的一组设备，是 Linux 里对 802.1.Q VLAN 技术的部分实现，主要完成对 802.1.Q VLAN Tag 的处理。

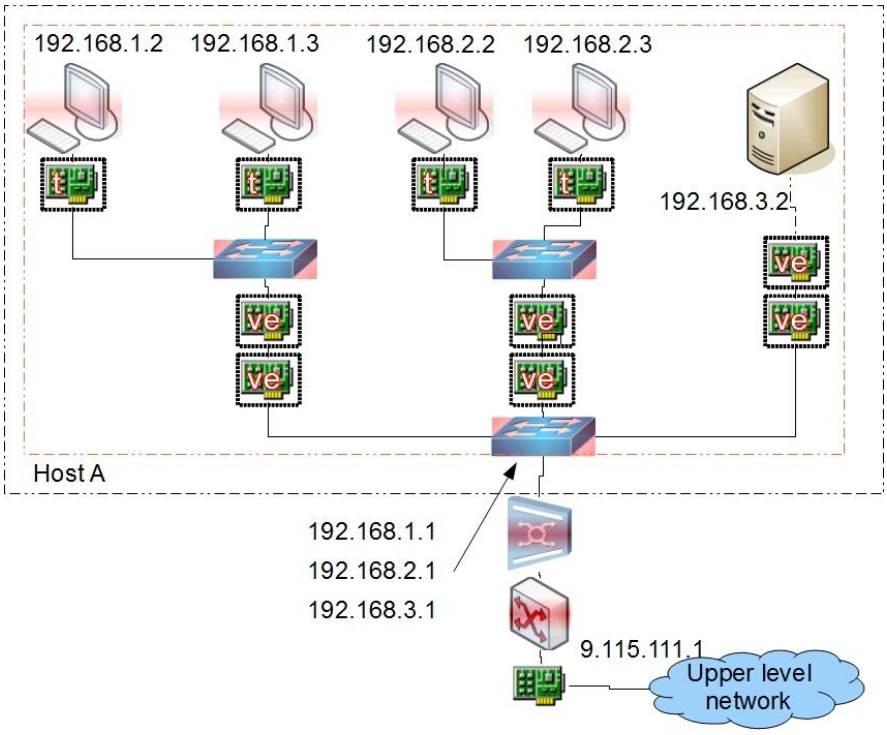
模拟传统以太网

图 **3** .现实世界中的传统以太网络 **A**



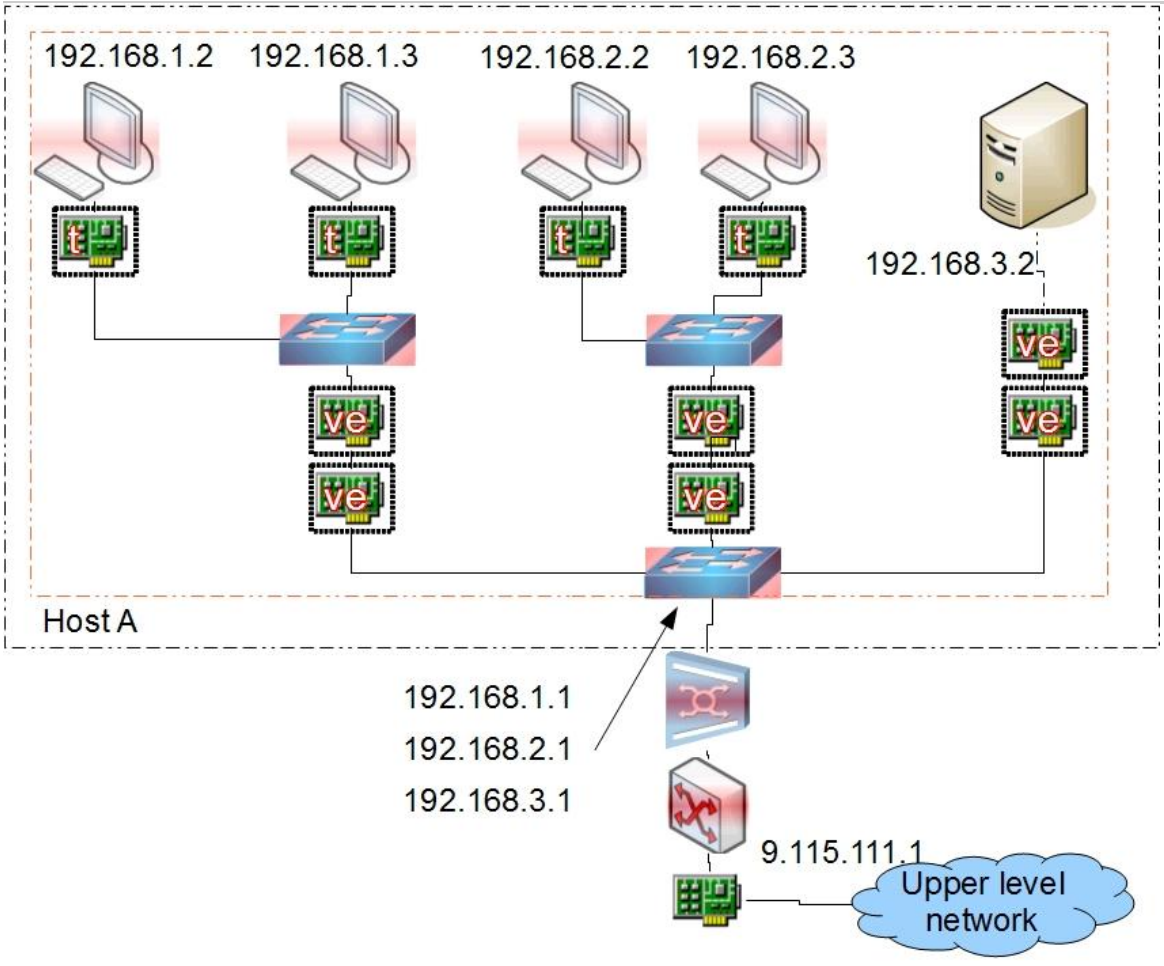
上图为一个典型的传统以太网结构：5 个终端机器通过各自的网卡连接接入层的交换机，交换机再通过汇聚端口连接第二级交换机，进而接入作为网关的路由器，路由器通过 NAT(Net Address Translate)转发数据到外界网络，从而构成一个封闭但是可以连接外网，并且只占有一个公共 IP 的私网环境。由于所有的终端都在同一个二级交换机下，根据以太网协议，二层的广播报文将在整个网络内传遍，构成了潜在的广播风暴风险。类似的网络结构广泛存在于公司、小区、家庭用户中。

图 **4** .虚拟网络 **A_V0**



点击查看大图

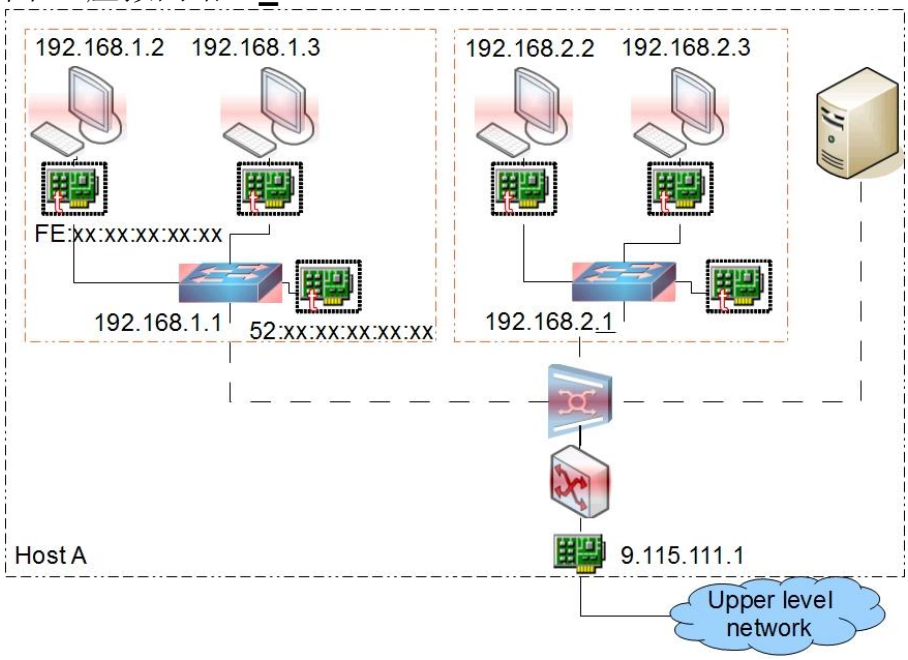
图 **4** .虚拟网络 **A_V0**



上图所示为虚拟化情况下，对网络 A 的一种比较准确的模拟。四台虚拟机通过 TAP 设备连接到接入层 Bridge 设备，接入层 Bridge 设备通过一对 VETH 设备连接到二级 Bridge 设备，主机通过一对 VETH 设备接入二级 Bridge 设备。二级 Bridge 设备进一

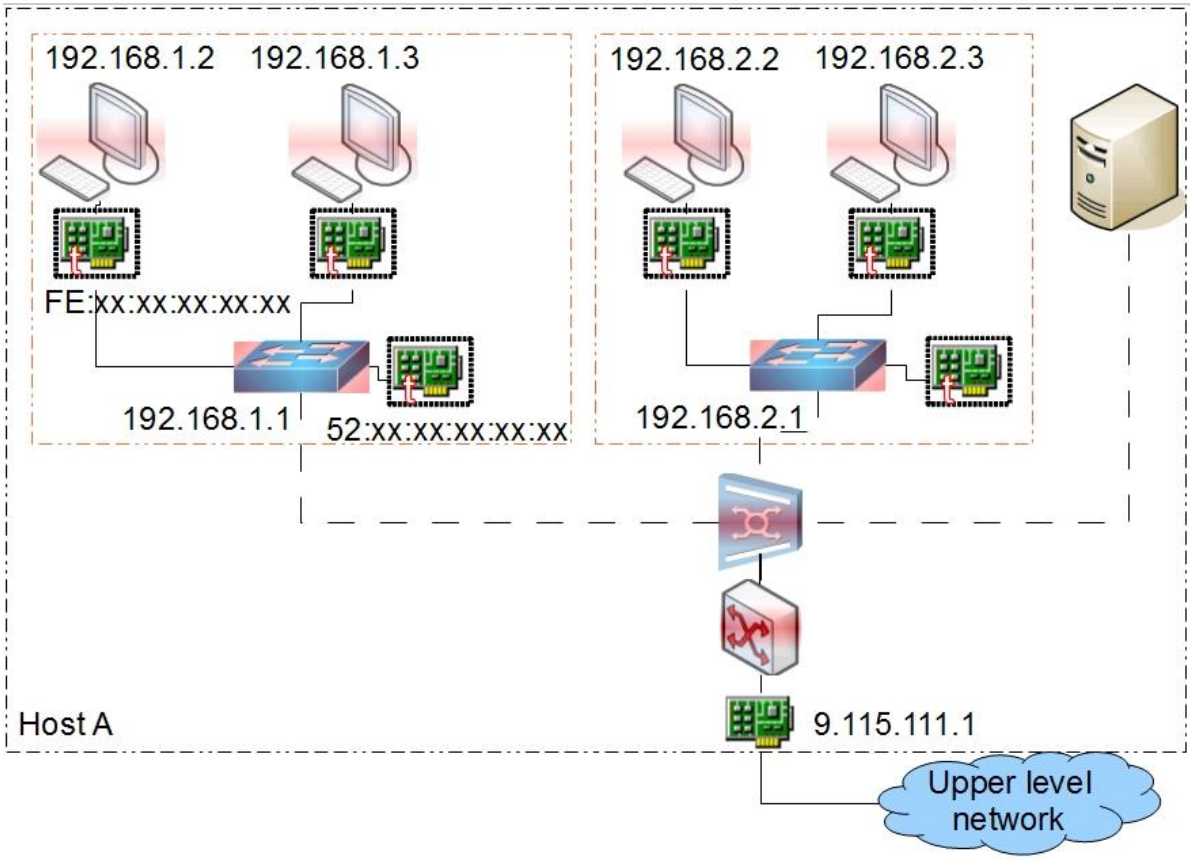
步通过 Linux 路由表，IP Tables 与物理网卡形成数据转发关系，最终和外部物理网络连接。此图中的元素与网络 A 中的元素近乎一一对应，Bridge 相当于现实世界中的二层交换机，VETH 设备相当于连接 Bridge 的网线，虚拟机看到的网络和网络 A 的物理机一样，广播域包括所有虚拟用户终端。但在通常情况下，虚拟机不一定需要二级的 Bridge 同时存在，它仅仅需要数据的转发功能，因此为了提高效率一般改变虚拟网络配置只保留最核心的功能。

