(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104486192 B
(45)授权公告日 2019.02.01

(21)申请号 201410743517.X

(22)申请日 2014.12.05

(65)同一申请的已公布的文献号
申请公布号 CN 104486192 A

(43)申请公布日 2015.04.01

(73)专利权人 国云科技股份有限公司
地址 523808 广东省东莞市松山湖高新技术产业开发区科汇路1号中科院云计算中心19楼

(72)发明人 熊梦 杨松 莫展鹏 季统凯

(74)专利代理机构 广东莞信律师事务所 44332
代理人 余伦

(51)Int.Cl.
H04L 12/46(2006.01)
H04L 29/12(2006.01)

(56)对比文件
CN 101668022 A,2010.03.10,
CN 102255903 A,2011.11.23,
CN 103746997 A,2014.04.23,
CN 103001953 A,2013.03.27,
US 2014/0254603 A1,2014.09.11,
常立伟.Quantum中多租户隔离与网络服务扩展研究.《中国优秀硕士学位论文全文数据库》.2014,第1-94页.

审查员 穆剑

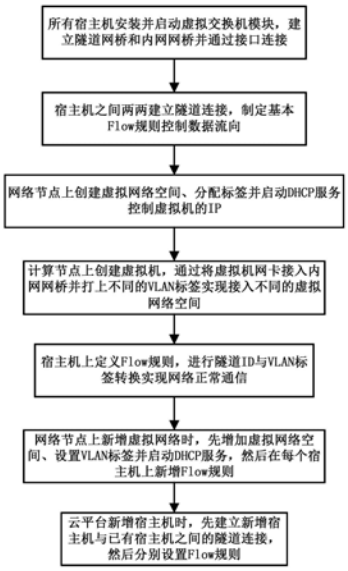
权利要求书1页 说明书5页 附图1页

(54)发明名称

一种虚拟网络隔离方法

(57)摘要

本发明涉及云计算技术领域,特别是一种虚拟网络隔离方法。本发明先在云平台的所有宿主机上安装和启动虚拟交换机模块,建立隧道网桥和内网网桥并通过接口连接;然后在宿主机之间两两建立隧道连接,指定基本Flow规则控制数据流向;接着选取网络节点创建虚拟网络空间并提供网络服务,选取计算节点创建虚拟机并接入各自的虚拟网络;进一步分别在网络节点和计算节点上面定义Flow规则,进行隧道ID与VLAN标签之间的转换、实现虚拟网络正常通信;最后根据需要可以灵活扩展网络节点上的虚拟网络和新增宿主机到云平台。本发明,解决了云平台中虚拟网络隔离所存在的通用性不强、网络连接不可控等问题;可以用于虚拟网络的隔离。



CN 104486192 B

1.一种虚拟网络隔离方法,其特征在于:所述的方法包括如下步骤:

步骤1:在云平台所有宿主机上安装并启动虚拟交换机模块,建立隧道网桥和内网网桥并通过接口连接;

步骤2:在宿主机之间两两建立隧道连接,制定基本Flow规则控制数据流向;

步骤3:选取一台宿主机作为网络节点,创建虚拟网络空间、分配VLAN标签并启动DHCP服务控制虚拟机的IP;

步骤4:计算节点上创建虚拟机,通过将虚拟机网卡接入内网网桥并打上不同的VLAN标签实现接入不同的虚拟网络空间;

步骤5:所有节点上定义Flow规则,进行隧道ID与VLAN标签转换实现网络正常通信;

步骤6:网络节点上新增虚拟网络时,先增加虚拟网络空间、设置VLAN标签并启动DHCP服务,然后在每个宿主机上新增步骤5所述Flow规则;

步骤7:云平台新增宿主机时,先建立新增宿主机与已有宿主机之间的隧道连接,然后分别设置步骤5所述Flow规则;

所述的隧道是一种通过使用互联网络的基础设施在网络之间传递数据的方式;使用隧道传递的数据或负载可以是不同协议的数据帧或数据包;隧道协议将其它协议的数据帧或数据包重新封装然后通过隧道发送,新的帧头提供路由信息,以便通过互联网传递被封装的负载数据;

所述的数据帧和数据包分别是指计算机二层网络和三层网络通信的基本数据单元;

所述的协议是指计算机网络通信上的一种规则约定;

隧道模式包括GRE隧道模式、VXLAN隧道模式和IPSec GRE隧道模式;

所述的虚拟交换机模块是指open vSwitch软件模块;

所述的网桥是指利用open vSwitch工具创建的虚拟网桥,其中内网网桥用以连接虚拟机,隧道网桥用以建立节点间的隧道连接,网桥间通过peer设备连接;

