异构备份式的虚拟网映射方法研究

季新生 赵 硕* 艾健健 程国振 齐 超
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：在云计算和数据中心环境中，底层单个物理服务器的失效将对上层虚拟网络的服务性能造成很大的影响，现有利用冗余备份的方法能够在一定程度上降低底层物理设备失效带来的影响，但未考虑到物理服务器的同构性所带来的问题，为此，该文提出一种异构备份式的虚拟网映射方法。首先，只对关键的虚拟机进行冗余备份，降低备份资源的开销；然后，确保提供备份虚拟机的物理服务器与原物理服务器的系统类型的异构性，提高虚拟网的弹性能力；最后，以最小化链路资源开销作为虚拟网的映射目标，进一步降低备份资源的开销。实验表明，该方法在保证虚拟网络映射性能的前提下，能够大大提高虚拟网络的弹性能力。

关键词：虚拟网路；异构性；同构性；备份；关键虚拟机
中图分类号： TP302 **文献标识码**： A **文章编号**：1009-5896(2018)05-1087-07
DOI: 10.11999/JEIT170730

Research on Heterogeneous-backup Virtual Network Embedding

JI Xinsheng ZHAO Shuo AI Jianjian CHENG Guozhen QI Chao
(*National Digital Switching System Engineering & Technological Research Center*, *Zhengzhou* 450002, *China*)

Abstract: The failure of a single physical server will cause a bad impact on the performance of the virtual networks in the cloud computing and data center environments. The existing approaches that provision redundant and backup physical resources are able to reduce the impact due to failure of physical devices, however, they do not pay attention to homogeneous issues of the physical servers. Thus, a heterogeneous-backup virtual network embedding method is proposed. Firstly, the redundant and backup physical resources are only provided to critical virtual machines so that the method can save the overhead of backup resources. Then, in order to improve resilience of the virtual networks, substrate nodes corresponding to the primary and backup embedding of each virtual machine must be heterogeneous. Finally, the total cost of provisioning bandwidth on the substrate links for the total virtual network is minimized as the objective function to further save the overhead of backup resources. Simulation experiments demonstrate that the proposed approach is able to significantly improve resilience of the virtual networks on the premise of guaranteeing the performance of virtual networks embedding.

Key words: Virtual network; Heterogeneity; Homogeneity; Backup; Critical virtual machine

1 引言

网络虚拟化(Network Virtualization, NV)[1–4]虽然大大提高了数据中心网络的灵活性、应用的多样性、资源的利用率，能够最大程度地满足租户的需求，但多个虚拟网共享相同底层物理资源，也带来了新的挑战。例如，单个物理服务器失效或者被攻击将会使得其承载的所有虚拟机失效，进而影响虚拟机所在的虚拟网络，导致其无法为租户提供正常的网络服务，大大影响用户的使用体验[5,6]。

现有的研究方法主要通过可生存性的虚拟网映射(Survivable Virtual Network Embedding, SVNE)[7]来应对物理网络设备的失效，以此提高虚拟网络的弹性能力，增强虚拟网络提供服务的可靠性。目前 SVNE 方法主要分为两类：(1)重映射式[8–10]；(2)冗余备份式[11–15]。对于重映射式，当物理网络设备失效时，将其承载的虚拟机以及相关的虚拟链路进行重映射，从而保证虚拟网可以为租户

收稿日期：2017-07-19；改回日期：2018-01-15；网络出版：2018-02-05
*通信作者：赵硕 zshuo2016@126.com
基金项目：国家自然科学基金创新研究群体项目(61521003)，信息工程大学新兴方向培育基金(2016610708)，国家重点研发计划项目(2016YFB0800100, 2016YFB0800101)，河南省科技攻关计划项目(172102210615)，国家自然科学基金(61602509)
Foundation Items: The Foundation for Innovative Research Groups of the National Natural Science Foundation of China (61521003), The Emerging Direction Nurturing Foundation in Information Engineering University (2016610708), The National Key R&D Program of China (2016YFB0800100, 2016YFB0800101), The Science and Technology Research Project of Henan Province (172102210615), The National Natural Science Foundation of China (61602509)

继续提供正常的服务，但由于虚拟网服务提供商与租户签订的服务等级协议(Service-Level Agreements, SLAs)中往往对时延具有严格的限制，而对虚拟机和与之相关的虚拟链路进行重映射将会造成一定时间段服务的中断，无法很好地满足SLAs；对于冗余备份式，在虚拟网映射阶段，提前为该虚拟网提供备份的物理资源，若虚拟网所在的物理网络设备失效，则迅速切换到备份的物理资源上，从而保证虚拟网服务的延续性。无论是重映射式还是冗余备份式，都未考虑到物理服务器的同构性将对虚拟网络的弹性能力造成较大的影响。