图 **5** .虚拟网络 **A_V1**



点击查看大图

图 **5** .虚拟网络 **A_V1**

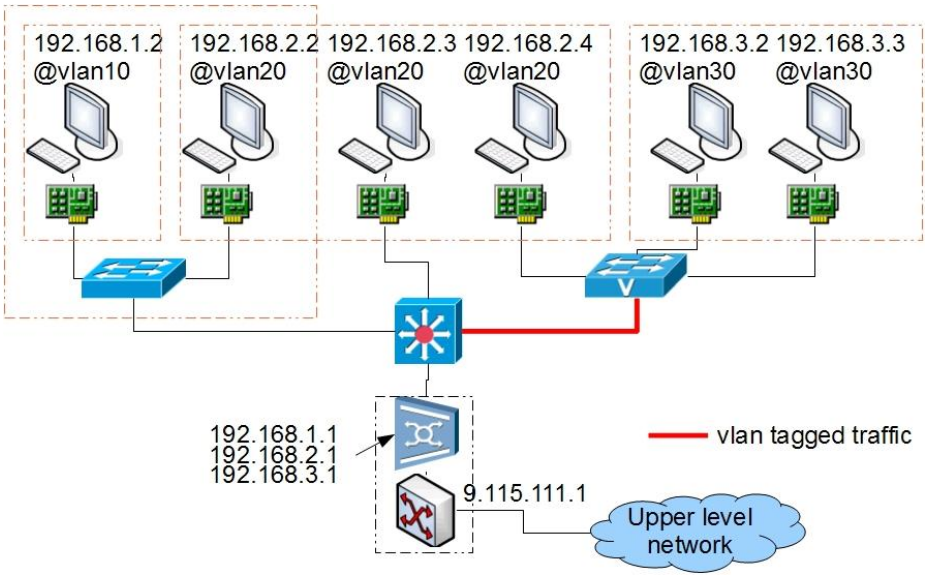


上图为虚拟化环境中一种常用的网络配置，对比网络 A_V0 有如下变化：不再一一映射网络 A，省去二级 Bridge，省去 VETH 设备。这种情况下，虚拟机仍然能通过虚拟网关上网，只不过探测不到二级 Bridge 的存在。由于效率较高，这种一级 Bridge 加 NAT 的网络被选为 Libvirt 的默认虚拟网络。图中的 Bridge 设备总是连接有一个 MAC 为 52:xx:xx:xx:xx:xx 的 TAP 设备，原因是 Linux 内核里 Bridge 的实现有一个缺陷：当加入的设备 MAC 为最小 MAC 时，MAC 学习会打断 Bridge 的工作，因此事先创建一个 MAC 值很小的设备 51:xx:xx:xx:xx:xx 绕过此问题。图中由于存在两个子网（192.168.1.0 网段与 192.168.2.0 网段），因此使用了两个 Bridge 设备以区分出两个广播域，和网络 A 产生了区别，这在没有 802.1Q VLAN 的情况下不可避免。

模拟 **802.1Q VLAN** 以太网

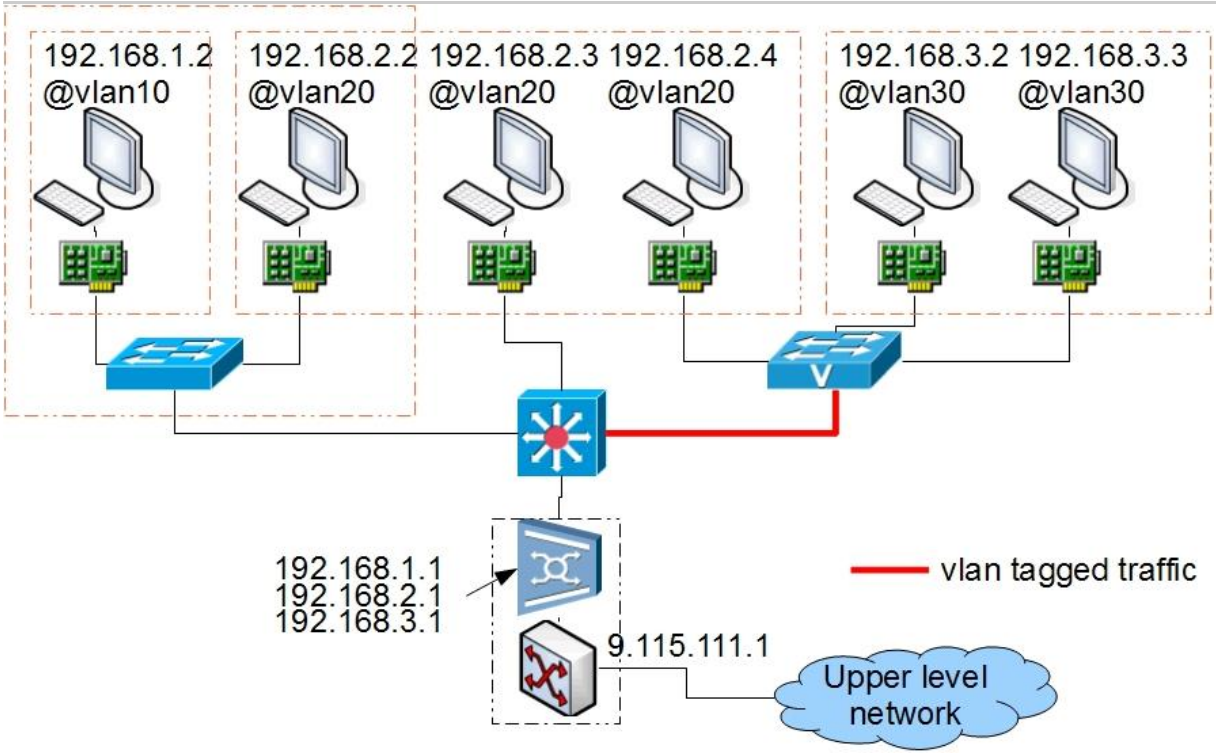
在虚拟化技术流行之间，通讯业界已制定了 802.1Q VLAN 标准，以解决复杂网络环境下广播风暴域的问题。使用 802.1Q VLAN 技术，可以把逻辑上的子网和物理上的子网分割开来，即物理上连接在同一交换机上的终端可以属于不同逻辑子网，处于不同逻辑子网的终端相互隔离，从而解决了前文描述的广播域混乱的问题。

