

基于 VRF 和 RT 实现 BGP/MPLS VPNs 中的 VPN 拓扑发现^{*}

梁海英^{1,2} 侯 锐² 兰丽辉² 高 远¹

(东北大学信息科学与工程学院 沈阳 110004)¹ (吉林师范大学计算机学院 四平 136000)²

摘 要 在 RFC 2547 中定义的 BGP/MPLS VPNs 允许服务提供者使用他们的 IP 骨干提供 VPN 服务,使用 BGP 对骨干网络的路由器分发 VPN 路由信息,使用 MPLS 转发 VPN 流量。BGP/MPLS VPNs 允许服务提供者在 VPN 内定义拥有任意数量结点的任意拓扑。服务提供者能建立使用相同核心网络的多个 VPN。目前大多数服务提供者手工地或通过使用配置的数据库实现 BGP/MPLS VPNs。本文描述的算法使 VPN 拓扑发现过程自动化。使用该算法,服务提供者能使用当前网络配置信息自动地发现 VPN 拓扑。

关键词 BGP, VPN 拓扑, RT, VRF, 发现

VRF/RT-based Discovery of VPN Topology in BGP/MPLS VPNs

LIANG Hai-Ying^{1,2} HOU Kun² LAN Li-Hui² GAO Yuan¹

(School of Information Science and Engineering, Northeastern University, Shenyang 110004)¹

(School of Computer, Jilin Normal University, Siping 136000)²

Abstract BGP/MPLS VPNs defined in RFC 2547 allow service providers to use their IP backbone to provide VPN services. BGP is used to distribute VPN routing information to the routers in the backbone network and MPLS is used to forward VPN traffic. BGP/MPLS VPNs allow service providers to define any arbitrary topology with any number of nodes in a VPN. The service provider can create multiple VPNs using the same core network. Currently most of the service providers track 2547 VPNs either manually or by using a provisioning database. Algorithms described in this paper aims at automating this VPN discovery procedure. Using our algorithms, service providers can automatically discover VPN topology that has already been configured using the current network configuration information.

Keywords BGP, VPN topology, RT, VRF, Discovery

1 引言

VPN(Virtual Private Network, 虚拟专用网)是指客户通过公共设施,使用和专用网络一样的安全策略,连接在一起的。RFC 2547^[1]定义了建立 VPN 的方法(即 BGP/MPLS VPNs)。在这个方法中,BGP(Border Gateway Protocol, 边界网关协议)用于发送 VPN 路由信息,MPLS(Multiprotocol Label Switching, 多协议标签交换)用于转发 VPN 流量。BGP/MPLS VPNs 允许服务提供者在 VPN 内定义拥有任意数量结点的任意拓扑。

在配置 VPN 之后,可以使用网络配置信息跟踪已有的 VPN 配置。虽然对提供配置 VPN 能力有可用的工具,但是为了定位可能在 VPN 里面发生的物理故障和测量 VPN 流量与服务质量^[2,3],还没有工具能考虑现有的 VRF 和 RT 关系,自动地建立组成 VPN 的不同组件。因此,拓扑发现工具提供必需的信息确保 VPN 按照规格配置,并根据组件拓扑提供可视化 VPN 信息。

本文提出了实现这个拓扑发现工具的算法。该算法不是针对个别厂商的,以便它能被任何厂商应用。IETF 草案^[4]介绍一个 PE 如何能使用协议发现 VPN 中的其他 PE。我们通过使用脱机算法发现所有配置的 VPN 的拓扑。