为了便于描述，本文将虚拟机映射所在的物理服务器称为原物理服务器，备份虚拟机映射所在的物理服务器称为备份物理服务器，另外，若两个物理服务器的操作系统类型相同，则称两个物理服务器之间是同构的，否则，称为异构的。当备份物理服务器和原物理服务器之间是同构的，恶意攻击者通过探测原物理服务器的系统类型，不断挖掘系统漏洞，漏洞挖掘成功后则可以发动攻击使得原物理服务器失效，那么其上的虚拟机所需完成的任务可以切换到备份的物理服务器上，但攻击者对备份物理服务器进行攻击时，不需要继续进行漏洞挖掘，可以直接发动攻击使得备份物理服务器失效，从而导致虚拟网无法提供正常的服务，因此，物理服务器的同构性将会对虚拟网络的弹性能力造成较大的影响。

针对因物理服务器的同构性导致虚拟网络的弹性能力下降的问题，本文在现有 SVNE 方法的基础上，利用基于 Linux 内核操作系统(例如：Ubuntu, CentOS, Kubuntu, Debian, Fedora, Red Hat 等)种类的多样性，将物理服务器操作系统的多样性引入到 SVNE 方法中，提出一种异构备份式的虚拟网映射方法。本文的主要贡献如下：(1)只对关键虚拟机进行冗余备份，降低备份资源开销。(2)保证备份物理服务器与原物理服务器的系统类型是异构的，提高虚拟网络的弹性能力。(3)以最小化链路带宽资源开销作为虚拟网映射的目标函数，进一步降低备份资源开销。(4)进行了仿真实验，通过虚拟网络映射成功率、链路带宽总开销、虚拟网映射时间和虚拟网弹性能力 4 个方面对本文提出的方法与现有的 SVNE 方法进行了对比分析。

2 网络模型和问题描述

2.1 网络模型

数据中心网络： 数据中心网络可以表示为一个不带权重的无向图 $G^s=(N^s,L^s)$，其中，N^s,L^s 分别表示物理服务器和物理链路的集合。对于每一个物理服务器 $u\in N^s$，对应有一个可用的 CPU 处理能力 $C(u)$ 和操作系统类型 $\mathrm{Mark}(u)$。同样，对于每个物理链路 $(u,v)\in L^s$，对应有一个可用的带宽资源 $B(u,v)$。

虚拟网请求： 虚拟网络请求是不断到达的，虚拟网请求的集合模型表示为 $G^v=\left\{G_1^v,G_2^v,\cdots,G_k^v\right\}$，其中 $k=1,2,\cdots,n$。第 i 个虚拟网请求可以表示为 $G_i^v=(N_i^v,L_i^v,\mathrm{Ar}_i^v,\mathrm{Dur}_i^v)$。类似地，$N_i^v,L_i^v$ 分别表示虚拟网 G_i^v 中虚拟机和虚拟链路的集合，Ar_i^v 表示虚拟网 G_i^v 的到达时刻，Dur_i^v 表示虚拟网 G_i^v 的生存时间。对于每一个虚拟机 $\overline{u}_i\in N_i^v$，对应有一个虚拟机资源需求 $C(\overline{u}_i)$，表示该虚拟机所需的 CPU 资源。每条虚拟链路 $(\overline{u}_i,\overline{v}_i)\in L_i^v$ 对应一个链路资源需求 $B(\overline{u}_i,\overline{v}_i)$，表示该虚拟链路所需的带宽资源。

2.2 问题描述

(1)攻击过程描述： 根据文献[16]知，攻击者实施攻击的过程主要分为 3 个阶段：探测系统信息、挖掘系统漏洞、执行攻击，如图 1 所示，其中，w_1，w_2，w_3 分别表示 3 个阶段持续时间占攻击花费总时间的比例，且满足式(1)。

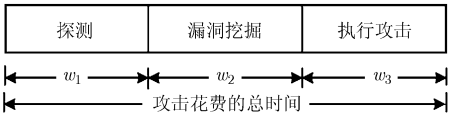

图 1 攻击过程

$$w_1+w_2+w_3=1,\quad w_1,w_2,w_3\in[0,1] \tag{1}$$

攻击者发动攻击的具体过程为：(1)探测系统信息阶段为攻击者根据网路信息确定攻击目标并收集目标系统的相关信息等；(2)挖掘系统漏洞阶段为攻击者从收集到的目标信息中提取可用的漏洞信息；(3)执行攻击阶段为利用系统漏洞获取系统的控制权，从而利用攻击工具发动攻击使得系统无法正常工作。

对于攻击者来说，挖掘系统漏洞阶段往往需要花费大量的时间，而且也可能无法成功地挖掘到系统的漏洞，但攻击者一旦成功的发现某种系统类型的漏洞，那么可以快速地对具有该系统类型的物理服务器实施攻击，使其不能正常工作。