图 **6** .现实世界中的 **802.1Q VLAN** 以太网络 **B**



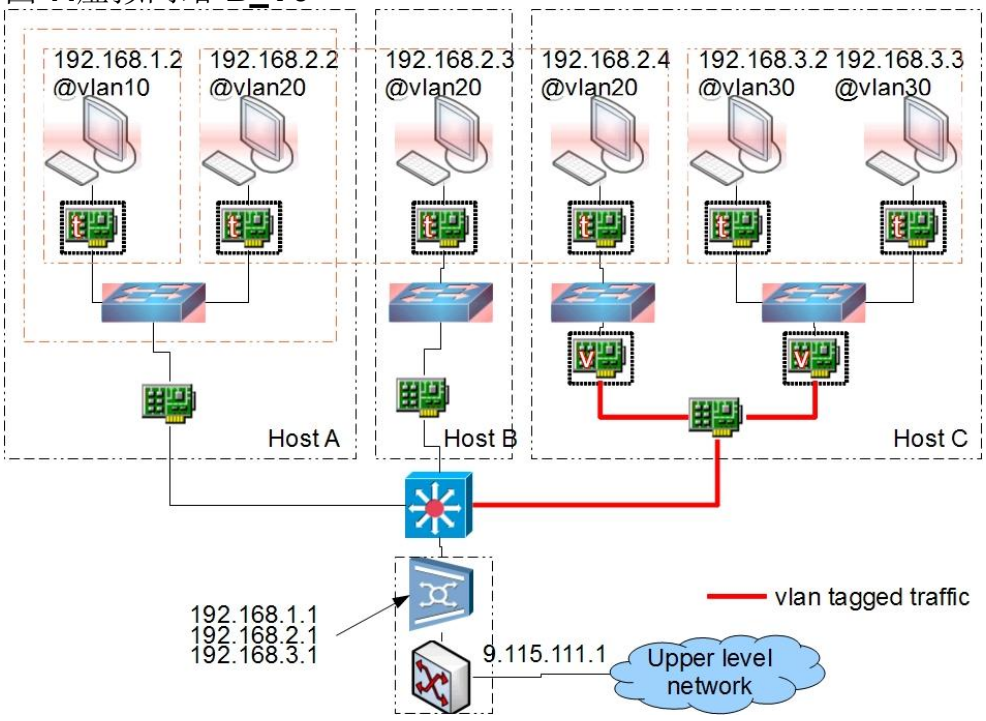
点击查看大图

图 **6** .现实世界中的 **802.1Q VLAN** 以太网络 **B**



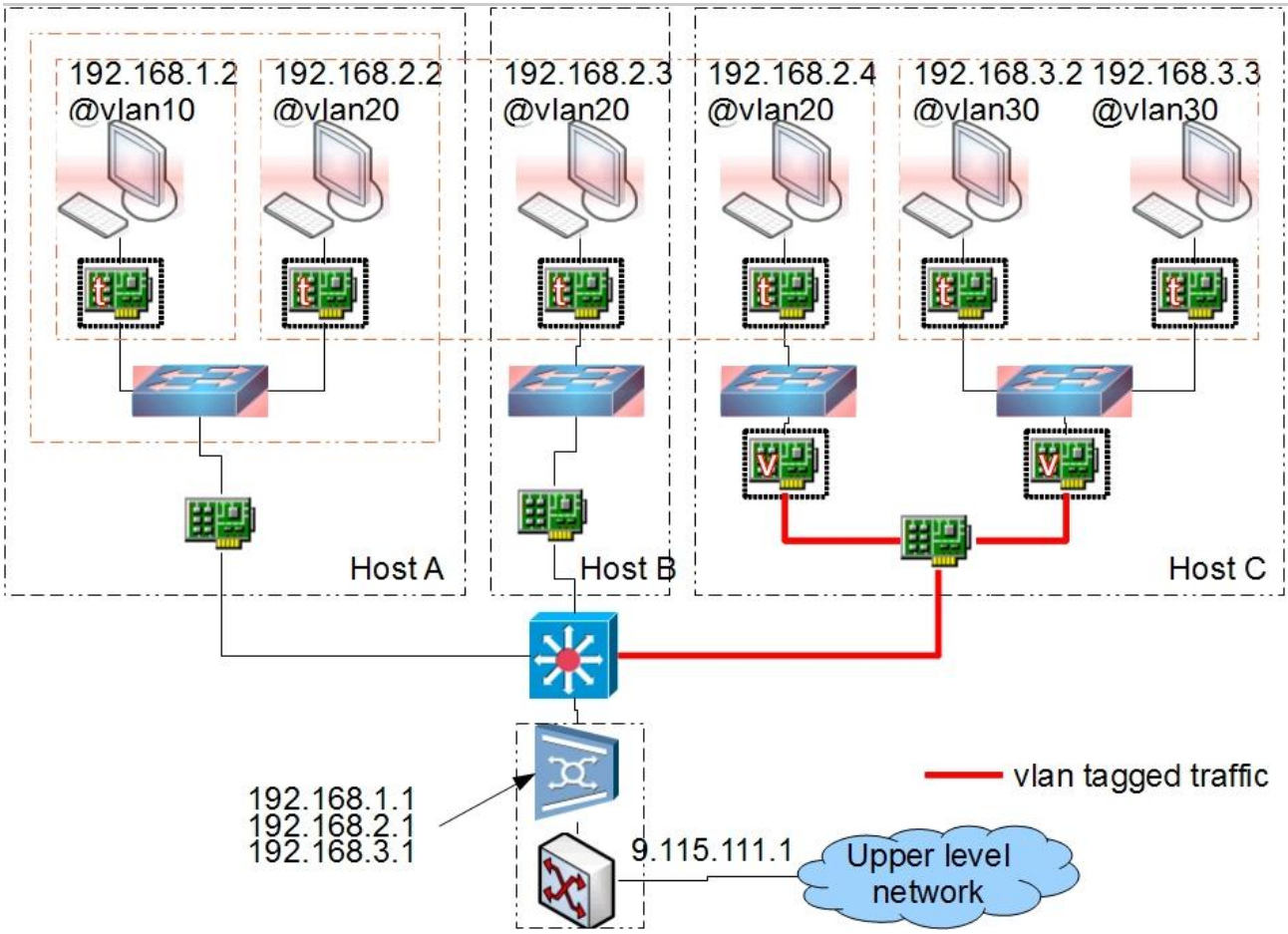
上图所示为一个现实世界中的 802.1Q VLAN 网络。六台电脑终端通过一级交换机接入网络，分属 VLAN 10、VLAN 20、VLAN 30。做为例子，图中左侧的交换机不支持 802.1Q VLAN，导致其连接的两台终端处于一个广播域中，尽管它们属于不同子网。作为对比，图中右侧的交换机支持 802.1Q VLAN，通过正确配置正确切割了子网的广播域，从而隔离了分属不同网段的终端。在连接外网之间，需要一个支持 802.1Q VLAN 的三层交换机，在进行数据外发时剥离 VLAN Tag，收到数据时根据 IP 信息转发到正确的 VLAN 子网。路由器根据 IP 信息进行 NAT 转换最终连接外网。

图 **7.**虚拟网络 **B_V0**



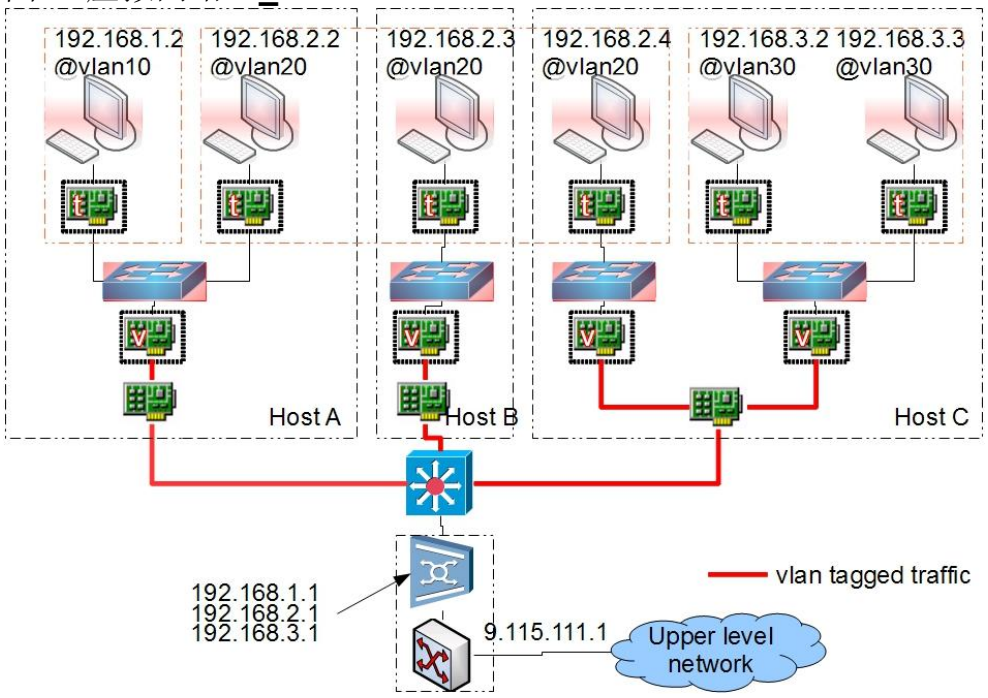
点击查看大图

图 **7.**虚拟网络 **B_V0**



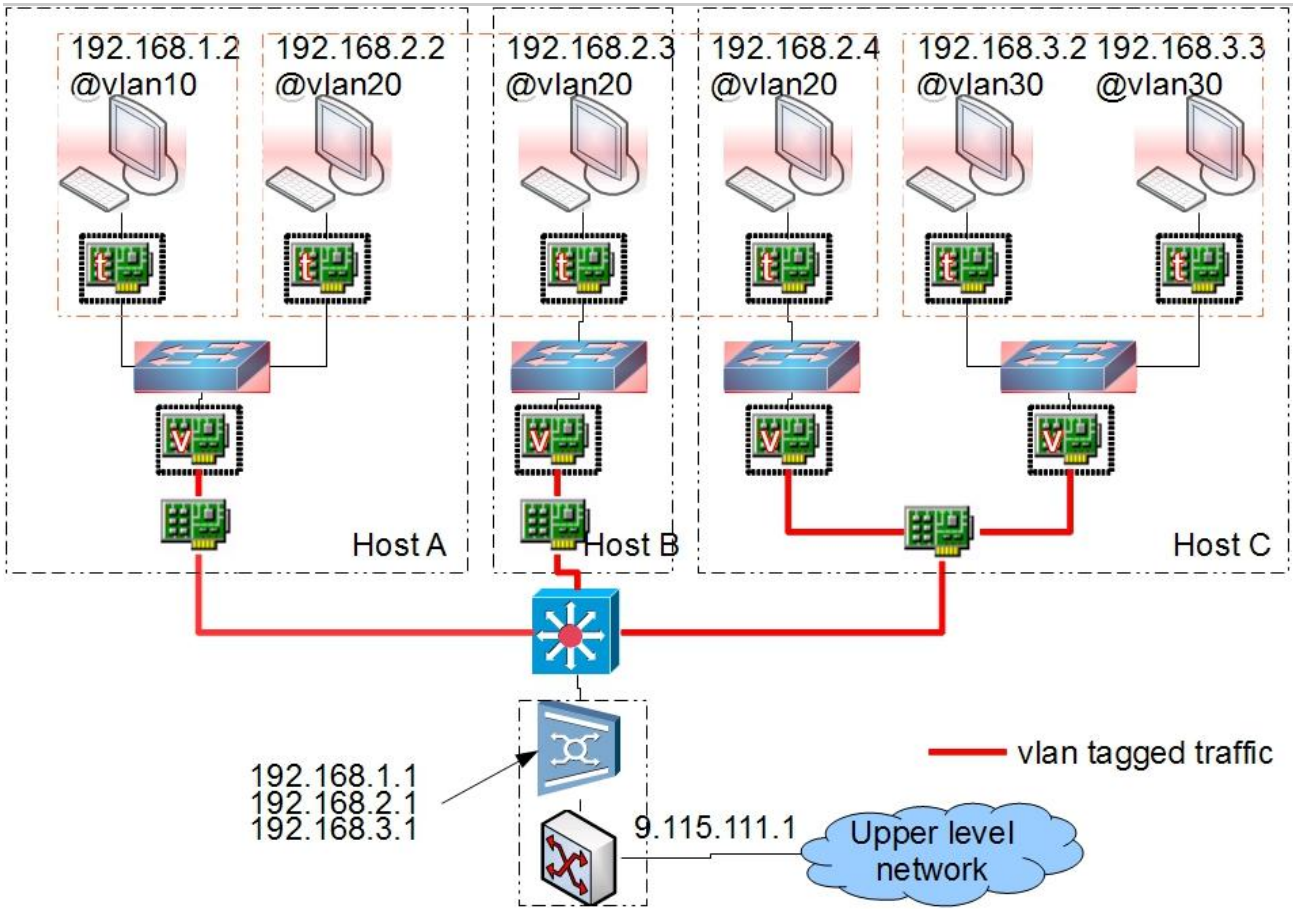
上图能在虚拟化的条件下对网络 B 进行较精确的模拟，六台虚拟机将和网络 B 中的真实 PC 看到一样的网络环境。Host C 上的 Bridge、VLAN Device 与物理网卡共同完成了网络 B 中的支持 802.1Q VLAN 的一级交换机的功能，从而隔离逻辑子网。Host B 上的 Bridge 仅仅起连接物理网卡与虚拟机的作用。Host A 上的 Bridge 相当于普通交换机，和网络 B 一样存在广播域交叉问题。

图 **8** .虚拟网络 **B_V1**



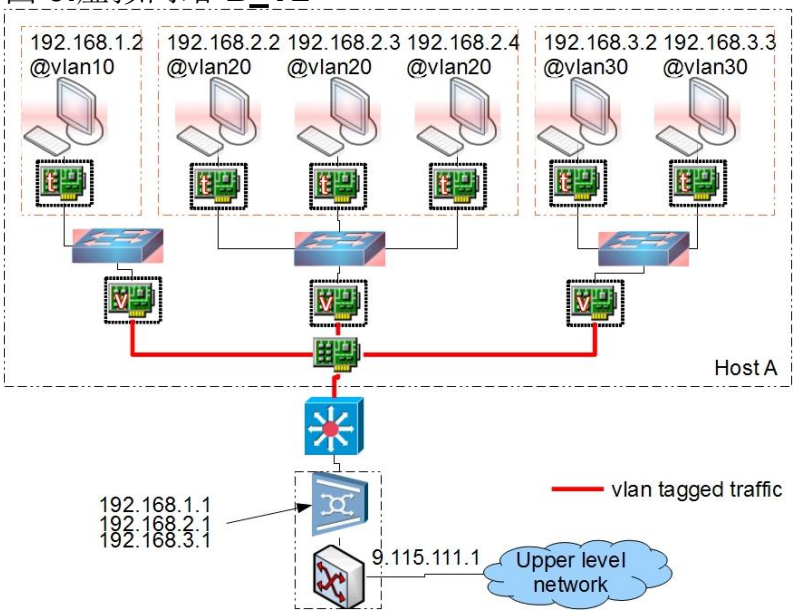
点击查看大图

图 **8** .虚拟网络 **B_V1**



上图通过在 Host A 与 Host B 上引入 VLAN 设备，解决了 B_V0 中存在的广播域交叉问题，虚拟机已经能正确用使用相互隔离的子网。大多数情况下虚拟机并不关心 Bridge 以上部分的网络情况，只要求正确隔离逻辑子网，并且他们可以运行在同一个 Host 上，因此常把网络加以变换简化。