2 BGP/MPLS VPNs 简介^[1,5,6]

2.1 BGP/MPLS VPNs 的技术术语

在图 1 中,有两个 VPN(即 VPN A 和 VPN B)使用相同的核心网络。

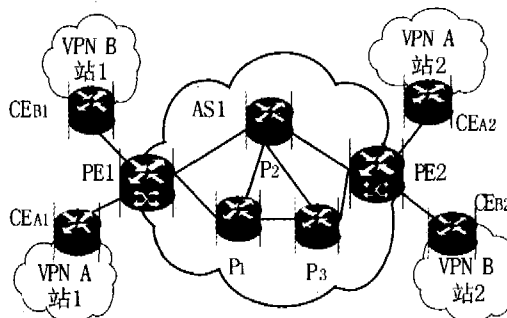


图 1 BGP/MPLS VPNs 的例子

2.1.1 CE(Customer Edge, 客户边界)路由器

CE 路由器是客户边界设备,它通过数据链路,如帧中继、ATM 或租用线路,将客户结点连接到或多个 PE 设备。CE 设备可以是主机或者是路由器,典型的是路由器,它和它直接连接的 PE 设备建立邻接。在此情况下,CE 路由器与 PE 设备之间通过静态路由、IGP(Interior Gateway Protocol, 内部网

^{*} 本文得到国家自然科学基金资助(①Internet 故障管理和路由稳定性研究,批准号:60073059;②Internet 域间路由稳定性和可管理性的研究,批准号:60273078)。梁海英 副教授,博士研究生;侯 锐 讲师,硕士;兰丽辉 讲师,硕士;高 远 教授,博士生导师。

关协议)、或EBGP(External BGP,外部边界网关协议),在AS(Autonomous System,自治系统)间交换客户VPN的网络可达性信息。

2.1.2 PE(Provider Edge,提供者边界)路由器

PE路由器是骨干网络的边缘设备,它直接连接到CE设备。PE路由器首先从CE设备获得当地VPN路由,然后与其它PE路由器用IBGP(Internal BGP,内部边界网关协议)在AS内交换VPN路由信息。

2.1.3 P(Provider,提供者)路由器

P路由器是服务提供者骨干网络的核心路由器,它不和CE设备相连。P路由器作为MPLS的过渡LSR(Label Switch Router,标签交换路由器),基于MPLS标签,在PE路由器之间交换VPN的数据流量。P路由器只需要维持到PE路由器的路由信息,而不需要维持到特定客户的VPN路由信息。

2.1.4 VRF(VPN Routing and Forwarding, VPN路由转发)表

VRF表定义连接到PE路由器中客户结点的成员关系。每个PE路由器为每个接口连接的客户结点维持VRF表,有几个直接相连的客户结点就有几个VRF表。只有属于同一VPN的各个客户结点,才能彼此交换信息。

2.1.5 RT(Route Target,路由目标)

RT是BGP的扩展团体属性^[7],将特定的RT和一条路由绑在一起,允许这条路由被放入从相应站点接收的路由流量的VRF表内。PE路由器使用这一属性输出当地路由到其他的VRF中,并且约束远程路由输入到它自己的VRF内。根据PE路由器上的VRF表考虑输出和输入路由。路由从VRF输出到BGP并且从BGP输入到VRF。输入路由目标将总是用于从当地BGP表采用的,并且被放在与具体的客户站点有关的VRF内的路由。输出路由目标将总是用于来自VRF的流量并且将它连同正确的路由目标和路由区别信息一起放入BGP表,以便,路由信息能被传输到连接到相同客户的其它的PE路由器。

2.2 通过BGP进行VPN路由分发

在BGP/MPLS VPNs中,分发VPN路由信息有5步^[4,5]。

第1步:从CE路由器和入口PE路由器分发。

第2步:在入口PE路由器输出到提供者的BGP。PE路由器为每个直接连接的VPN维持VRF表,该表定义连接到这个PE路由器上的VPN客户关系。入口PE路由器从它直接连接的CE路由器获得路由后,将这些路由加上特定的RT属性,安装到相应的VRF表中,然后将此路由转换成VPN-IPv4路由,输出到提供者的BGP。

第3步:穿过服务提供者骨干网络。入口PE路由器能与出口PE路由器通过BGP交换这些VPN-IPv4路由。

第4步:在出口PE路由器从提供者BGP输出。一旦出口PE路由器从其它PE路由器得到VPN-IPv4路由,它就为自己连接的VPN安装这些与RT属性相连的路由。

第5步:从出口PE路由器向CE路由器分发。这些被选中的VPN-IPv4路由被转换回IPv4路由,并输入到相应的VRF表中。安装到VRF表中的路由均可以被分配给相应的CE路由器。