(2)现有备份式的虚拟网映射存在的问题： 图 2 给出了备份式的虚拟网映射实例，其中颜色的类型代表该物理服务器的操作系统类型，例如，物理服务器 E 和 F 颜色相同，表示物理服务器 E 和 F

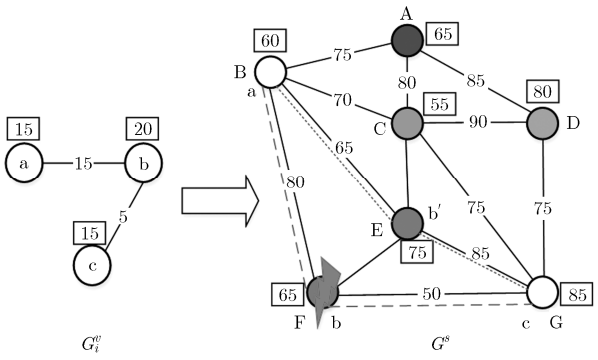

图 2 备份式的虚拟网映射实例

具有相同的系统类型，反之，若颜色不同，表示系统类型不同。在图 2 中，虚拟网 G_i^v 进行备份式的虚拟网映射时未考虑物理服务器系统的同构性，虚拟机 b 和备份虚拟机 b′ 所在的物理服务器具有相同的系统类型，若攻击者通过挖掘服务器 F 的系统漏洞，对服务器 F 成功进行了攻击使其失效，这时虚拟网提供商将虚拟机 b 所需完成的任务快速切换到虚拟机 b′ 上继续为租户提供服务，但由于服务器 E 和 F 具有相同的系统类型，具有相同的系统漏洞，因此攻击者不需要对服务器 E 进行系统漏洞的挖掘，可以快速地对服务器 E 发动攻击使其失效，那么虚拟网 G_i^v 将不能继续为租户提供正常的服务。

3　异构备份式的虚拟网映射方法

3.1 构建增强型虚拟网拓扑

本文以节点度的大小作为衡量关键虚拟机的指标，只对关键虚拟机和与之相关的虚拟链路进行冗余备份。如图 3 所示，根据节点的度的大小作为指标，得到虚拟机 b 为虚拟网 G_i^v 中的关键虚拟机，然后对虚拟机 b 和与之相关的虚拟链路进行冗余备份。

为了方便数学模型的建立以及虚拟网映射的描述，本文将冗余备份的虚拟成分和虚拟网构建成一个增强型虚拟网。$\overline{\mathrm{u}}_i'$ 表示虚拟机 $\overline{\mathrm{u}}_i$ 对应的备份虚拟机，$N_i^{v'}$ 表示虚拟网 G_i^v 中所有备份虚拟机的集合，$\left(\overline{u}_i', \overline{v}_i'\right)$ 表示备份虚拟链路，$L_i^{v'}$ 表示虚拟网 G_i^v 中所有备份虚拟链路的集合。因此，我们将增强型虚拟网请求的集合模型表示为 $G^{vv}=\left\{G_1^{vv}, G_2^{vv}, \cdots, G_k^{vv}\right\}$，第 i 个虚拟网请求可以表示为 $G_i^{vv}=\left(N_i^{vv}, L_i^{vv}, \mathrm{Ar}_i^v, \mathrm{Dur}_i^v\right)$，其中，虚拟机集合 $N_i^{vv}=N_i^v \cup N_i^{v'}$，虚拟链路集合 $L_i^{vv}=L_i^v \cup L_i^{v'}$。如图 3 所示，将虚拟网 G_i^v 和备份虚拟机 b′ 相关的虚拟成分构建成增强型虚拟网 G_i^{vv}。

3.2 异构备份式映射

对增强型虚拟网进行异构备份式映射，保证关键虚拟机 $\overline{u}_i$ 与备份虚拟机 $\overline{u}_i'$ 所在物理服务器的系统类型是异构的，同时保证备份虚拟机 $\overline{u}_i'$ 相关的备份虚拟链路不经过 $\overline{u}_i$ 所在的物理服务器，避免造成同时失效。如图 3 所示，在物理网络 G^s 中，备份虚拟机 b′ 可以选择服务器 A, C, D 进行映射，但不可以选择服务器 E。同时，为尽量降低冗余备份带来的资源开销，本文以最小化带宽资源开销作为映射的目标：在保证异构备份式映射的前提下，尽量缩短虚拟链路所映射的路径长度，降低链路带宽资源开销。例如，在图 3 中，服务器 A, C, D 都满足异构备份式映射的条件。但选择服务器 C 比 A 或者 D 所需的备份路径长度更短，花费的链路带宽资源更小，因此选择服务器 C 作为备份虚拟机 b′ 映射的物理服务器。