图 **9.**虚拟网络 **B_V2**



上图表示了一个经常在虚拟化中使用的 802.1Q VLAN 网络。对于所有虚拟机来说，它们处于和网络 B 相同的逻辑子网中，并且由于 VLAN Device 的引入，避免了 B 中的 VLAN 10 与 VLAN 20 的广播域交叉问题。多个虚拟机需要接入同一个 VLAN 时，只需使用一个 Bridge 来扩展，而不必像现实世界中的交换机那样使用多级交换机进

行数据汇聚，因为 Bridge 拥有近乎无限多个端口用于连接其他设备，没有物理端口数限制。物理网卡输出的将是带 VLAN Tag 的数据，和网络 B 一样，需要一个支持 802.1Q VLAN 的三层交换机进行处理。

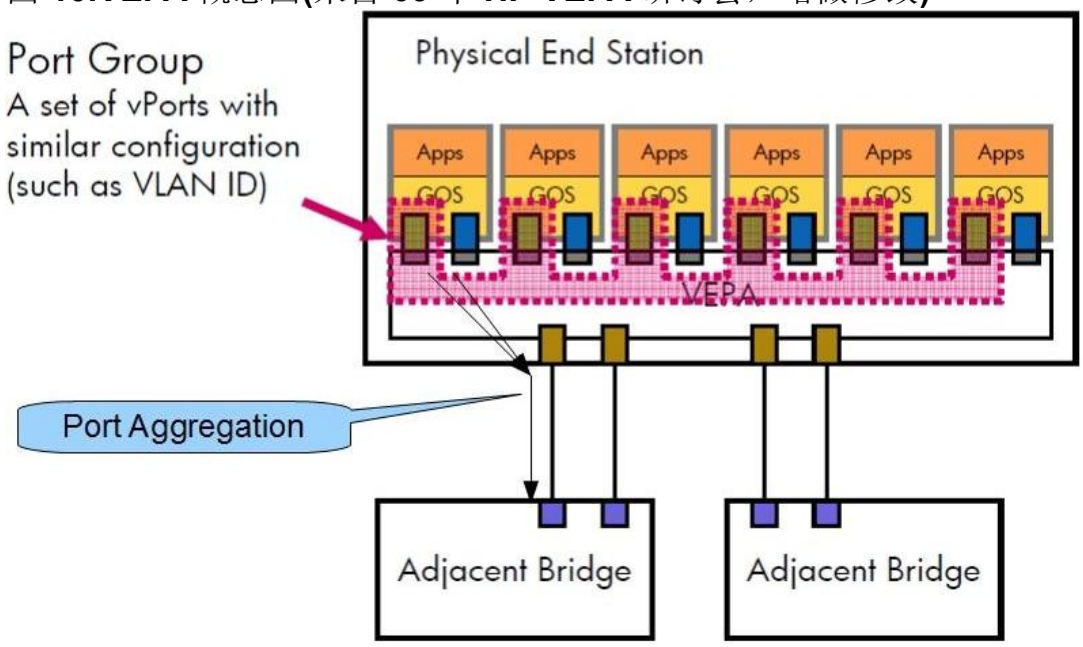
使用虚拟化网络扩展技术模拟现实网络

网络标准侧的扩展技术

针对云计算中的复杂网络问题，业界主要提出了两种扩展技术标准：802.1Qbg 与 802.1Qbh。802.1Qbh Bridge Port Extension 主要由 Vmware 与 Cisco 提出，尝试从接入层到汇聚层提供一个完整的虚拟化网络解决方案，尽可能达到软件定义一个可控网络的目的。它扩展了传统的网络协议，因此需要新的网络设备支持，成本较高。802.1Qbg Edge Virtual Bridging (EVB) 主要由 HP 等公司联合提出，尝试以较低成本利用现有设备改进软件模拟的网络。本文主要针对后者做解析。

802.1Qbg 的一个核心概念是 VEPA（Virtual Ethernet Port Aggregator ），简单来说它通过端口汇聚和数据分类转发，把 Host 上原来由 CPU 和软件来做的网络处理工作转移到一级交换机上，减少 Host CPU 负载，同时使得在一级的交换机上做虚拟机网络流量监控成为可能，从而更清晰地分割服务器与网络设备的工作范围，方便系统的管理。

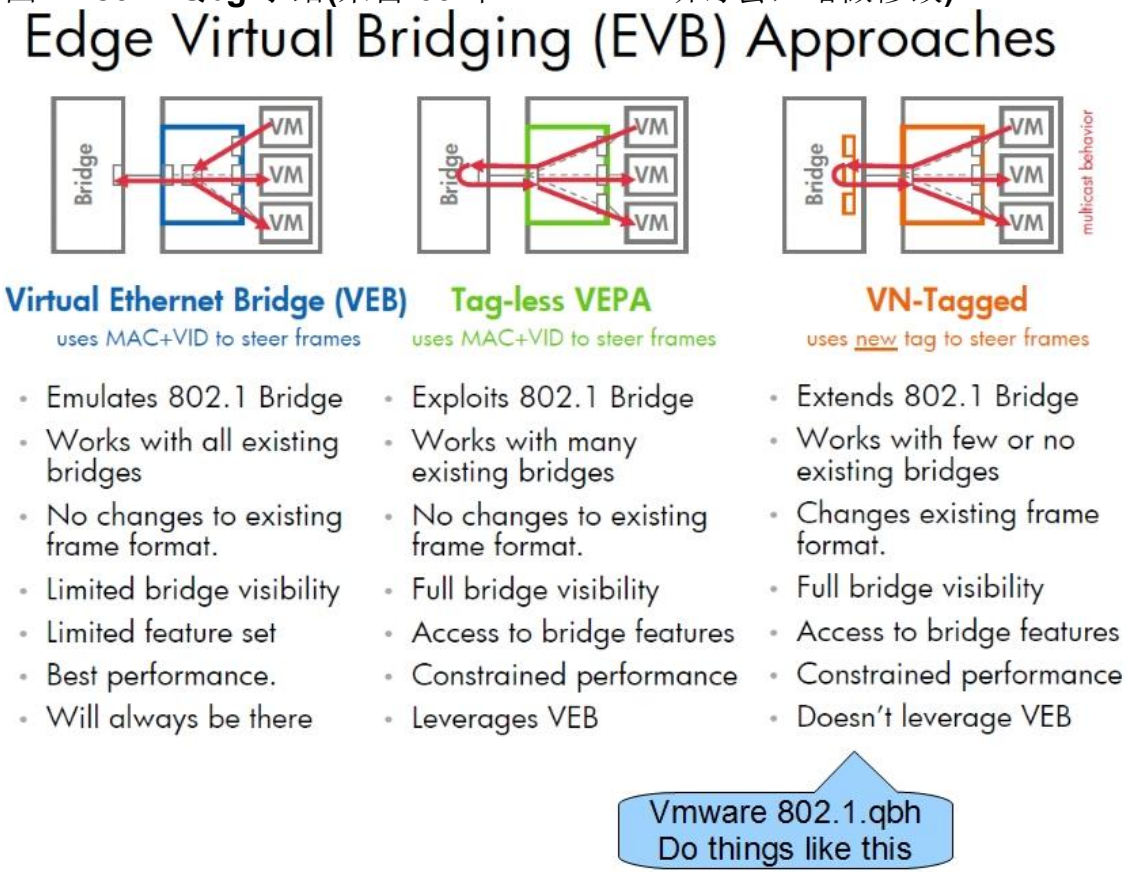
图 **10.VEPA** 概念图**(**来自 **09** 年 **HP VEPA** 研讨会，略做修改**)**



上图显示了 VEPA 中的基本概念：在物理终端上，即虚拟机运行的 Host 上，需要一个设备将虚拟端口根据一定的规则进行分组，完成端口分组功能(Port Group)。同时这个设备能够对外抽象出被分为一组的端口，将属于同一组端口的数据一起投递出去，完成端口汇聚功能(Port Aggregation)。图中画出了虚拟端口中的数据流向：所有来自虚拟端口的数据将和同组数据汇聚后投递到临近的一级交换机上，物理终端不再进行二层协议解析处理。同一物理终端里的虚拟端口之间的通讯，也必须通过一级交换机转发回来，而不能走捷径在物理终端内部进行转发，会增加一些一级交换机的流量负载。这样做的好处是网络处理的任务重新回到了专用网络设备端，同时所有的虚拟机网络流量变的对网络设备透明，方便网络管理员使用专用网络设备进行管控，不

再与 Host Server 牵扯不清。需要注意的是，VEPA 模式只能用在接入层的一级交换机上，网络里不能同时存在两层 VEPA 设备，顾称之为边缘虚拟化。

图 **11.802.1Qbg** 小结**(**来自 **09** 年 **HP VEPA** 研讨会，略做修改**)**

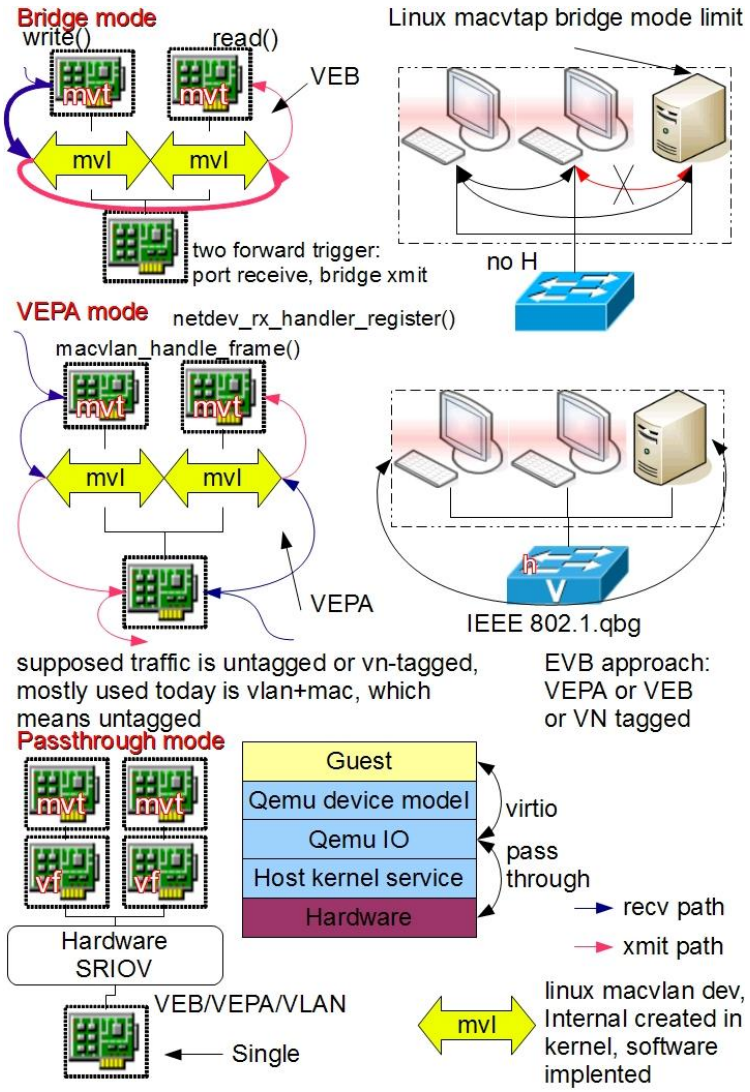


上图为 802.1Qbg 技术总结图：802.1Qbg 也在不断发展，VEB(Virtual Ethernet Bridge)表示虚拟机接入的一级数据交换功能模块，在 Linux 里可以看成用 Bridge 设备提供的 Host 内数据交换功能。Tag-less VEPA 为前文描述的 VEPA 数据流导出模式。这两种模式下由于通讯协议没有被修改，因此可以利用现有设备以很低的成本实现，其中 VEPA 模式只要刷新现有的交换机程序使其支持 Hairpin 模式完成数据的回传即可。作为长期解决方案，802.1Qbg 计划支持 VN-tagged 模式，即扩展通信协议使用新的 Tag 来标记数据。和 802.1Qbh 一样，这必将需要新的硬件支持，带来成本的上升。