所述的peer设备是指具有两个端口的虚拟网络设备,包括:veth、patch。

2.根据权利要求1所述的虚拟网络隔离方法,其特征在于:所述的隧道连接是指设置一条本地节点IP到远程节点IP之间网络连接的规则;

所述的隧道ID是指一个整数,是隧道连接上的一个唯一标示;

所述的VLAN标签是指1到4096之间的整数,计算机网络中利用VLAN标签来实现数据链路层的网络隔离。

3.根据权利要求1所述的虚拟网络隔离方法,其特征在于:所述的Flow规则是指保存在Flow Table中的一些行为约定,通过它来定义如何处理接收到的数据帧。

4.根据权利要求2所述的虚拟网络隔离方法,其特征在于:所述的Flow规则是指保存在Flow Table中的一些行为约定,通过它来定义如何处理接收到的数据帧。

5.根据权利要求1至4任一项所述的虚拟网络隔离方法,其特征在于:所述的宿主机是指云平台中的物理服务器节点,包括计算节点和网络节点,其中计算节点是指可以在其上创建虚拟机的节点,网络节点则是指在其上启动DHCP服务、为虚拟机提供IP控制的节点;

所述的虚拟机IP控制是指通过虚拟机MAC与IP绑定来实现虚拟机只能使用被分配到的唯一IP进行网络通信。

一种虚拟网络隔离方法

技术领域

[0001] 本发明涉及云计算技术领域，特别是一种虚拟网络隔离方法。

背景技术

[0002] 在虚拟化平台中，尤其是在公有云平台中，考虑到安全问题以及用户数据的隐私问题，一般都需要对虚拟网络进行隔离，一般对虚拟网络的进行隔离都需要物理网络的支持，需要三层交换机提前划分VLAN，并添加相应的路由规则。实现方式如下：

[0003] 1、为宿主机的网卡配置trunk模式，并在宿主机上为每个VLAN创建一个网桥；

[0004] 2、创建虚拟机时，把虚拟机的网卡桥接到虚拟机所属VLAN对应的网桥上；

[0005] 3、通过VLAN之间的隔离可以实现虚拟网络之间的隔离，通过物理交换机配置相应的路由规则控制VLAN之间的互访规则。

[0006] 然而，上述方法存在以下的弊端：

[0007] 1、通用性不强，对于小型的私有网来说，一般没有配置三层交换机，在这种情况，上述方法就无法对虚拟网络进行划分，也无法对虚拟网络进行隔离。

[0008] 2、网络连接不可控，由于虚拟机的网络都是直接连通物理网络，因此对网络连接的控制要在外部通过交换机的配置实现，在内网出现攻击需要局部隔离时，只能通过粗粒度的VLAN进行阻断，不利于攻击源的排查。

发明内容

[0009] 本发明解决的技术问题在于提供一种虚拟网络隔离方法，解决了云平台中虚拟网络隔离所存在的通用性不强、网络连接不可控等问题。

[0010] 本发明解决上述技术问题的技术方案是：

[0011] 所述的方法包括如下步骤：

[0012] 步骤1：在云平台所有宿主机上安装并启动虚拟交换机模块，建立隧道网桥和内网网桥并通过接口连接；

[0013] 步骤2：在宿主机之间两两建立隧道连接，制定基本Flow规则控制数据流向；

[0014] 步骤3：选取一台宿主机作为网络节点，创建虚拟网络空间、分配VLAN标签并启动DHCP服务控制虚拟机的IP；

[0015] 步骤4：计算节点上创建虚拟机，通过将虚拟机网卡接入内网网桥并打上不同的VLAN标签实现接入不同的虚拟网络空间；

[0016] 步骤5：所有节点上定义Flow规则，进行隧道ID与VLAN标签转换实现网络正常通信；

[0017] 步骤6：网络节点上新增虚拟网络时，先增加虚拟网络空间、设置VLAN标签并启动DHCP服务，然后在每个宿主机上新增步骤5所述Flow规则；

[0018] 步骤7：云平台新增宿主机时，先建立新增宿主机与已有宿主机之间的隧道连接，然后分别设置步骤5所述Flow规则。

[0019] 所述的隧道是一种通过使用互联网络的基础设施在网络之间传递数据的方式;使用隧道传递的数据或负载可以是不同协议的数据帧或包;隧道协议将其它协议的数据帧或包重新封装然后通过隧道发送,新的帧头提供路由信息,以便通过互联网传递被封装的负载数据;

[0020] 所述的数据帧和数据包分别是指计算机二层网络和三层网络通信的基本数据单元;