4　异构备份式的虚拟网映射模型和算法

4.1 异构备份式的虚拟网映射模型

(1)约束条件：

(a)虚拟机映射约束：

$$\sum_{u \in N^s} \alpha_u^{\overline{u}_i}=1,\ \forall \overline{u}_i \in N_i^{vv},\ \forall G_i^{vv} \in G^{vv} \tag{2}$$

$$\sum_{\overline{u}_i \in N_i^{vv}} \alpha_u^{\overline{u}_i} \le 1,\ \forall u \in N^s,\ \forall G_i^{vv} \in G^{vv} \tag{3}$$

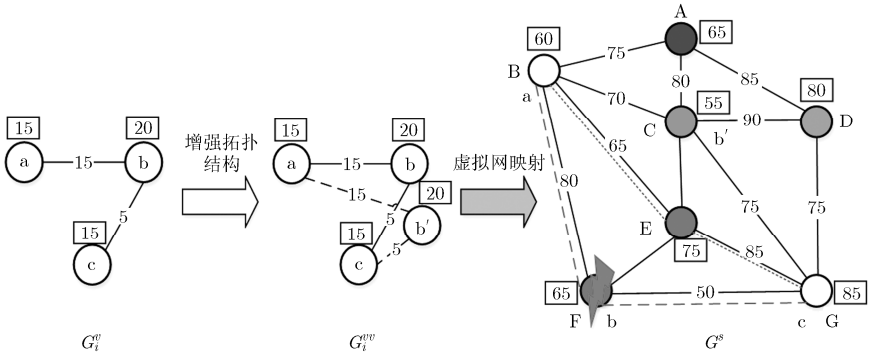

图 3 异构备份式的虚拟网映射实例

$$\sum_{G_i^{vv}\in G^{vv}}\sum_{\overline{u}_i\in N_i^{vv}} C(\overline{u}_i)\times\alpha_u^{\overline{u}_i}\le C(u),\ \forall u\in N^s \tag{4}$$

$$\sum_{\overline{u}_i\in N_i^{vv}}\alpha_u^{\overline{u}_i}=0,\ \forall u\in N^{s,f},\ \forall G_i^{vv}\in G^{vv} \tag{5}$$

式(2)表示一个虚拟机只能完整地映射到一个物理服务器上；式(3)表示同一虚拟网中的不同虚拟机不能映射到同一物理服务器上；式(4)表示一个物理服务器上承载的所有虚拟机的 CPU 资源需求之和不能超过该服务器的 CPU 资源；式(5)表示虚拟机不能映射到已经失效的物理服务器上。

(b)虚拟链路映射约束：

$$\sum_{G_i^{vv}\in G^{vv}}\sum_{(\overline{u}_i,\overline{v}_i)\in L_i^{vv}} B(\overline{u}_i,\overline{v}_i)\times\beta_{(u,v)}^{(\overline{u}_i,\overline{v}_i)}\le B(u,v),\ \forall(u,v)\in L^s \tag{6}$$

$$\sum_{v\in N(u)}\beta_{(u,v)}^{(\overline{u}_i,\overline{v}_i)}-\sum_{v\in N(u)}\beta_{(u,v)}^{(\overline{u}_i,\overline{v}_i)}=\alpha_u^{\overline{u}_i}-\alpha_u^{\overline{v}_i},\ \forall\overline{u}_i,\ \overline{v}_i\in N_i^{vv},\ \forall u\in N^s,\ \forall G_i^{vv}\in G^s \tag{7}$$

$$\sum_{(\overline{u}_i,\overline{v}_i)\in L_i^{vv}}\beta_{(u,v)}^{(\overline{u}_i,\overline{v}_i)}=0,\ \forall u,\ v\in N^{s,f},\ \forall G_i^{vv}\in G^v \tag{8}$$

式(6)表示一条物理链路上承载的所有虚拟链路的带宽资源需求之和不能超过该条物理链路的带宽资源；式(7)表示虚拟链路必须满足流约束条件；式(8)表示虚拟链路不能映射到已经失效的物理链路上。

(c)备份约束：

$$\sum_{(\overline{u}_i',\overline{v}_i')\in L_i^{vv'}}\sum_{v\in N(u)}\beta_{(u,v)}^{(\overline{u}_i',\overline{v}_i')}=0,\text{ if }\alpha_u^{\overline{u}_i}=1,\ \forall u\in N^s,\ \forall G_i^{vv}\in G^v \tag{9}$$

$$\mathrm{Mark}(u)\oplus\mathrm{Mark}(u')=1\text{ , if }\alpha_u^{\overline{u}_i}\times\alpha_{u'}^{\overline{u}_i'}=1,\ \forall u,\ u'\in N^s,\ \forall\overline{u}_i,\overline{u}_i'\in N_i^{vv},\ \forall G_i^{vv}\in G^v \tag{10}$$