在图1中,PE1有两个VRF表,VRF_{A1}包含与CE_{A1}交换的VPN路由和转发信息,而VRF_{B1}包含与CE_{B1}交换的VPN

路由和转发信息。CE_{A1}被连接到PE1的一个端口,而客户路由器CE_{B1}被连接到相同的提供者的边缘路由器PE1的另一个端口上,即两个CE被连接到同PE上。因为VPN A有两个站点,即站点1和站点2,则VRF_{A1}输出的RT即为VRF_{A2}输入的RT,这意味与VRF_{A2}相联系的CE_{A2}知道如何到达与VRF_{A1}相联系的CE_{A1}上的主机。同样地,对于CE_{A1}要到达CE_{A2}上的主机,VRF_{A2}输出的RT也应该是VRF_{A1}输入的RT。这样,在CE_{A1}上的主机与CE_{A2}上的主机之间的双向流量就可以流动了,这意味建立了VRF_{A2}和VRF_{A1}之间的双向VPN连接。因此,VRF和RT定义了VPN的拓扑。

3 VPN拓扑类型

如上所述,能由VRF使用RT和这些RT的输出及输入提供VPN拓扑。一些通常被提供的VPN拓扑包括下面三种类型^[8],如图2所示。

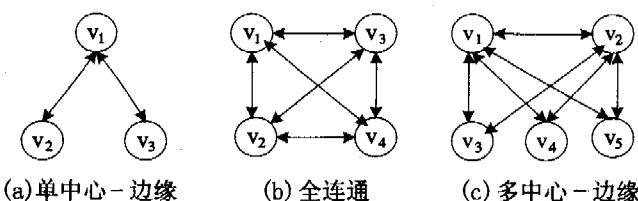


图2 VPN拓扑种类

(1)单中心-边缘:在这类拓扑中,单一中心VRF能对一组边缘VRF发送和接受VPN流量,但这组边缘VRF彼此不能交换VPN流量。图2(a)显示单中心-边缘拓扑,其中 v_1 是中心VRF, v_2 和 v_3 是边缘VRF。

(2)全连通:在这类拓扑中,一组VRF能彼此交换VPN流量。即,VRF被完全地连接。图2(b)显示全连通拓扑,其中所有VRF即 v_1 、 v_2 、 v_3 和 v_4 彼此之间都可以交换VPN流量。

(3)多中心-边缘(也叫做部分连通):在这类拓扑中,一组中心VRF既能通过全连通方式在彼此之间交换VPN流量,也能与一组边缘VRF以中心-边缘方式交换VPN流量。其中这组中心VRF之间彼此交换VPN流量,但那组边缘VRF之间彼此不交换VPN流量。图2(c)是多中心-边缘拓扑,其中 v_1 和 v_2 形成全连通的两个中心,而 v_3 、 v_4 和 v_5 是边缘VRF。

4 拓扑发现算法

4.1 几个定义

当配置VRF时,可能使用最少的RT。例如,为了配置全连通拓扑,只需要一个RT,只要在所有的VRF上定义一个RT,并且由所有的VRF输出和输入,那么在每对VRF之间就建立了VPN连接,因此导致全连通拓扑。同样地,为了配置单中心-边缘或多中心-边缘只需要2个RT。一个RT将会由所有边缘输入而由中心输出;另一个RT则由所有边缘输出而由中心输入。我们将通过使用最少数量的RT配置的组件叫做原子组件和分子组件。

定义1[原子组件] 拥有两个RT的单中心-边缘和拥有一个RT的全连通叫做原子组件。图2(a)和(b)是原子组件的例子。

定义2[分子组件] 对重叠的拥有原子组件的链路和结点没有任何限制的拥有两个RT的多中心-边缘叫做分子组件。图2(c)是分子组件的例子。注意,它由4个原子组件组

成:1个全连通原子组件和3个单中心-边缘原子组件。

要解决的问题是发现组成 VPN 的不同组件,而且这个问题是本文的重点。我们算法的目标是将 VPN 表示为原子组件和分子组件集。将全连通原子组件集表示为 F ,将单中心-边缘原子组件集表示为 HS ,将多中心-边缘分子组件集表示为 M 。将单向链路集表示为 U 。