Linux Host 侧的扩展技术

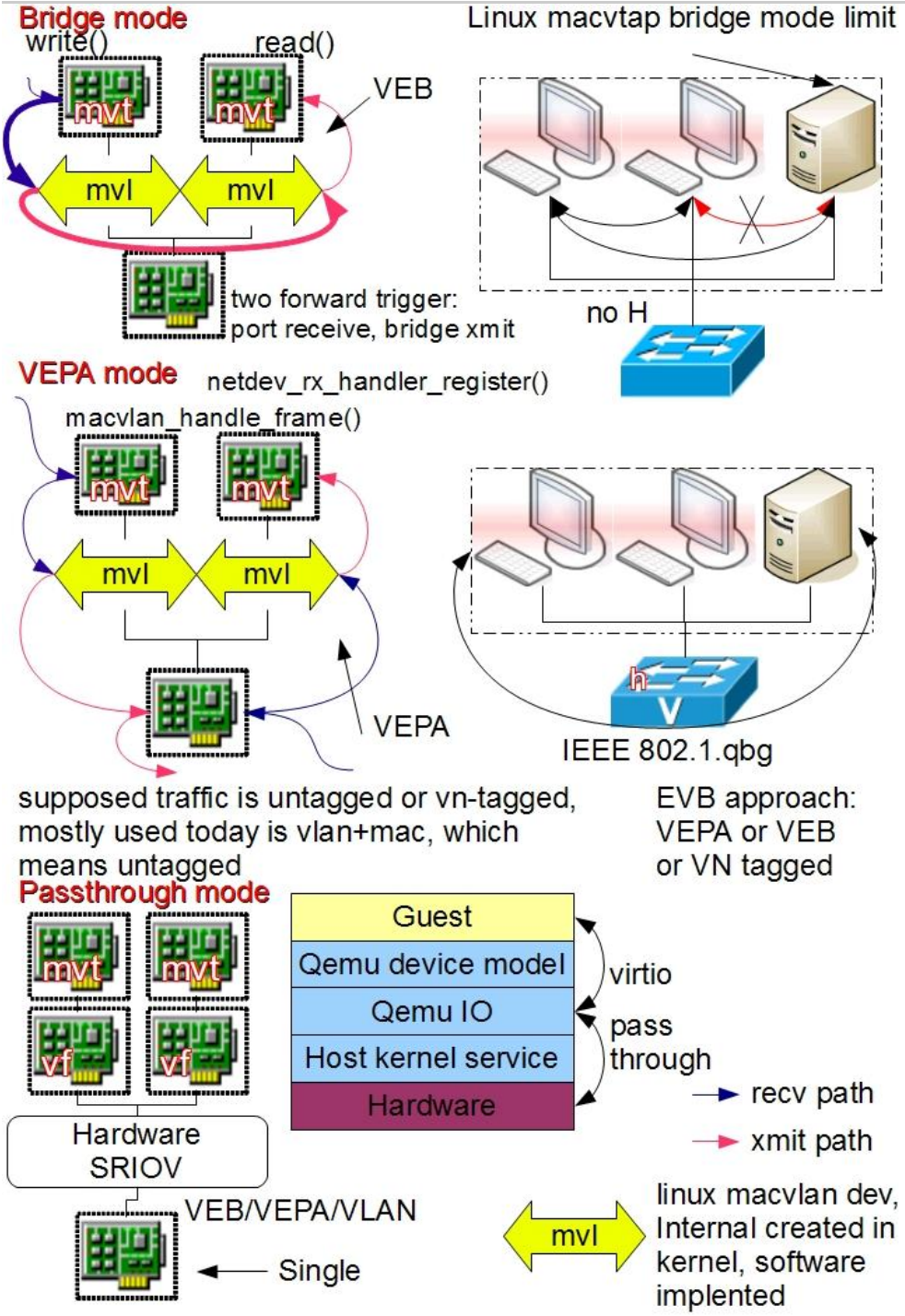
为支持新的虚拟化网络技术，Linux 引入了新的网络设备模型：MACVTAP。MACVTAP 的实现基于传统的 MACVLAN。和 TAP 设备一样，每一个 MACVTAP 设备拥有一个对应的 Linux 字符设备，并拥有和 TAP 设备一样的 IOCTL 接口，因此能直接被 KVM/Qemu 使用，方便地完成网络数据交换工作。引入 MACVTAP 设备的目标是：简化虚拟化环境中的交换网络，代替传统的 Linux TAP 设备加 Bridge 设备组合，同时支持新的虚拟化网络技术，如 802.1 Qbg。

图 **12 Linux MACVTAP** 设备原理



点击查看大图

图 **12 Linux MACVTAP** 设备原理

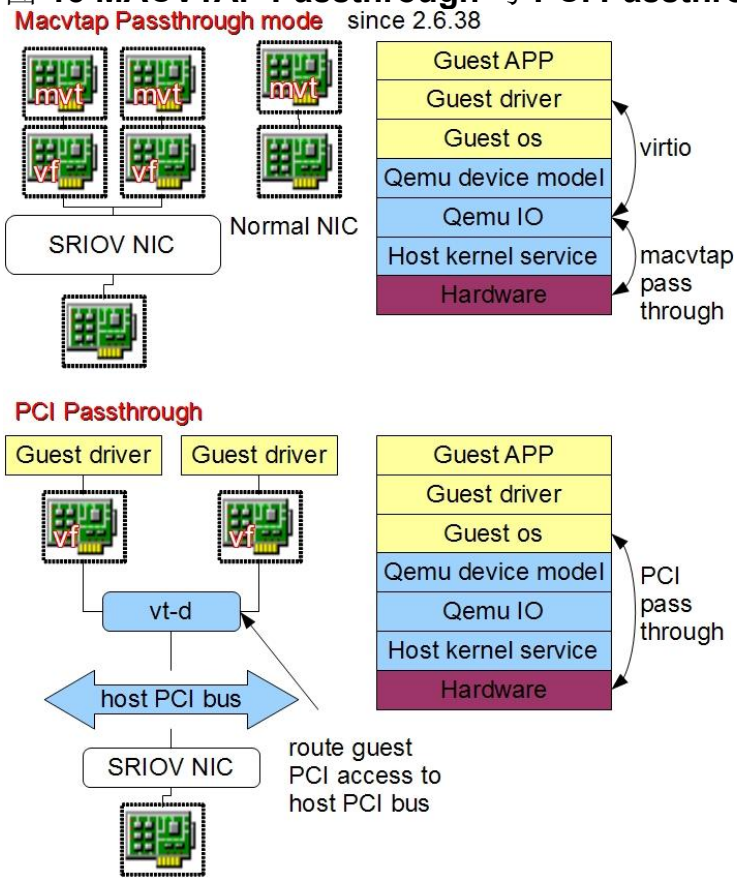


上图详细解释了 MACVTAP 设备原理。MACVTAP 设备和 VLAN 设备类似，是以一对多的母子关系出现的。在一个母设备上可以创建多个 MACVTAP 子设备，一个 MACVTAP 设备只有一个母设备，MACVTAP 子设备可以做为母设备，再一次嵌套的创建 MACVTAP 子设备。母子设备之间被隐含的桥接起来，母设备相当于现实世界中的交换机 TRUNK 口。实际上当 MACVTAP 设备被创建并且模式不为 Passthrough 时，内核隐含的创建了 MACVLAN 网络，完成转发功能。MACVTAP 设备有四种工作模式：Bridge、VEPA、Private，Passthrough。Bridge 模式下，它完成与 Bridge 设备类似功能，数据可以在属于同一个母设备的子设备间交换转发，虚拟机相当于简单接入了一个交换机。当前的 Linux 实现有一个缺陷，此模式下 MACVTAP 子设备无法和 Linux Host 通讯，即虚拟机无法和 Host 通讯，而使用传统的 Bridge 设备，通过给 Bridge 设置 IP 可以完成。VEPA 模式是对 802.1Qbg 标准中的 VEPA 机制的部分软件实现，工作在此模式下的 MACVTAP 设备简单的将数据转发到母设备中，完成

数据汇聚功能，通常需要外部交换机支持 Hairpin 模式才能正常工作。Private 模式和 VEPA 模式类似，区别是子 MACVTAP 之间相互隔离。Passthrough 模式下，内核的 MACVLAN 数据处理逻辑被跳过，硬件决定数据如何处理，从而释放了 Host CPU 资源。

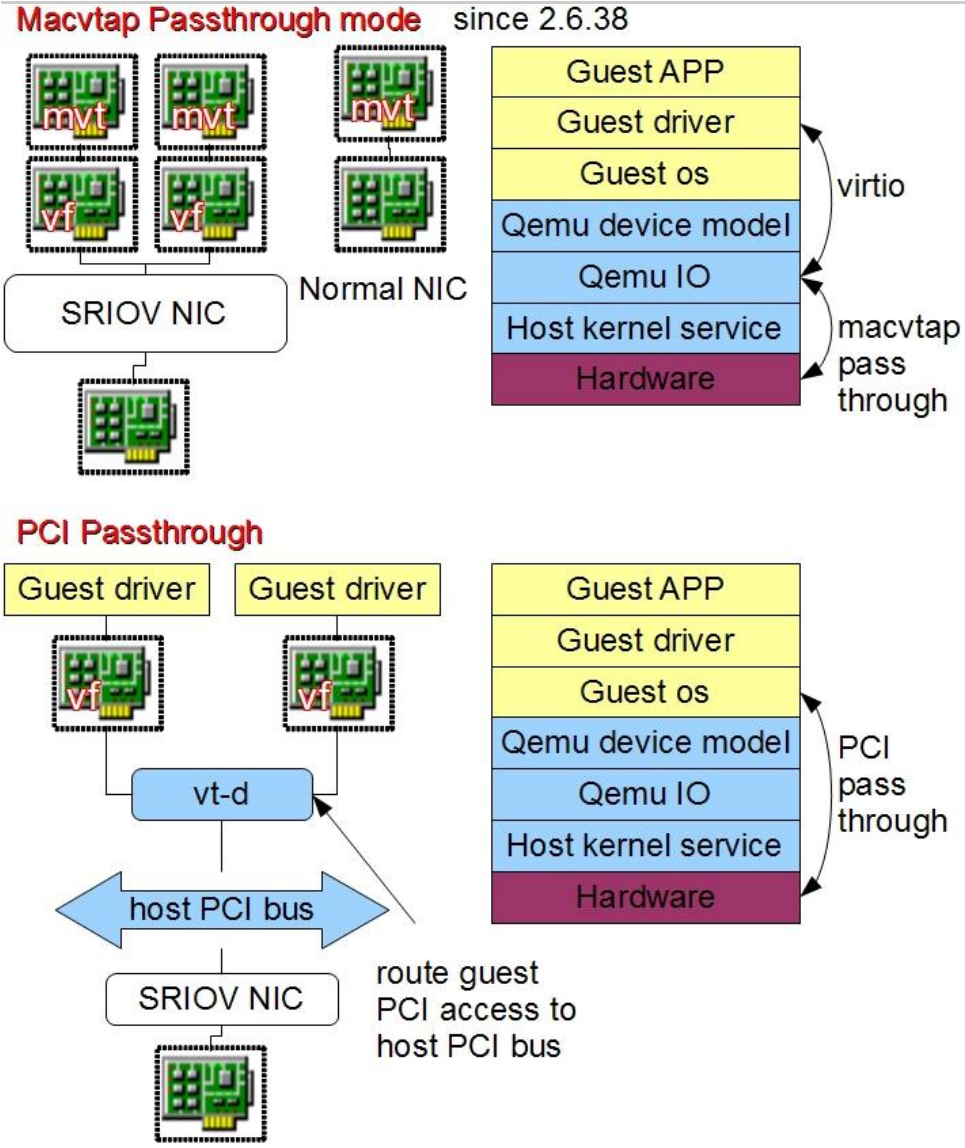
图中画出了 SR-IOV(Single Root I/O Virtualization)网络设备存在的情况下，通过 MACVTAP 设备使用 VF 的一种情况。其中 VF 设备是支持 SR-IOV 的物理网卡虚拟出来的虚拟网卡，每一个虚拟网卡都可以被当成一个真实的网卡使用，虚拟网卡之间相互隔离，从而分享了硬件资源。