式(9)表示与备份虚拟机 $\overline{u}_i'$ 相关的备份虚拟链路 $\left(\overline{u}_i',\overline{v}_i'\right)$ 不能经过虚拟机 $\overline{u}_i$ 所在的物理服务器；式(10)表示虚拟机 $\overline{u}_i$ 所在的物理服务器 u 的系统类型与备份虚拟机 $\overline{u}_i'$ 所在的物理服务器 u' 的系统类型不相同。

(2)目标函数：

$$\mathrm{Min}\sum_{(\overline{u}_i,\overline{v}_i)\in L_i^{vv}}\left[\partial\times B(\overline{u}_i,\overline{v}_i)\right],\ \forall G_i^{vv}\in G^{vv} \tag{11}$$

其中，∂ 表示单位链路带宽资源的开销大小。对于虚拟网映射来说，虚拟机映射到物理服务器上所需的 CPU 资源开销是固定的，但是虚拟链路的带宽资源开销是依据映射路径长度的大小而变化的，因此，为了尽量降低异构备份式的虚拟网映射的资源开销，我们将最小化虚拟链路的带宽资源开销作为虚拟网的映射目标函数。

4.2 算法设计

本文以式(2)~式(10)为约束条件，以最小化虚拟链路的带宽资源开销为映射目标，来设计异构备份式的虚拟网映射算法，算法流程如表 1 所示。

5 实验仿真

5.1 实验环境配置

本文实验采用 GT-ITM 工具生成虚拟网络及数据中心网络拓扑。数据中心网络设置为具有 100 个物理服务器，347 条物理链路组成，物理服务器的系统类型有 6 种，服从[0,5]的均匀分布。底层物理服务器 CPU 资源和链路的带宽资源符合[50, 100]的均匀分布。VN 中虚拟机的数目服从[6,15]的均匀分布，关键虚拟机的比例设置为 20%，虚拟机之间的连接概率为 0.5，虚拟机和虚拟链路所需资源分别服从[5,20]和[5,15]的均匀分布，每 100 个时间单元平均到达 4 个 VN 请求，VN 请求的生命周期服从参数为 1000 的指数分布。

本实验以文献[12]、文献[15]和本文提出的方法进行实验对比。文献[12]的方法是对所有的虚拟机进行同构备份式的虚拟网映射，文献[15]的方法是对关键虚拟机进行同构备份式的虚拟网映射。为了便于描述，将文献[12]、文献[15]以及本文提出的方法分别用 SVNE (Survivable Virtual Network Embedding), Hom-SVNE (Homogeneous-SVNE), Het-SVNE(Heterogeneous-SVNE)来表示。

5.2 性能分析

本节实验通过虚拟网络映射成功率和虚拟网络的弹性能力两个方面，对 SVNE, Hom-SVNE, He-SVNE 3 种方法的性能进行对比分析。

(1)虚拟网络映射成功率： 图 4 表示 SVNE, Hom-SVNE, Het-SVNE 3 种方法在虚拟网络映射成功率方面的比较。从图中可以看出，3 种方法的虚拟网络映射成功率的高低顺序为：Hom-SVNE> Het-SVNE>SVNE。由于 SVNE 方法对虚拟网中的所有虚拟机进行冗余备份，需要花费更多的物理网络资源，因此导致虚拟网络映射成功率下降明显；造成 Het-SVNE 方法的虚拟网络映射成功率略低于 Hom-SVNE 的原因主要有两个：(1)Het-SVNE 方法在进行备份物理服务器的选取时，需要保证原物理

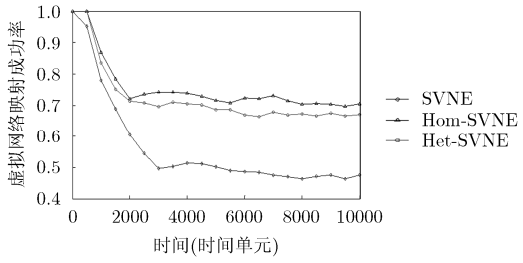

图 4 虚拟网络映射成功率

表 1 异构备份式的虚拟网映射算法

算法 1　异构备份式的虚拟网映射算法

输入：虚拟网请求拓扑 $G_i^v=\left(N_i^v,L_i^v\right)$，物理网络拓扑 $G^s=\left(N^s,L^s\right)$，关键虚拟机所占的比例 $R\in[0,1]$

输出：虚拟网映射方案 $M_i:G_i^v\to G_i^s\left(N_i^s,L_i^s\right)$

(1) $Q\leftarrow$ 虚拟机集合 N_i^v 按照节点度的大小进行降序排列；

(2) $N_i^{v-cr}=Q[1:Q.\text{le}\times R]$；$N_i^{v'}=N_i^{v-\text{cr}}$；$N_i^{vv}=N_i^v\cup N_i^{v'}$　//关键虚拟机的集合；备份关键虚拟机的集合；增强型虚拟网 G_i^{vv} 中虚拟机的集合