4.2 拓扑发现算法

使用 RT 描述的 VPN,能被分解成几个不同的组件集。用 $(f_1, f_2, \dots, f_x)$ 表示使用结点 $f_i, i=1, \dots, x$ 建立的全连通。用 $(h \rightarrow s_1, s_2, \dots, s_x)$ 表示单中心-边缘,其中 h 作为中心, $s_i, i=1, \dots, x$ 作为边缘。同样地,用 $(h_1, h_2, \dots, h_y \rightarrow s_1, s_2, \dots, s_x)$ 表示多中心-边缘,其中 $h_i, i=1, \dots, y$ 作为中心,且 $(h_1, h_2, \dots, h_y)$ 是全连通的, $s_i, i=1, \dots, x$ 作为边缘。

对于给定的网络,设 VRF 数为 n , RT 数为 m 。将 VRF 依次编号为 $v_1, v_2, \dots, v_n$,将 RT 依次编号为 $r_1, r_2, \dots, r_m$ 。

我们算法的步骤如下。

第1步:准备 VRF-RT 表(VR)

根据要实现 VPN 拓扑发现的 BGP/MPLS VPNs 中的配置信息,准备一张叫作 VR 的 $m \times n$ 的表,其中,RT $r_k, 1 \leq k \leq m$ 形成表的第 k 行,VRF $v_i, 1 \leq i \leq n$ 形成表的第 i 列。依照指定的 RT 与 VRF 的输入输出关系,用 B, E 或 I 填入 VR 表的相应元素处。

其中, E 元素表示由行显示的 RT 被由列显示的 VRF 输出。同样地, I 元素表示由行显示的 RT 被由列显示的 VRF 输入。 B 元素表示由行显示的 RT 被由列显示的 VRF 输入且输出。

这一步的时间复杂度为 $O(nm)$ 。

注意:从 VR 表中删除只有元素且这个元素为 B 的行,也删除所有元素都为 E 或都为 I 的行。这将删除不能形成单向或双向链路的不能参与任何 VPN 的所有 RT。并把与之相对应的 RT 从 RT 集中删除。

第2步:构造矩阵(AM)

构造基于 VR 表的 $n \times n$ 的矩阵 AM。矩阵的行与列是由 VR 表中的 VRF 构成的。规则如下:

如果 $VR(r_k, v_i) = E \wedge VR(r_k, v_j) = I, i \neq j, i, j = 1, \dots, n$, 那么 $AM(v_i, v_j) = r_k$ 。

注意,将 B 既作为 E 也作为 I 看待。矩阵的对角线元素为空。

在最坏情况下,为了确定 $AM(v_i, v_j)$ 的值,需要检查 VR 表中与 VRF v_i 相对应列的所有元素,因此时间复杂度为 $O(n^2 m)$ 。

第3步:确定单向链路

如果一条链路连接的结点间没有进入相反方向的另外一条链路,那么这条链路为单向链路。从矩阵 AM 找出单向链路放入 U 中。因此,

$$U = \{(v_i, v_j) r_k | AM(v_i, v_j) = r_k \wedge AM(v_j, v_i) = \Phi, i \neq j, 1 \leq i, j \leq n, 1 \leq k \leq m\}.$$

同时,从矩阵 AM 中删除所有单向链路对应的元素,即删除满足单向链路条件的 $AM(v_i, v_j)$ 。

为了确定 U ,算法会需要对矩阵 AM 中的所有元素检查一次,因此时间复杂度为 $O(n^2)$ 。

第4步:确定全连通原子组件集

如果在 VR 表中的一个 RT 行有多于一个的 B ,那么这行中带 B 的 VRF 集作为全连通原子组件,放入 F 中。

定义 $b_k = \{v_i | VR(r_k, v_i) = B, 1 \leq i \leq n\}$,它是输入和输出 r_k 的所有 VRF 集。因此,

$F = \{b_k | |b_k| > 1, 1 \leq k \leq m\}$,其中 $|b_k|$ 表示集合 b_k 中的元素个数。

同时,在矩阵 AM 中删除满足如下条件的元素 $AM(v_i, v_j)$ 和 $AM(v_j, v_i)$,即令 $AM(v_i, v_j) = AM(v_j, v_i) = \Phi$ 。