[0021] 所述的协议是指计算机网络通信上的一种规则约定;

[0022] 所述的隧道模式包括GRE隧道模式、VXLAN隧道模式和IPSec GRE隧道模式。

[0023] 所述的虚拟交换机模块是指open vSwitch软件模块;

[0024] 所述的网桥是指利用open vSwitch工具创建的虚拟网桥,其中内网网桥用以连接虚拟机,隧道网桥用以建立节点间的隧道连接,网桥间通过peer设备连接;

[0025] 所述的peer设备是指具有两个端口的虚拟网络设备,可以是veth、patch等。

[0026] 所述的隧道连接是指设置一条本地节点IP到远程节点IP之间网络连接的规则;

[0027] 所述的隧道ID是指一个整数,是隧道连接上的一个唯一标示;

[0028] 所述的VLAN标签是指1到4096之间的整数,计算机网络中利用VLAN标签来实现数据链路层的网络隔离。

[0029] 所述的Flow规则是指保存在Flow Table中的一些行为约定,通过它来定义如何处理接收到的数据帧。

[0030] 所述的宿主机是指云平台中的物理服务器节点,包括了计算节点和网络节点等,其中计算节点是指可以在其上创建虚拟机的节点,网络节点则是指在其上启动DHCP服务、为虚拟机提供IP控制的节点;

[0031] 所述的虚拟机IP控制是指通过虚拟机MAC与IP绑定来实现虚拟机只能使用被分配到的唯一IP进行网络通信。

[0032] 本发明方案的有益效果如下:

[0033] 1、本发明的方法是一种通用的方法,不需要三层交换机的支持,既适用于小型的局域网,也适用于大型的局域网和广域网;

[0034] 2、本发明的方法的虚拟网络可控性较强,在虚拟机网络出现问题时,只需要断开网桥之间的连接即可,有利于网络故障的排查;

[0035] 3、本发明的方法的内网地址可重用,由于每个虚拟网络空间都采用单独的DHCP服务,因此内网IP地址可以重复,也可以使用回调的方式对内网IP地址与MAC地址进行绑定,对ARP攻击有很好的防护作用。

附图说明

[0036] 下面结合附图对本发明进一步说明:

[0037] 图1为本发明实现流程图。

具体实施方式

[0038] 如图所示,本发明在云平台所有宿主机上安装虚拟交换机模块,即open vSwitch软件,创建隧道网桥和内网网桥并通过接口连接,具体过程如下:

[0039] //安装open vSwitch

[0040] #rpm-ivh kmod-openvswitch-2.3.0-1.el6.x86_64.rpm

[0041] #rpm-ivh openvswitch-2.3.0-1.x86_64.rpm

[0042] //创建隧道网桥和内网网桥并启动

[0043] #ovs-vsctl add-br br-tun

[0044] #ovs-vsctl add-br br-int

[0045] #ifconfig br-tun up

[0046] #ifconfig br-int up

[0047] 添加peer设备连接隧道和内网网桥,这里的peer设备以patch设备为例:

[0048] #ovs-vsctl add-port br-int patch-tun

[0049] #ovs-vsctl set interface patch-tun type＝patch

[0050] #ovs-vsctl set interface patch-tun options:peer＝patch-int

[0051] #ovs-vsctl add-port br-tun patch-int

[0052] #ovs-vsctl set interface patch-int type＝patch

[0053] #ovs-vsctl set interface patch-int options:peer＝patch-tun

[0054] 在宿主机之间两两建立隧道连接,以节点A(30.30.1.2)与节点B(30.30.1.4)之间建立隧道连接为例:

[0055] 节点A上:

[0056] #ovs-vsctl add-port br-tun vxlan0--set interface vxlan0 type＝vxlan

[0057] options:df_default＝true options:in_key＝flow

[0058] options:local_ip＝30.30.1.2 options:out_key＝flow

[0059] options:remote_ip＝30.30.1.4

[0060] 节点B上:

[0061] #ovs-vsctl add-port br-tun vxlan0--set interface vxlan0 type＝vxlan

[0062] options:df_default＝true options:in_key＝flow

[0063] options:local_ip＝30.30.1.4 options:out_key＝flow

[0064] options:remote_ip＝30.30.1.2

[0065] 选取节点A作为网络节点,创建虚拟网络空间,分配VLAN标签5,启动IP网段为192.168.5.1/24的IHCP服务,具体操作如下:

[0066] //在虚拟交换机上添加本地vlan5端口

[0067] #ovs-vsctl add-port br-int tap5 tag＝5 -- set interface tap5 type＝internal

[0068] //创建并启动vlan5的虚拟网络服务:

[0069] #ip netns add dhcp-5

[0070] #ip link set tap5 netns dhcp-5

[0071] #ip netns exec dhcp-5 ip addr add 192.168.5.1/24 dev tap5

[0072] #ip netns exec dhcp-5 ifconfig tap5 promisc up

[0073] #ip netns exec dhcp-5 /usr/sbin/dnsmasq --strict-order

[0074] --bind-interfaces --conf-file＝ --domain＝local

[0075] --pid-file＝/opt/xm/test.pid --interface tap5 --except-interface＝lo

[0076] --dhcp-range＝192.168.5.1,static,120s --dhcp-option＝3,192.168.5.254

[0077] --dhcp-lease-max＝256 --dhcp-hostsfile＝/opt/xm/network.conf

[0078] --dhcp-script＝/opt/xm/update2db.py --leasefile-ro

[0079] 如上启动DHCP服务命令所示，network.conf配置文件格式如下，表示将192.168.5.2这个IP与MAC为d0:0d:11:22:33:44的虚拟机绑定，控制其IP：

[0080] d0:0d:11:22:33:44,vml,192.168.5.2

[0081] ……

[0082] 在宿主机上建立的每个隧道上设置flow规则，先为每个隧道设置一个ID，然后为从隧道进来的隧道ID为与本隧道对应的数据包打上虚拟网络空间对应的VLAN标签；为本地发送的具有虚拟机所在虚拟网络空间VLAN标签的数据包擦除标签，同时打上隧道ID，以vlan标签为5，tun_id为0x5为例，具体操作如下：

[0083] ////////table 0

[0084] #ovs-ofctl del-flows br-tun //删除已存在的所有flow规则

[0085] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝1 in_port＝1 actions＝resubmit(,2)" //从port1进来的，由table 2处理

[0086] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝1 in_port＝2 actions＝resubmit(,4)" //从port2进来的，由table 4处理

[0087] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝0 actions＝drop" //默认丢弃

[0088] /////////table 2

[0089] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝0 table＝2 dl_dst＝00:00:00:00:00:00/01:00:00:00:00:00 actions＝resubmit(,20)" //对于单播，由table 20处理

[0090] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝0 table＝2 dl_dst＝01:00:00:00:00:00/01:00:00:00:00:00 actions＝resubmit(,22)" //对于多播，由table 22处理

[0091] ////////table 3

[0092] //默认丢弃

[0093] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝0 table＝3 actions＝drop"

[0094] ///////table 4

[0095] //默认丢弃

[0096] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝0 table＝4 actions＝drop"

[0097] //将隧道id为0x5的数据为其添加vlan5并转发给table10处理

[0098] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝1 table＝4 tun_id＝0x5 actions＝mod_vlan_Vid:5,resubmit(,10)"

[0099] / /// //table 10

[0100] #ovs-ofctl add-flow br-tun "hard_timeout＝0 idle_timeout＝0 priority＝1 table＝10 actions＝learn(table＝20,priority＝1,hard_timeout＝300,NXM_OF_VLAN_TCI[0..11],NXM_OF_ETH_DST[]＝NXM_OF_ETH_SRC[],load:0->NXM_OF_VLAN_TCI[],lo ad:NXM_NX_TUN_ID[]->NXM_NX_TUN_ID[],output:NXM_OF_IN_PORT[]),output:1" //学习规则,将学习到的内容保存在table20中

[0101] / //// /table 20

[0102] //如果没有学习到什么,直接到22,如果原来学习到一些规则,则按照规则处理

[0103] #ovs-ofctl add-flow br-tun "hard timeout＝0 idle timeout＝0 priority＝0 table＝20 actions＝resubmit(,22)"

[0104] / ////table 22

[0105] //默认丢弃

[0106] #ovs-ofctl add-flow br-tun "hard timeout＝0 idle timeout＝0 priority＝0 table＝22 actions＝drop"

[0107] //将vlan为5的数据擦除vlan,并设置隧道id为0x5并从端口2出去

[0108] #ovs-ofctl add-flow br-tun"hard timeout＝0 idle timeout＝0 table＝22 dl_vlan＝5 actions＝strip_vlan,set_tunnel:0x5,output:2"