(3)**for each**　$\overline{u}_i\in N_i^{vv}$　**do**

(4)　$\overline{u}_i$. mapping = false

(5)　**for each**　$u\in N^s$　**do**

(6)　　**if**　u.ability $\geq\overline{u}_i$.requirement　**then**　//物理服务器是否能够满足虚拟机映射约束条件

(7)　　**if**　$\overline{u}_i\in N_i^{v'}$ &&　$\alpha_u^{\overline{u}_i}\times\alpha_{u'}^{\overline{u}_i'}=1$　**then**　//判断是否是备份虚拟机

(8)　　**if**　$\text{Mark}(u)\oplus\text{Mark}(u')\ !=1$　**then**　//备份虚拟机和虚拟机所在的物理服务器的系统类型是否是异构的

(9)　　**continue**　//若不是异构的，则查看一下物理服务器

(10)　　**end if**

(11)　　**end if**

(12)　　$\overline{u}_i$.mapping = true and **break**

(13)　　**end if**

(14)　**end for**

(15)　**if**　$\overline{u}_i$.mapping = false **then** reject　G_i^v and **return**

(16)　**end if**

(17) **end for**

(18)　$L_i^{v'}=L_i^v\left(N_i^{v'}\right)$；$L_i^{vv}=L_i^v\cup L_i^{v'}$　//备份虚拟机相关的虚拟链路的集合；增强型虚拟网 G_i^{vv} 中虚拟链路的集合

(19)　**for each**　$(\overline{u}_i,\overline{v}_i)\in L_i^{vv}$　**do**

(20)　　$(\overline{u}_i,\overline{v}_i)$.mapping = false

(21)　　**for each**　$(u,v)\in L^s$　**do**

(22)　　**if**　(u,v).ability $\geq(\overline{u}_i,\overline{v}_i)$.requirement,$(u,v)$　**then**

(23)　　$P\leftarrow(u,v)$　//满足虚拟链路 $(\overline{u}_i,\overline{v}_i)$ 的所有物理路径的集合

(24)　　**end if**

(25)　　**end for**

(26)　　将集合 P 中元素按照跳数的大小进行升序排列；

(27)　　**for** i=1 to P.length **do**

(28)　　**if**　$(\overline{u}_i,\overline{v}_i)\in L_i^{v'}$ && $\alpha_u^{\overline{u}_i}=1$　**then**　// 判断是否是备份虚拟链路

(29)　　**if**　$\sum_{(\overline{u}_i',\overline{v}_i')\in L_i^{v'}}\sum_{v\in N(u)}\beta_{(u,v)}^{(\overline{u}_i',\overline{v}_i')}\ !=0$ **then**　//备份虚拟链路 $\left(\overline{u}_i',\overline{v}_i'\right)$ 是否经过虚拟机 $\overline{u}_i$ 所在的物理服务器

(30)　　**continue**　//若经过，则不选择该条路径，查看下一条物理路径

(31)　　**end if**

(32)　　**end if**

(33)　　$\beta_{P[i]}^{(\overline{u}_i,\overline{v}_i)}=1$　//将虚拟链路 $(\overline{u}_i,\overline{v}_i)$ 映射到底层物理路径 $P[i]$ 上

(34)　　$(\overline{u}_i,\overline{v}_i)$.mapping = true and **break**

(35)　　**end for**

(36)　　**if**　$(\overline{u}_i,\overline{v}_i)$.mapping = false **then** reject　G_i^v and **return**

(37)　　**end if**

(38) **end for**

服务器和备份物理服务器系统类型的异构性，导致虚拟机可以选择映射的物理服务器的数目减少；(2)保证备份物理服务器与原物理服务器系统类型的异构性，可能伴随着备份虚拟链路路径长度的增加，需要消耗更多的链路带宽资源。根据实验数据计算得到，Het-SVNE 方法的虚拟网络映射成功率相较于 Hom-SVNE 平均下降 3.51%。

(2)虚拟网络的弹性能力：　为了更加直观地比

较 SVNE, Hom-SVNE, Het-SVNE 3 种方法在面对攻击者攻击时虚拟网络弹性能力的大小，本文通过定义虚拟网的瘫痪系数 ξ_i，将其作为虚拟网 G_i^{vv} 弹性能力大小的评价指标。

定义 1(虚拟网的瘫痪系数) 虚拟网中关键虚拟机失效的数量占总关键虚拟机数量的比例

$$\xi_i = \frac{\left|N_{i,\text{cri \&\& fail}}^{vv}\right|}{\left|N_{i,\text{cri}}^{vv}\right|}, \quad \left|N_{\text{cri}}^{vv}\right| \neq 0 \tag{12}$$