条件: $v_i, v_j \in F \wedge VR(r_k, v_i) = B \wedge VR(r_k, v_j) = B, i \neq j, 1 \leq i, j \leq n, 1 \leq k \leq m$ 。

最坏的情形下,这一步需要检查一次 VR 表中所有元素,因此时间复杂度为 $O(nm)$ 。

第5步:建立候选中心集

为了找出所有单中心-边缘原子组件,我们先要选择中心。在矩阵 AM 中选择有一个或多个出度的 VRF 组成候选中心集,用 CH 表示。

$$CH = \{v_h | \exists i, k AM(v_h, v_i) = r_k, 1 \leq i \leq n, 1 \leq k \leq m\}.$$

在最坏的情形下,这一步的执行要访问矩阵 AM 的所有元素,时间复杂度为 $O(n^2)$ 。

第6步:建立优选中心集

只有当用来确定全连通原子组件的 RT 在 VR 表中的相应行中的某一元素中有 I 时,在全连通原子组件中的结点才能成为多中心-边缘分子组件中的中心。为了进一步确定分子组件,我们先准备优选中心集,用 PH 表示。因此

$$PH = \bigcap_{k=1}^m \{v_k | (v_k \in F) \wedge (\exists i, j VR(r_j, v_k) = B \wedge AM(v_k, v_i) = r_j \wedge VR(r_i, v_k) = I, 1 \leq i \leq n, 1 \leq j \leq m)\}$$

在最坏情形下, F 集可能有 n 个 VRF (即 VPN 由全连通原子组件组成),这意味可能检查 VR 表中的所有元素。时间复杂度为 $O(nm)$ 。

第7步:确定单中心-边缘原子组件集

我们确定 CH 中多少个中心是单中心-边缘原子组件的一部分。为此,一定有两个不同的 RT:一个 RT 作为 VRF 集到候选中心的输出,另一个 RT 作为从相同 VRF 集到候选中心的输入。

当 CH 中有元素时,重复做下列各子步骤,直到 CH 等于空集为止:

(1)对每个 VRF $v_h \in CH$,找出所有从 v_h 输出的 RT $r_k, 1 \leq k \leq m$ 。为每个 RT r_k 形成边缘集 $S(v_h, r_k)$ 。即 $S(v_h, r_k) = \{s | AM(v_h, s) = r_k\}$ 。

(2)对每个边缘集 $S(v_h, r_k), 1 \leq k \leq m$,找出使用相同 RT 输出到中心的最大 VRF 子集,并计算候选中心 v_h 的入度。

$$\text{入度}(v_h) = \max_{1 \leq k \leq m} |\{s | AM(s, v_h) = r_j, s \in S(v_h, r_k), r_j \neq r_k, 1 \leq j, k \leq m\}|.$$

(3)找出拥有最大入度的中心 $v_h \in CH$ 。如果有多个中心同时具有最大入度,那么选择中心 $v_h \in PH$ 。创造 $(v_h \rightarrow \{s_i | AM(s_i, v_h) = r_j, s_i \in S(v_h, r_k), r_j \neq r_k, 1 \leq j, k \leq m\})$ 。并将这个单中心-边缘, $(v_h \rightarrow s_1, \dots, s_x)$ 包含在 HS 集中。因此, $HS = HS \cup \{(v_h \rightarrow s_1, \dots, s_x)\}$ 。

(4)从矩阵 AM 中删除与这个单中心-边缘组件 $(v_h \rightarrow s_1, \dots, s_x)$ 有关的所有元素 $AM(v_h, s_i)$ 和 $AM(s_i, v_h)$ 。即令 $AM(v_h, s_i) = AM(s_i, v_h) = \Phi$ 。