[0109] 以上步骤实现了基于隧道的虚拟网络通信,如果需要扩展虚拟网络和宿节点,只需分别执行上述对应的操作过程即可。

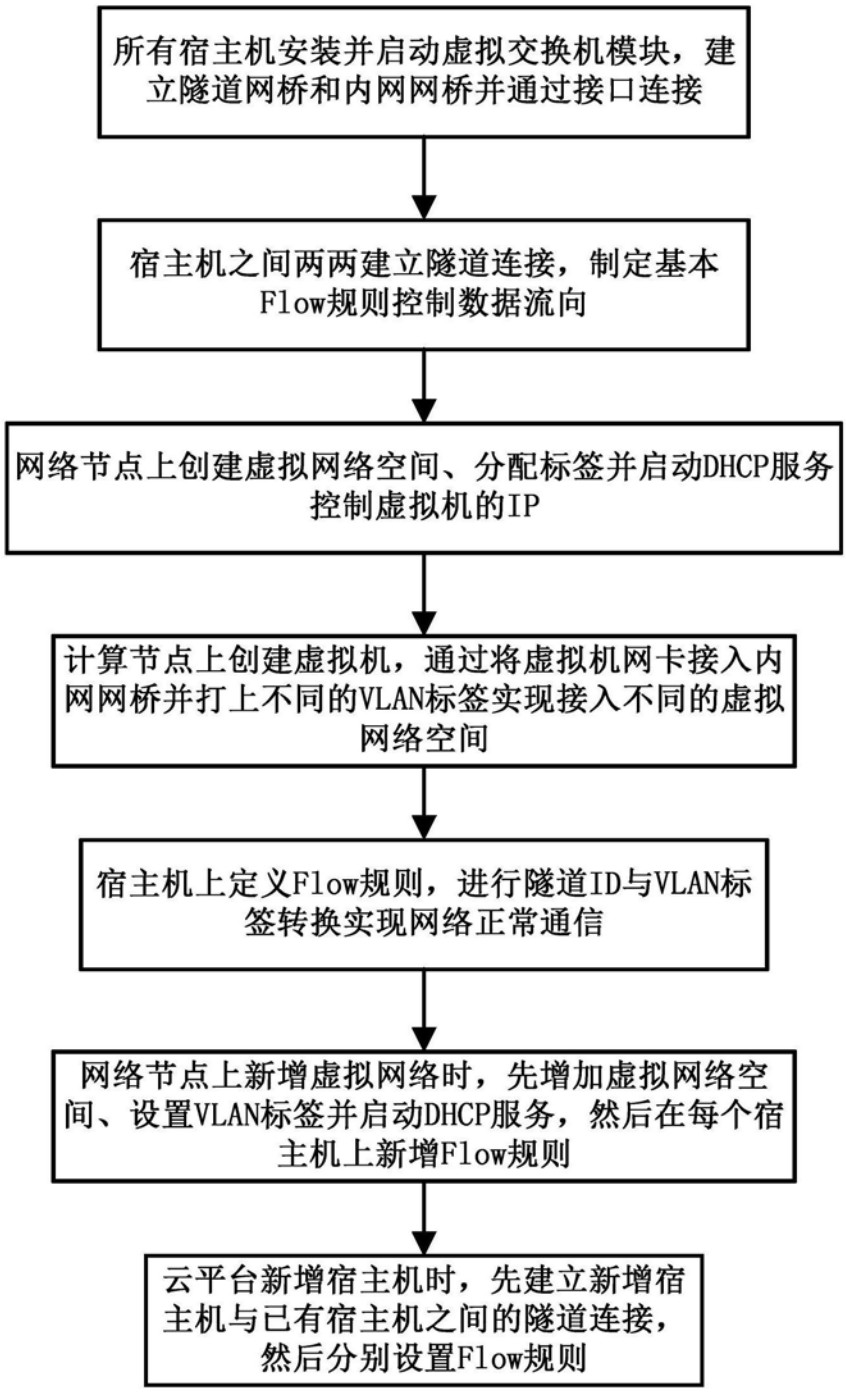

所有宿主机安装并启动虚拟交换机模块，建立隧道网桥和内网网桥并通过接口连接
宿主机之间两两建立隧道连接，制定基本Flow规则控制数据流向
网络节点上创建虚拟网络空间、分配标签并启动DHCP服务控制虚拟机的IP
计算节点上创建虚拟机，通过将虚拟机网卡接入内网网桥并打上不同的VLAN标签实现接入不同的虚拟网络空间
宿主机上定义Flow规则，进行隧道ID与VLAN标签转换实现网络正常通信
网络节点上新增虚拟网络时，先增加虚拟网络空间、设置VLAN标签并启动DHCP服务，然后在每个宿主机上新增Flow规则
云平台新增宿主机时，先建立新增宿主机与已有宿主机之间的隧道连接，然后分别设置Flow规则

图1