其中，$\xi_i \in [0,1]$，ξ_i 表示虚拟网 G_i^{vv} 的瘫痪系数，$N_{i,\text{cri}}^{vv}$ 为虚拟网 G_i^{vv} 中关键虚拟机的集合，$N_{i,\text{cri \&\& fail}}^{vv}$ 为虚拟网 G_i^{vv} 中失效的关键虚拟机的集合。在式(12)中，对于同一个虚拟网，其含有关键虚拟机的数量是固定的，即 $\left|N_{i,\text{cri}}^{vv}\right|$ 为定值。因此，当面对相同攻击时，若 ξ_i 的值越大，表示虚拟网 G_i^{vv} 中失效的关键虚拟机数量越多，那么虚拟网提供的服务受影响的程度越大，该虚拟网的弹性能力越小，反之，若 ξ_i 的值越小，表示该虚拟网的弹性能力越大。

本实验假设：(1)攻击者具有锁定特定攻击目标的能力；(2)攻击者攻击成功一个未知系统漏洞的物理服务器所需花费的时间统一设置为 T；(3)攻击者可以承受的忍耐时间用 $T_{\max}$ 表示，其中 $T_{\max} \geq T$。因此，在攻击者攻击花费时间为 $T_{\max}$ 时，同构系统类型的物理服务器和异构系统类型的物理服务器被攻击的失效个数可以分别表示为 $1+(T_{\max}-T)/((1-w_2)T)$ 和 $T_{\max}/T$。下面对虚拟网采用 SVNE, Hom-SVNE, Het-SVNE 3 种方法，在面对同一攻击者攻击时虚拟网络弹性能力的大小进行比较。

图 5 表示当攻击者的忍耐时间 $T_{\max}=2T$，挖掘系统漏洞所占的时间比例 w_2 为 1/2 时，采用 3 种方法的虚拟网络弹性能力的比较。从图中可以直观的看出，对于同一虚拟网，采取 SVNE 和 Hom-SVNE 方法被攻击失效的物理服务器数量相同，虚拟网瘫痪系数 ξ 也将相同，ξ 为 50%或者 75%。而采取 Het-SVNE 方法的原物理服务器和备份物理服务器的系统是异构的，采取 Het-SVNE 方法被攻击失效的物理服务器数量小于其他两种方法，虚拟网瘫痪系数 ξ 为 33%或者 50%。通过实验数据计算得到，相较于 SVNE 和 Hom-SVNE 方法，采用 Het-SVNE 方法进行虚拟网映射，虚拟网络的弹性能力要平均增大 19.8%。

图 6 展示了当攻击者的忍耐时间 $T_{\max}=2T$，挖掘系统漏洞所占的时间比例 w_2 分别为 1/2 和 2/3 时，采取 Hom-SVNE 和 Het-SVNE 两种方法，对虚拟网弹性能力的影响。从图中可以看出，对于同一虚拟网，当挖掘系统漏洞所占的时间比例 w_2 由 1/2 变为 2/3 时，采取 Hom-SVNE 和 Het-SVNE 方法的虚拟网瘫痪系数之差将变大，造成该结果的原因是：在攻击者花费的攻击时间不变的情况下，当挖掘系统漏洞所占的时间比例 w_2 越大，对于同构系统类型的物理服务器，攻击者在攻击成功第一个物理服务器之后，对其它物理服务器攻击所需花费的时间则越少，因此能够攻击掉的物理服务器的数量越多，那么虚拟网的瘫痪系数 ξ 将越大；但对于异构系统类型的物理服务器，攻击者对于每一个物理服务器都需要进行漏洞挖掘，因此虚拟网的瘫痪系数 ξ 不受影响。由此可以得出，挖掘系统漏洞所占的时间比例 w_2 越大，采用 Hom-SVNE 方法的虚拟网的弹性能力越小，而采用 Het-SVNE 方法的虚拟网的弹性能力不受影响。

6 结束语

本文通过分析当前同构备份式的虚拟网映射(Hom-SVNE)方法因物理服务器的同构性带来的问题，提出了一种异构备份式的虚拟网映射(Het-SVNE)方法。本文以最小化链路带宽资源开销为映射目标建立了异构备份式的虚拟网映射模型，设计了异构备份式虚拟网映射算法，并进行了仿真实验。实验结果表明，相较于现有的 Hom-SVNE 方法，Het-SVNE 方法虽在虚拟网映射成功率方面平均下降 3.51%，但在虚拟网络的弹性能力方面平均增加 19.8%(挖掘系统漏洞所占的时间比例 w_2 为 1/2)，并