(5)从 CH 中删除孤立 VRF (即,没有输入 RT 和输出 RT 的 VRF)。

对于每个 VRF,在(1)中花费 $O(m)$ 时间。在(2)中为了找出每个中心的入度,在最坏情况下,考虑为这个中心形成边

缘的所有 VRF, 将花费 $O(m)$ 时间。在(3)中为了找出拥有最大入度的中心, 在最坏的情形下, 需要 $O(n)$ 时间。(4)和(5)花费的时间是一个常量。因此, 对于 CH 中的每个元素, 在上述的子步骤中花费的时间是 $O(m+n)$ 。CH 集可能包含 n 个 VRF, 因此它会花费 $O(mn+n^2)$ 时间运行上述的子步骤。

第 8 步: 确定多中心-边缘分子组件集

下列的步骤准备 M 集。

(1) 从 F 集中取出元素是 PH 集成员的全连通组件, 形成新的全连通组件集 F。即, $F = \{b_k | b_k \in F \wedge b_k \notin PH, 1 \leq k \leq m\}$ 。如果全连通组件集 F 没有变化, 那么就停止。

(2) 检查全连通组件 b_k 的每个 VRF 是否是单中心-边缘集 S 的中心。

(3) 为每个中心 $\in b_k$ 的单中心-边缘原子组件检查, 由中心输出的 RT 是否相同, 而且被输出的 RT 是否与创造全连通 b_k 的 RT 相同。也检查被中心输入的 RT 是否相同。

(4) 如果检查通过, 那么将全连通和相应的单中心-边缘组件形成多中心-边缘集, 将其放入 M 中, 而且将其分别从 F 和 HS 中删除。返回(1)。

F 集中的每个结点可能不是全连通组件的一部份。在这种情况下, 检查每个 RT, 子步骤花费 $O(m)$ 时间。因此, 必需运行上述的子步骤的时间是 $O(nm)$ 。

第 7 步和第 8 步是算法核心, 在这两步发现 VPN 中的原子组件和分子组件集。这是算法的主要目标。

5 实验及分析

第 1 步: 根据 BGP/MPLS VPN 的配置信息得到它的 VR 表如表 1 所示, 其中 $n=9, m=6$ 。

表 1 VRF-RT 表(VR)

VRT RT	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
r_1	B	B	I	I					
r_2				B	B	B	B		
r_3	I	I	E	E					
r_4						E		I	I
r_5						I		E	E
r_6	E	B							

第 2 步: 建立对应矩阵 AM 如表 2 所示。

表 2 矩阵(AM)

VRT RT	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
V_1		r_1, r_6	r_1	r_1					
V_2	r_1		r_1	r_1					
V_3	r_3	r_3							
V_4	r_3	r_3			r_2	r_2	r_2		
V_5				r_2		r_2	r_2		
V_6				r_2	r_2		r_4	r_4	
V_7				r_2	r_2	r_2			
V_8						r_5			
V_9						r_5			

第 3 步: 没有单向链路。

第 4 步: 找出全连通原子组件集 $F = \{(v_1, v_2), (v_4, v_5, v_6, v_7)\}$, 结果矩阵如表 3 所示。

表 3 删除全连通原子组件后的矩阵(AM)

VRT VRF	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
V_1				r_1	r_1				
V_2				r_1	r_1				
V_3	r_3	r_3							
V_4	r_3	r_3							
V_5									
V_6								r_4	r_4
V_7									
V_8						r_5			
V_9						r_5			