图 **13 MACVTAP Passthrough 与 PCI Passthrough**



点击查看大图

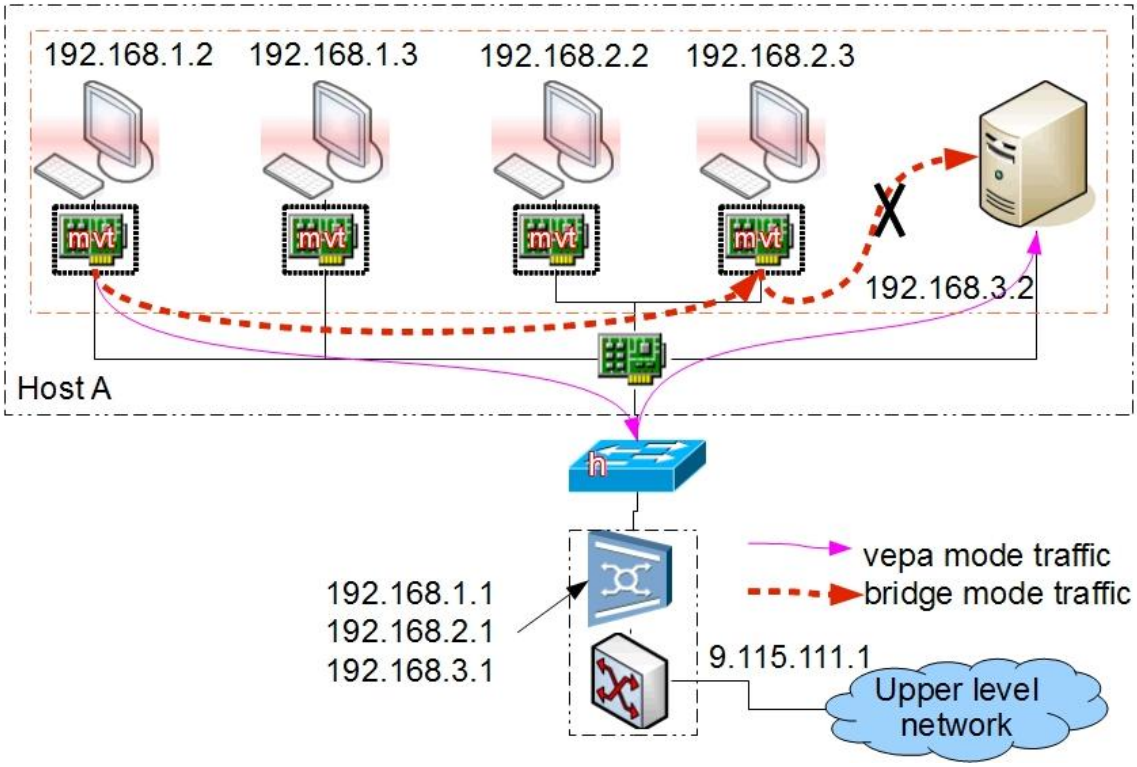
图 **13 MACVTAP Passthrough 与 PCI Passthrough**



MACVTAP Passthrough 概念与 PCI Passthrough 概念不同，上图详细解释了两种情况的区别。PCI Passthrough 针对的是任意 PCI 设备，不一定是网络设备，目的是让 Guest OS 直接使用 Host 上的 PCI 硬件以提高效率。以 X86 平台为例，数据将通过需要硬件支持的 VT-D 技术从 Guest OS 直接传递到 Host 硬件上。这样做固然效率很高，但因为模拟器失去了对虚拟硬件的控制，难以同步不同 Host 上的硬件状态，因此当前在使用 PCI Passthrough 的情况下无法做动态迁移。MACVTAP Passthrough 仅仅针对 MACVTAP 网络设备，目的是饶过内核里 MACVTAP 的部分软件处理过程，转而交给硬件处理。在虚拟化条件下，数据还是会先到达模拟器 I/O 层，再转发到硬件上。这样做效率有损失，但模拟器仍然控制虚拟硬件的状态及数据的走向，可以做动态迁移。综上所述，对于一个 SRIOV 网络设备，可以用两种模式使用它：MACVTAP Passthrough 与 PCI Passthrough，取决于用户对效率与功能的选择。

模拟传统以太网

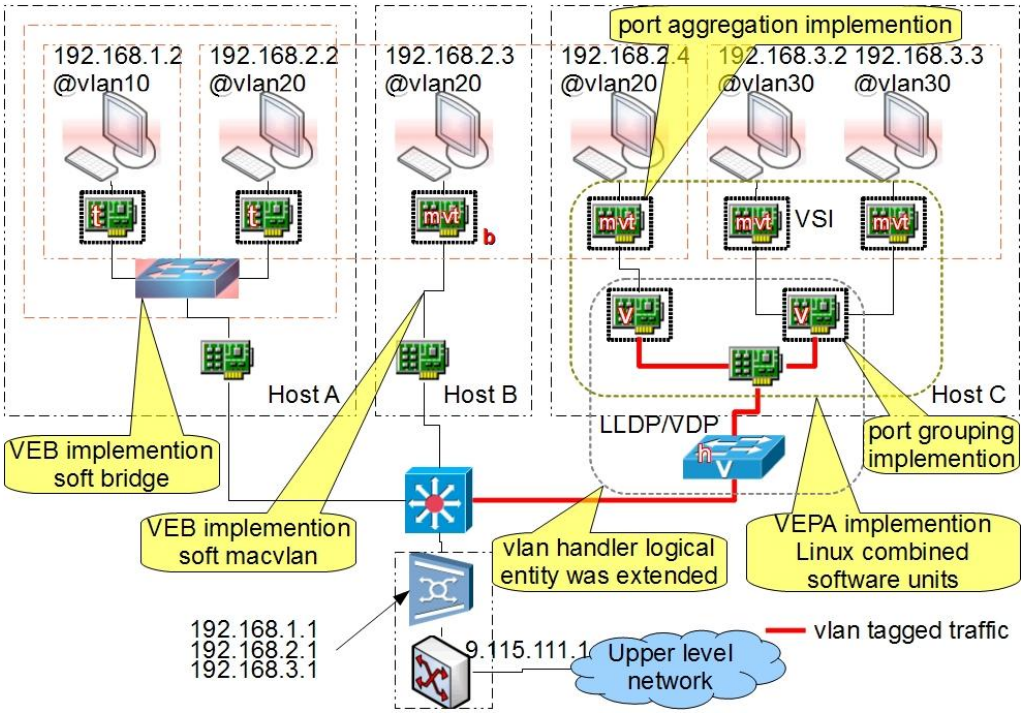
图 **14** .虚拟网络 **A_M0**



此图表示使用 MACVTAP 设备对现实网络 A 进行模拟的情况。为减少图例，图中的 MACVTAP 可以工作在 Bridge 或 VEPA 模式，曲线分别表示两种模式下数据的流向。工作在 Bridge 模式下时，数据无法从虚拟机流向寄主 Linux 系统用户程序。工作在 VEPA 模式下时没有此限制，但一级交换机必须工作在 Hairpin 模式。此虚拟网络类似地映射了网络 A，但仍然存在广播域混乱问题，原因是虚拟端口没有被分组。如前文所述，工作在 VEPA 模式的 Linux MACVTAP 设备只实现了数据汇聚功能。对比网络 A_V1，可以看到 MACVTAP 设备代替了 TAP 与 Bridge 设备组合。此网络没有使用寄主 Linux 系统的路由和 IP Tables，这些任务重新由外界物理网络设备承担，这也是 802.1Qbg 技术的目标之一，即让专业的网络设备承担网络数据处理任务，因此使用 MACVTAP 设备无法像网络 A_V1 那样使用寄主 Linux 系统的附属网络服务。