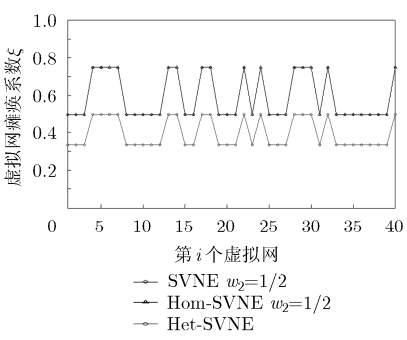

图 5 虚拟网的弹性能力对比

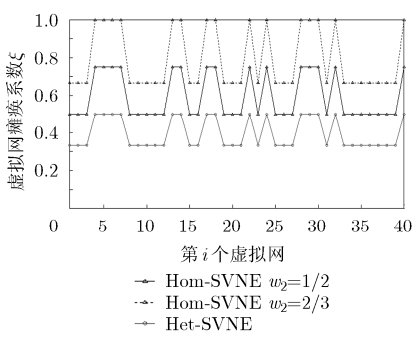

图 6 w_2 的大小对虚拟网弹性能力的影响

且随着 w_2 的增大，虚拟网弹性能力方面的优势越明显。因此，本文提出的方法在保证虚拟网络映射性能的前提下，能够较大地提高虚拟网络的弹性能力。

参 考 文 献

[1] ZHANG Q, CHENG L, and BOUTABA R. Cloud computing: state-of-the-art and research challenges[J]. *Journal of Internet Services and Applications*, 2010, 1(1): 7–18. doi: 10.1007/s13174-010-0007-6.

[2] JAIN R and PAUL S. Network virtualization and software defined networking for cloud computing: a survey[J]. *IEEE Communications Magazine*, 2013, 51(11): 24–31. doi: 10.1109/MCOM.2013.6658648.

[3] CHOWDHURY N M M K and BOUTABA R. Network virtualization: state of the art and research challenges[J]. *IEEE Communications Magazine*, 2009, 47(7): 20–26. doi: 10.1109/MCOM.2009.5183468.

[4] CHOWDHURY N M M K and BOUTABA R. A survey of network virtualization[J]. *Computer Networks*, 2010, 54(5): 862–876. doi: 10.1016/j.comnet.2009.10.017.

[5] LI Z, LIANG M, O'BRIEN L, *et al*. The cloud's cloudy moment: A systematic survey of public cloud service outage [J]. *International Journal of Cloud Computing and Services Science*, 2013, 2(5): 20–31. doi: 10.11591/closer.v2i5.5125.

[6] GILL P, JAIN N, and NAGAPPAN N. Understanding network failures in data centers: Measurement, analysis, and implications[C]. ACM SIGCOMM Computer Communication Review, New York, USA, 2011: 350–361.

[7] HERKER S, KHAN A, and AN X. Survey on survivable virtual network embedding problem and solutions[C]. International Conference on Networking and Services (ICNS), Lisbon, Portugal, 2013: 99–104.

[8] SHAHRIAR N, AHMED R, KHAN A, *et al*. ReNoVatE: recovery from node failure in virtual network embedding[C]. Network and Service Management (CNSM), Montreal, Canada, 2016: 19–27.

[9] SHAHRIAR N, AHMED R, CHOWDHURY S R, *et al*. Generalized recovery from node failure in virtual network embedding[J]. *IEEE Transactions on Network and Service Management*, 2017, 14(2): 261–274. doi: 10.1109/TNSM.2017.2693404.

[10] YU H, ANAND V, QIAO C, *et al*. Migration based protection for virtual infrastructure survivability for link failure[C]. Optical Fiber Communication Conference and Exposition (OFC/NFOEC), Los Angeles, USA, 2011: 1–3.

[11] CHOWDHURY S R, AHMED R, KHAN M M A, *et al*. Protecting virtual networks with drone[C]. Network Operations and Management Symposium (NOMS), Istanbul, Turkey, 2016: 78–86.

[12] CHOWDHURY S R, AHMED R, KHAN M M A, *et al*. Dedicated protection for survivable virtual network embedding[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(4): 913–926. doi: 10.1109/TNSM.2016.2574239.

[13] KHAN M M A, SHAHRIAR N, AHMED R, *et al*. Simple: survivability in multi-path link embedding[C]. Network and Service Management (CNSM), Barcelona, Spain, 2015: 210–218.

[14] XU J, TANG J, KWAITK, *et al*. Survivable virtual infrastructure mapping in virtualized data centers[C]. Cloud Computing (CLOUD), Honolulu, USA, 2012: 196–203.

[15] ZHANG Q, ZHANI M F, JABRI M, *et al*. Venice: reliable virtual data center embedding in clouds[C]. International Conference on Computer Communications, Toronto, Canada, 2014: 289–297.

[16] SEXTON J, STORLIE C, and NEIL J. Attack chain detection[J]. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2015, 8(5/6): 353–363. doi: 10.1002/sam.11296.

季新生：　男，1968年生，教授，博士生导师，研究方向为网络空间安全、拟态安全等.

赵　硕：　男，1992年生，硕士生，研究方向为虚拟网映射、虚拟网安全.

艾健健：　男，1989年生，博士生，研究方向为软件定义网络、网络空间安全.

程国振：　男，1986年生，助理研究员，研究方向为虚拟网映射、拟态安全等.

齐　超：　男，1991年生，博士生，研究方向为软件定义网络、拟态安全.