第 5 步: 找出候选中心集。

$CH = \{v_1, v_2, v_3, v_4, v_6, v_8, v_9\}$ 。

第 6 步: 找出优选中心集 $PH = \{v_1, v_2\}$ 。

第 7 步: 确定单中心-边缘原子组件集。分析过程如表 4 所示。

表 4 确定单中心-边缘原子组件集的过程

边缘集	元素	输出到中心	入度
$S(v_1, r_1)$	$\{v_3, v_4\}$	$\{v_3, v_4\}$ 使用 r_3 到 v_1	$\text{Max} \{v_3, v_4\} =2$
$S(v_2, r_1)$	$\{v_3, v_4\}$	$\{v_3, v_4\}$ 使用 r_3 到 v_2	$\text{Max} \{v_3, v_4\} =2$
$S(v_3, r_3)$	$\{v_1, v_2\}$	$\{v_1, v_2\}$ 使用 r_1 到 v_3	$\text{Max} \{v_1, v_2\} =2$
$S(v_4, r_3)$	$\{v_1, v_2\}$	$\{v_1, v_2\}$ 使用 r_1 到 v_4	$\text{Max} \{v_1, v_2\} =2$
$S(v_6, r_4)$	$\{v_8, v_9\}$	$\{v_8, v_9\}$ 使用 r_5 到 v_6	$\text{Max} \{v_8, v_9\} =2$
$S(v_8, r_5)$	$\{v_6\}$	$\{v_6\}$ 使用 r_4 到 v_8	$\text{Max} \{v_6\} =1$
$S(v_9, r_5)$	$\{v_6\}$	$\{v_6\}$ 使用 r_4 到 v_9	$\text{Max} \{v_6\} =1$

从表 4 中可以看到, 最大入度是 2, 符合条件的有 v_1, v_2, v_6 。我们选择 $(v_1 \rightarrow v_3, v_4)$, 因为 $v_1 \in PH$ 。因此, $HS = \{(v_1 \rightarrow v_3, v_4)\}$, $CH = \{v_2, v_3, v_4, v_6, v_8, v_9\}$ 。删除这个原子组件后的矩阵如表 5 所示。

表 5 删除一个原子组件后的矩阵 AM

VRT VRF	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
V_1									
V_2				r_1	r_1				
V_3				r_1	r_1				
V_4									
V_5									
V_6								r_4	r_4
V_7									
V_8						r_5			
V_9						r_5			

依此类推, 我们找出原子组件 $(v_2 \rightarrow v_3, v_4)$ 和 $(v_6 \rightarrow v_8, v_9)$ 。

因此 $HS = \{(v_1 \rightarrow v_3, v_4), (v_2 \rightarrow v_3, v_4), (v_6 \rightarrow v_8, v_9)\}$, 此时 $CH = \{\}$ 。

第 8 步: 找出多中心-边缘分子组件集。我们发现 $(v_1, v_2) \in F$, 并且 v_1 和 v_2 都是 HS 中的中心, $(v_1 \rightarrow v_3, v_4)$ 和 $(v_2 \rightarrow v_3, v_4)$ 都在 HS 中。由 v_3 和 v_4 输出的 RT 为 r_3 , 并且由 v_1 和 v_2 输入的 RT 相同, 也为 r_3 。

因此, $F = \{v_4, v_5, v_6, v_7\}$, $HS = \{(v_6 \rightarrow v_8, v_9)\}$, $M = \{(v_1, v_2 \rightarrow v_3, v_4)\}$ 。

通过上述 8 步发现的 VPN 拓扑如图 3 所示。

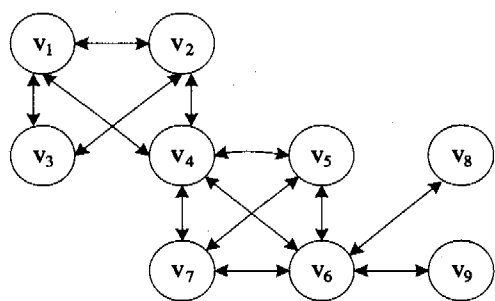


图3 VPN拓扑发现

结论 本文利用在 BGP/MPLS VPNs 中存储的配置信息 VRF 和 RT, 描述了使 VPN 拓扑发现过程自动化的算法。我们通过分析 VPN 结构, 并利用简单的原子组件和分子组件表示 VPN, 设计了 VPN 拓扑发现算法。我们的算法使用矩阵模型表现了 VPN 并识别简单的组件。使用该算法, 服务提供者能使用当前网络配置信息自动地发现 VPN 拓扑。

(上接第 21 页)

统对整个操作系统造成的性能影响主要体现在系统调用和文件系统访问操作上。原因是审计系统在系统调用处采集审计信息, 并将这些信息写入磁盘, 这是一种比较耗时的操作。提高审计系统性能的方法, 包括选择合适的审计点、减少审计记录的内容、使用审计记录缓冲区以较少写磁盘的次数、采取合理的审计配置、优化冗余代码等。开发审计系统时应采取以上各种措施, 以尽量减少审计系统所带来的系统负载。