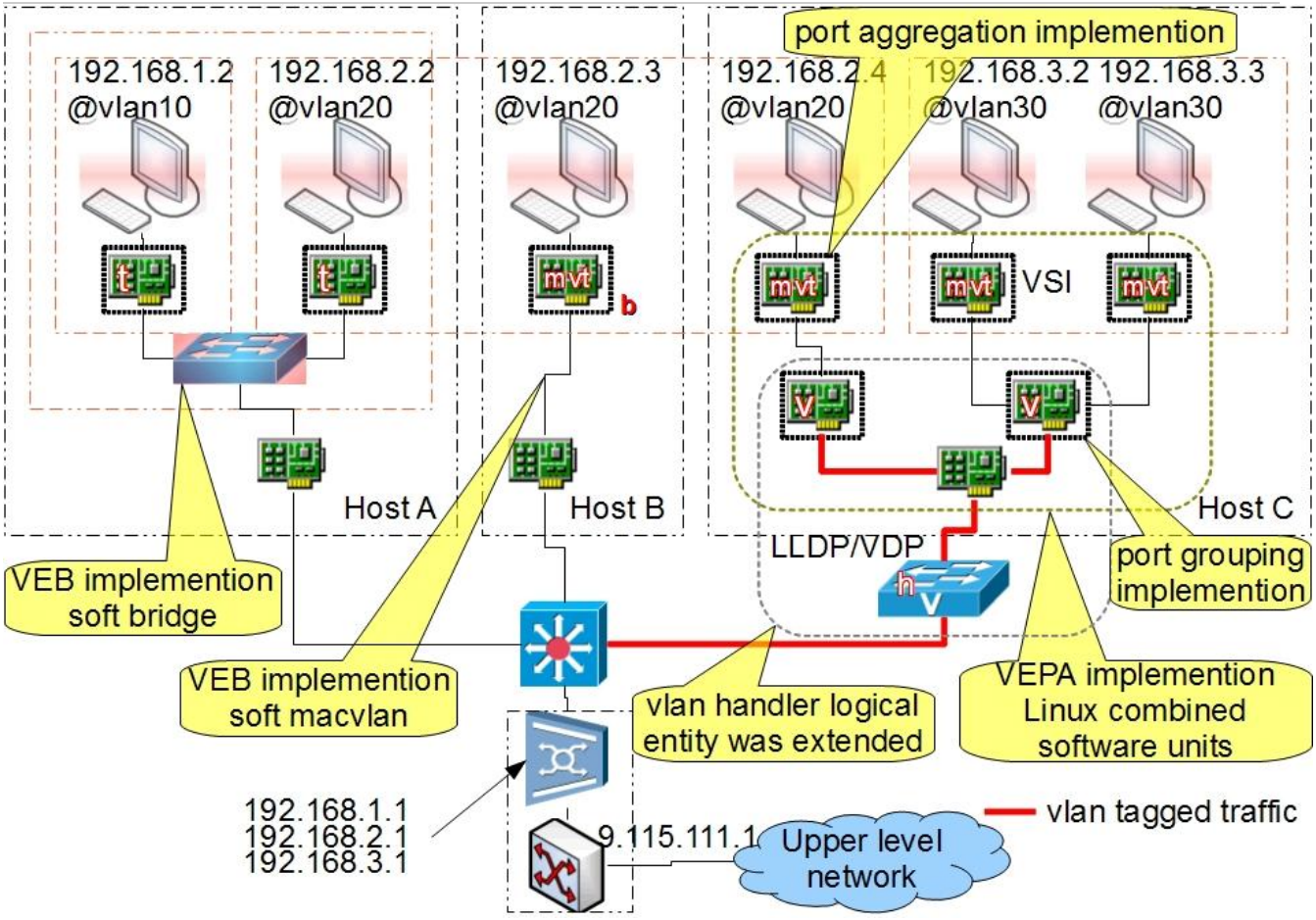
模拟 **802.1Q VLAN** 以太网

图 **15.**虚拟网络 **B_M0**



点击查看大图

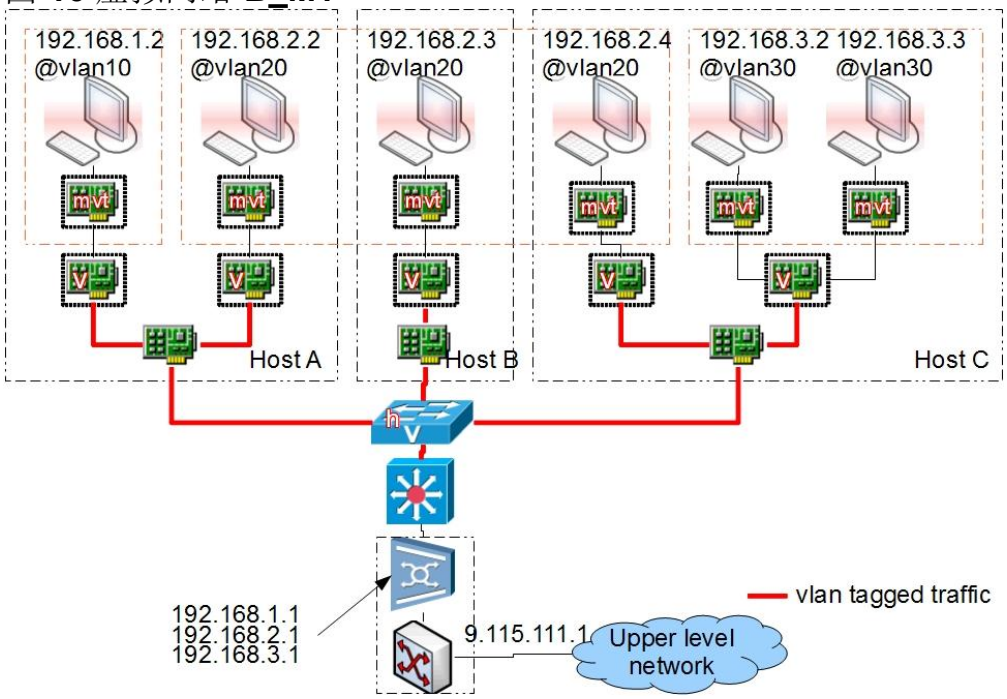
图 **15.**虚拟网络 **B_M0**



上图较为精确的用 MACVTAP 设备模拟了网络 B。Linux Bridge 设备与工作在 Bridge 模式下的 MACVTAP 设备都可以看成是对 802.1Qbg VEB 概念的软件实现。Host C 上加入了 Linux VLAN 设备，参照 VEPA 标准，VLAN Tag 可以用来对数据及虚拟端口进行分组。在 Host C 上，工作在 VEPA 模式的 MACVTAP 设备完成了汇聚功能，

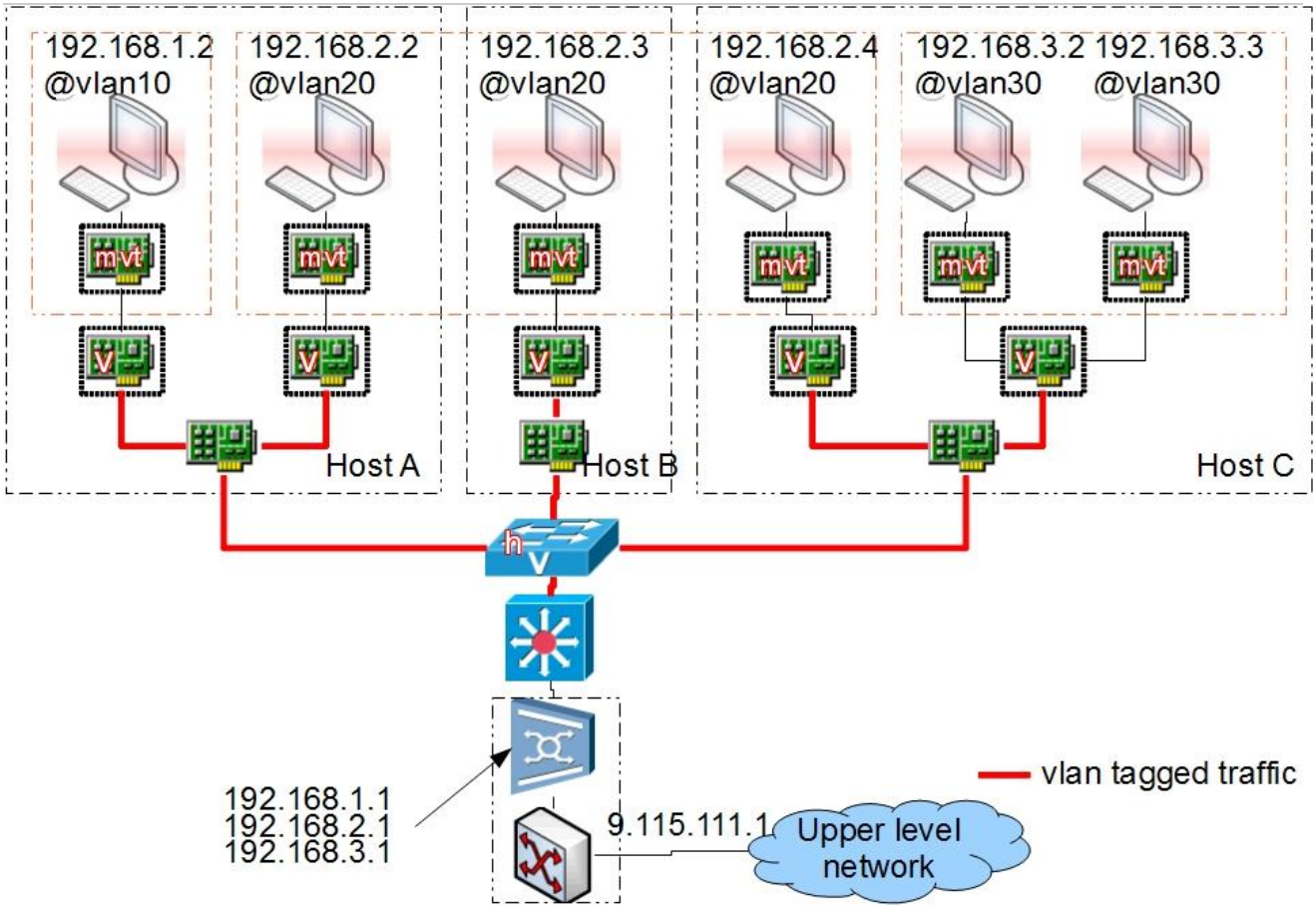
VLAN 设备完成了分组功能，两者组合形成了一个完整的对 VEPA 技术的软件实现，从而正确的隔离了 Host C 上的虚拟机所处的逻辑子网。Host C 与工作在 Hairpin 模式的一级交换机组合，能导出 Host C 上所有虚拟机的网络数据到网络设备侧进行管控。读者可以对比网络 B_V0 找出有哪些设备被 MACVTAP 设备代替。和网络 B_V0 一样，Host A 上由于没有引入 VLAN 设备，还存在广播域交叉问题。

图 **16** 虚拟网络 **B_M1**



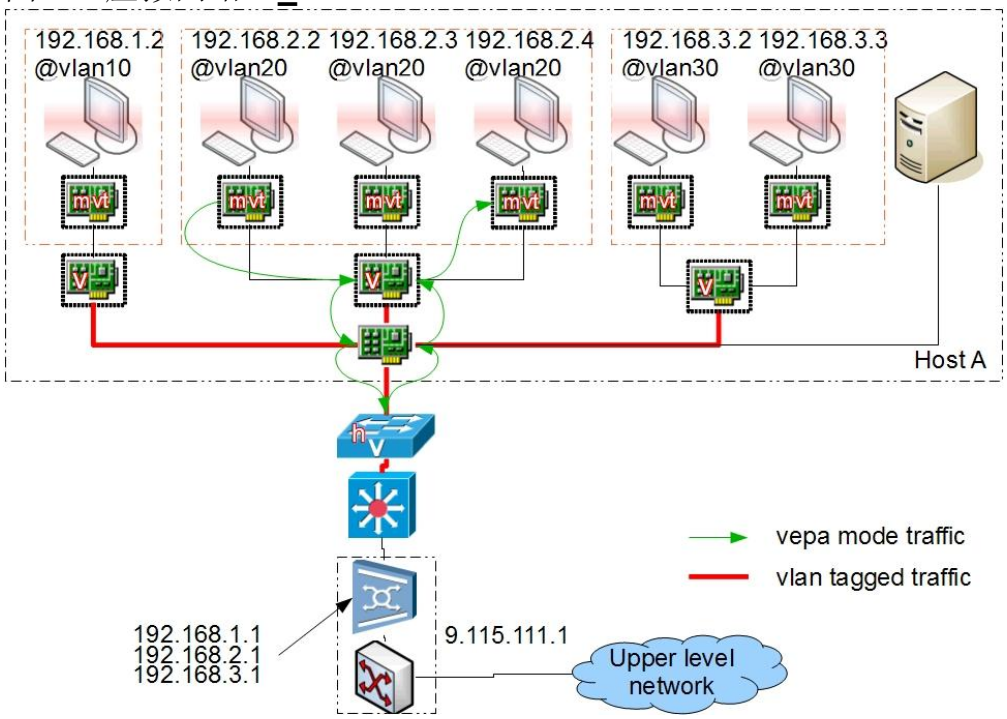
点击查看大图

图 **16** 虚拟网络 **B_M1**



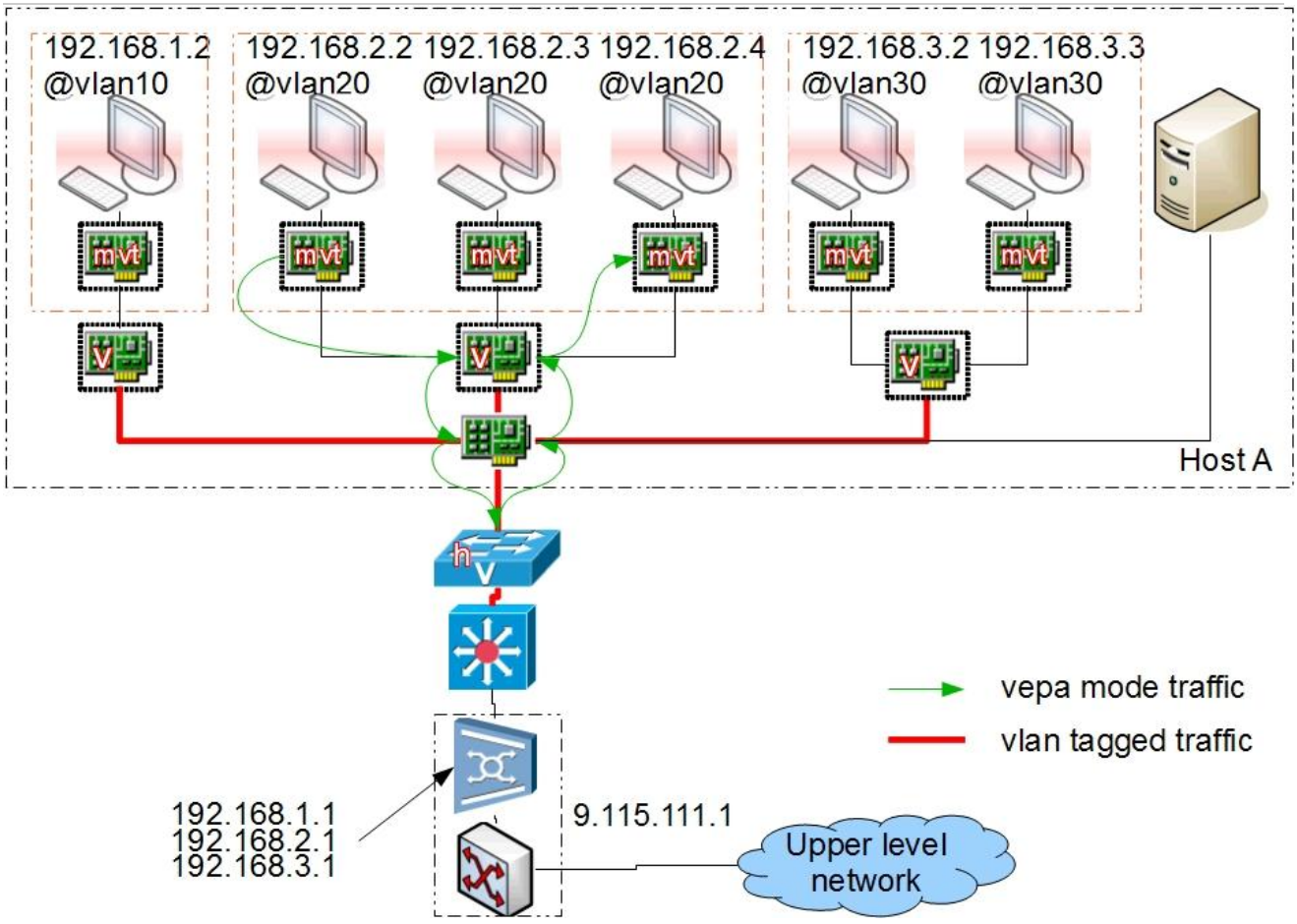
上图通过在 Host A 与 Host B 上引入 VLAN 与 MACVTAP 设备，解决了网络 B_M0 中的广播域问题，与网络 B_V1 类似。

图 **17** .虚拟网络 **B_M2**



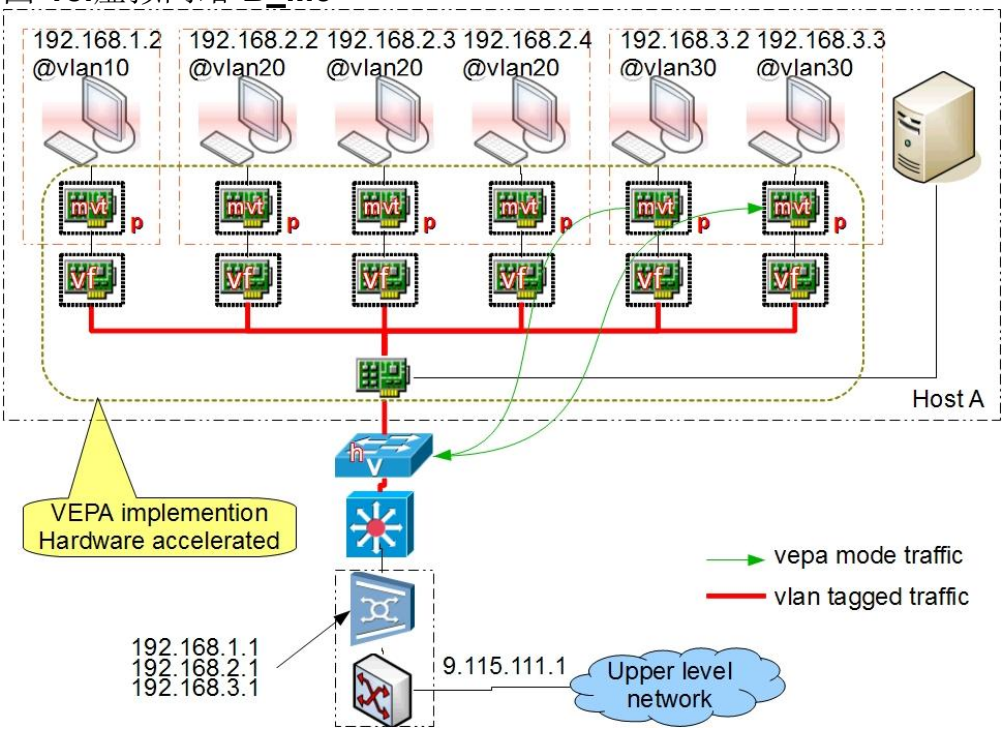
点击查看大图

图 **17** .虚拟网络 **B_M2**



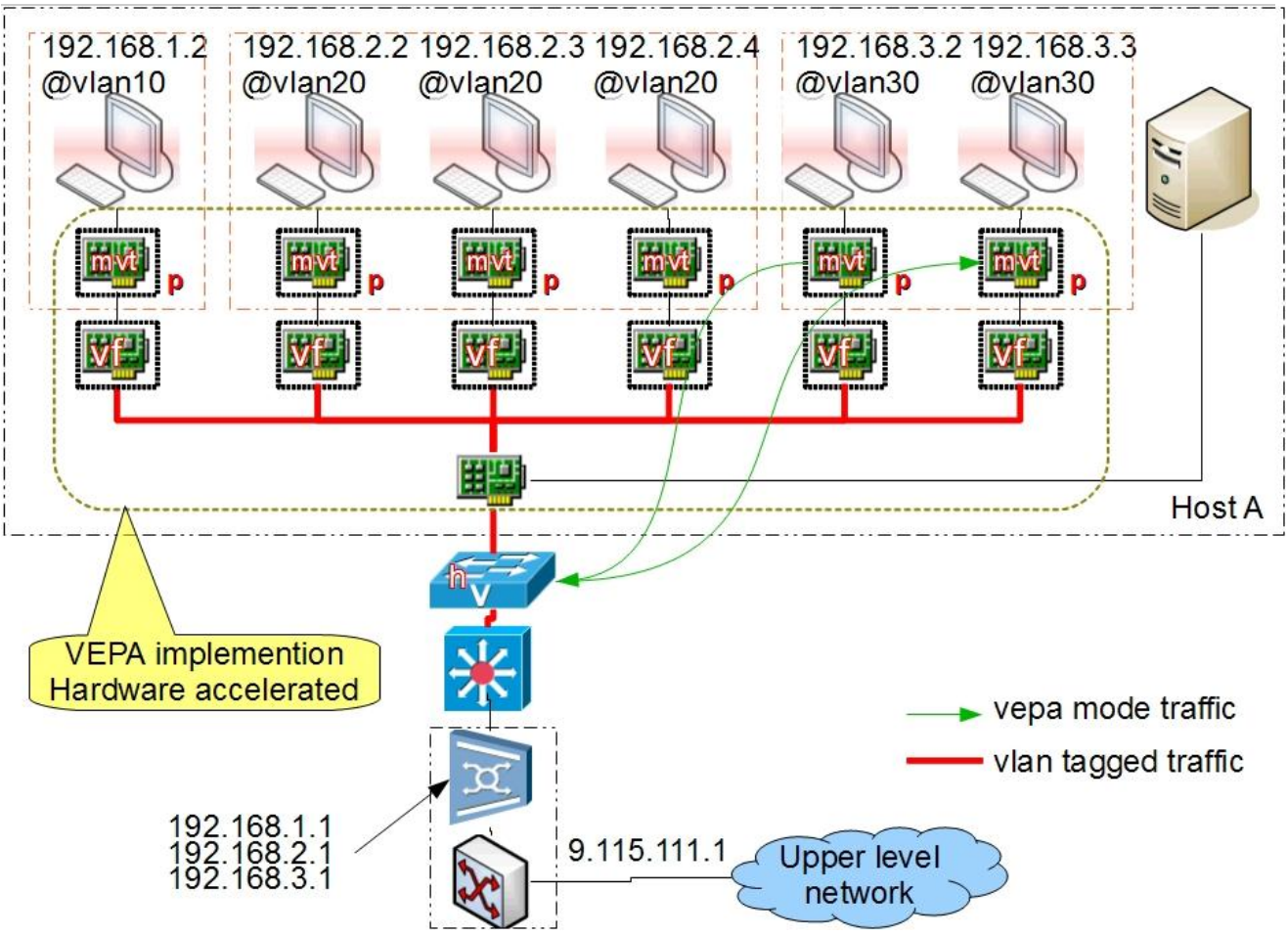
上图将所有虚拟机放置到同一个 Host 上，进一步节省了硬件资源，形成与网络 B_V2 类似的环境。可以看到，此环境下数据的走向稍有不同：虚拟机之间的通讯必须通过外部交换机，无法像 B_V2 那样内部完成。

图 **18.**虚拟网络 **B_M3**



点击查看大图

图 **18.**虚拟网络 **B_M3**



上图所示为使用 MACVTAP Passthrough 技术的一种网络配置，在不影响虚拟机动态迁移功能的前提下，进一步减少了寄主 Host CPU 负载，提高效率。

总结

虚拟化环境中的网络看似复杂，其实质是为虚拟客户机创建和现实世界中类似的网络结构。本文详细描述了 Linux 上虚拟网络的结构与意义，按照文中的原理，用户可以零成本地使用 Linux 软件实现的 Bridge、VLAN、MACVTAP 设备定制与现实世界类似的虚拟网络，也可以用非常低的成本按照 802.1Qbg 中的 VEPA 模型创建升级版的虚拟网络，引出虚拟机网络流量，减少 Host 服务器负载。当有支持 SRIOV 的网卡存在时，可以使用 Passthrough 技术近一步减少 Host 负载。

参考资料

学习

- Vconfig Man Page， vconfig 工具帮助文档。
- 802.1Q VLAN implementation for Linux，Linux 中 VLAN 模块如何实现的文档说明。
- IPROUTE2 Utility Suite Howto，Linux 里的 ip 工具使用说明。
- Virtual networking in Linux，以虚拟化应用为中心讲述主流的虚拟网络技术，主要以 openvswith 为例。
- Linux BRIDGE-STP-HOWTO，Linux 中的 bridge 设备使用说明。
- Linux Kernel Networking (Network Overview) by Rami Rosen，Linux 内核里的各种网络概念的含义，目的及用法简单介绍。
- 802.1Qbg - Edge Virtual Bridging，IEEE 802.1Qbg 标准。
- 在 developerWorks Linux 专区寻找为 Linux 开发人员（包括 Linux 新手入门）准备的更多参考资料。

讨论

- 加入 developerWorks 中文社区。查看开发人员推动的博客、论坛、组和维基，并与其他 developerWorks 用户交流。