结论 通用标准 CC 已成为了当今评估安全信息系统安全性的常用标准。本文以安全审计系统开发为例, 描述了如何使用各种保证措施, 以满足 CC 标准对安全保证的要求, 为今后开发高保证的安全信息系统, 尤其是安全操作系统, 提供了理论指导和实现思路。

参考文献

- 1 National Institute of Standards and Technology. Invitational Workshop on Information Technology (IT) Assurance and Trustworthiness. March 1995
- 2 Williams J R, Schaefer M, Landoll D J. Pretty Good Assurance. In: Proc. of the New Security Paradigms Workshop, 1995
- 3 蔡昱, 张玉清, 孙铁, 等. 安全评估标准综述. 计算机工程与应用, 2004, 40: 129~132
- 4 CSC-STD-001-83, Department of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria. DoD Computer Security Center, August 1983
- 5 DoD 5200. 28-STD, Department of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria. National Computer Security Center, Ft Meade, MD, USA, Dec. 1985
- 6 France, Germany, the Netherlands, the United Kingdom. Information Technology Security Evaluation Criteria, Version 1. 2.

参考文献

- 1 Rosen E, Rekhter Y. RFC2547, BGP/MPLS VPN. IETF, March 1999
- 2 HP Openview Network Services Management Solution for MPLS Networks. <http://www.hp.com>
- 3 Kim Youngtak, Choi Hyung-Woo, Kim Hyo-Sung. A QoS-guaranteed DiffServ-aware-MPLS VPN and its Network Management System. SNPD, 2003
- 4 Ould-Brahim H, Rosen E C, Rekhter Y. Using BGP as an Auto-Discovery Mechanism for Layer-3 and Layer-2 VPNs. IETF draft, June 2005
- 5 Tomsu P, Wieser G. MPLS-Based VPNs Designing Advanced Virtual Networks. Pearson Education, December 2001
- 6 Semeria C. RFC2547bis, BGP/MPLS VPN Fundamentals. Juniper Networks Inc White Paper, 2001
- 7 Sangli S R, Tappan D. BGP Extended Communities Attribute. Internet Draft, draft-ietf-idr-bgp-ext-communities-07. txt, March 2004
- 8 White R, McPherson D, Srihari S. Practical BGP. Addison Wesley Professional, July 06, 2004

Office for Official Publications of the European Communities, Jun. 1991

- 7 Canadian System Security Centre. The Canadian Trusted Computer Product Evaluation Criteria, Draft Version 3. 0e. Government of Canada, April 1992
- 8 Joint Technical Committee 1. Evaluation Criteria for IT Security? Part 1, 2, 3. ISO/IEC 15408-1, 1999(E), The International Organization for Standardization and the International Electrotechnical Commission, 1999
- 9 中国信息安全产品测评认证中心. GB/T 18336-2001、GB/T 18336-2001. 信息技术安全技术信息技术安全性评估准则
- 10 McLean J, Heitmeyer C. High Assurance Computer Systems: A Research Agenda. America in the Age of Information, National Science and Technology Council Committee on Information and Communications Forum, Bethesda, 1995
- 11 Ferraiolo K, Gallagher L, Thompson V. Building a Case for Assurance from Process. In: Proceeding of the 21 National Information Systems Security Conference, October 1999
- 12 LSPP. Labeled Security Protection Profile, Version 1. b. National Security Agency. October 1999
- 13 石文昌, 孙玉芳. 信息安全国际标准 CC 的结构模型分析. 计算机科学, 2001, 28(1): 8~11
- 14 石文昌, 孙玉芳. 通过 CC 标准的思想确定 RS-Linux 的安全可信度. 广西科学, 2001, 8(1): 15~18
- 15 BSM. SunSHIELD Basic Security Module Guide (Solaris 8). Sun Microsystems, Inc, February 2000
- 16 LMBench - Tools for Performance Analysis. <http://lmbench.sourceforge.net/>
- 17 McVoy L, Staelin C. lmbench: Portable tools for performance analysis. In: Proc. Winter 1996 USENIX, San Diego, CA, January 1996, 279~284