第 46 卷 第 2 期 计 算 机 工 程 2020 年 2 月
Vol. 46 No. 2 Computer Engineering February 2020

·热点与综述· 文章编号：1000-3428(2020)02-0021-07 文献标志码：A 中图分类号：TN918.91

一种基于冗余跳变的虚拟机动态迁移方法

孙志勇，季新生，游 伟，李英乐

（国家数字交换系统工程技术研究中心，郑州 450002）

摘 要： 在5G核心网虚拟化环境中，虚拟机共用同一物理服务器会带来一系列的安全问题，如发生侧信道攻击、虚拟节点溢出攻击等，造成用户隐私信息泄露。现有基于虚拟机动态迁移的防御方法是一种有效的主动防御技术，但虚拟机频繁迁移导致了迁移资源开销大和迁移安全性低的问题。为此，提出一种基于冗余跳变的虚拟机迁移方法，对不同虚拟机的迁移频率建立评估计算模型，在保证虚拟机隐私信息安全的前提下减小虚拟机迁移频率，对部分虚拟机采用冗余跳变的方法，以应对虚拟机频繁迁移带来的安全风险。实验结果表明，与现有虚拟机动态迁移方法相比，该方法在取得相同安全防护效果的同时，能够缩短平均迁移收敛时间并降低迁移开销。

关键词： 信息泄露；虚拟机迁移；迁移算法；冗余跳变；侧信道攻击；虚拟节点溢出攻击

开放科学（资源服务）标志码（OSID）：

中文引用格式： 孙志勇，季新生，游伟，等. 一种基于冗余跳变的虚拟机动态迁移方法[J]. 计算机工程，2020，46(2)：21-27，34.

英文引用格式： SUN Zhiyong，JI Xinsheng，YOU Wei，et al. A virtual machine dynamic migration method based on redundant transition[J]. Computer Engineering，2020，46(2)：21-27，34.

A Virtual Machine Dynamic Migration Method Based on Redundant Transition

SUN Zhiyong，JI Xinsheng，YOU Wei，LI Yingle

（China National Digital Switching System Engineering and Technological R&D Center，Zhengzhou 450002，China）

【Abstract】 In 5G core network virtualization environment，the virtual machines sharing the same physical server brings a series of problems，such as Side-Channel Attack（SCA），Virtual Node Escape Attack（VNEA） and so on，causing user private information disclosure. The existing defense method based on dynamic migration of virtual machines is an effective active defense technology，but the frequent migration of virtual machines leads to some problems，such as high resource cost and low migration security. Therefore，this paper proposes a virtual machine migration method based on redundant transition. With this method，an evaluation and calculation model is established for the migration frequency of different virtual machines. On the premise of ensuring the privacy information security of virtual machines，the migration frequency is reduced. The redundant transition method is applied to part of virtual machines to cope with the security risks brought by the frequent migration of virtual machines. Experimental results show that compared with the existing virtual machine dynamic migration method，the proposed method can reduce average migration convergence time and migration cost while maintaining the same security protection effect.

【Key words】 information leakage；virtual machine migration；migration algorithm；redundant transition；Side-Channel Attack（SCA）；Virtual Node Escape Attack（VNEA）

DOI： 10.19678/j.issn.1000-3428.0054337

0 概述

第五代移动通信系统（5G）需要适应增强移动宽带、海量机器类通信和超高可靠低时延通信三大应用场景，而不同应用场景在移动性、计费、安全、策略控制、延时和可靠性等方面具有差异化的要求，需要不同类型的网络服务。为此，5G网络引入网络功能虚拟化（Network Function Virtualization，NFV）[1]技术，通过在通用IT硬件平台上部署虚拟网络功能（Virtual Network Function，VNF）的方式，实现网元功能和专属硬件平台的解耦，并以网络切片[2]的形式向不同应用场景的用户提供灵活、多样的定制化

基金项目： 国家自然科学基金（61801515）；国家自然科学基金创新研究群体项目（61521003）；国家重点研发计划（2016YFB0801605）。

作者简介： 孙志勇（1994—），男，硕士研究生，主研方向为新一代移动通信网络技术；季新生，教授；游 伟、李英乐，讲师。

收稿日期： 2019-03-22 **修回日期：** 2019-05-06 **E-mail：** sunxdstu@163.com

服务。网络切片技术通过共用底层物理资源,能够达到资源灵活调度、资源利用率提升、网络服务按需部署的目的,但是虚拟节点共享物理资源也带来了一些安全问题,如攻击者通过侧信道攻击(Side-Channel Attack, SCA)[3-4]、虚拟节点溢出攻击(Virtual Node Escape Attack, VNEA)[5]等方式窃取用户隐私信息。

目前,针对虚拟化环境下节点信息泄露问题的研究主要分为两类。第 1 类方法对物理服务器的软硬件进行修改[6]。文献[7]提出一种 Stopwatch 架构来预防侧信道攻击,其主要解决 I/O 接口受攻击的问题。文献[8-9]通过对操作系统层进行修改和分区加密,以抵御侧信道攻击。文献[10]通过在物理主机中设置能隐藏程序运行时间的管理程序来抵御侧信道攻击,该方法主要预防特定类型的信息窃取攻击,难以应对多种类型或未知类型的攻击。第 2 类方法对虚拟机进行动态迁移,这种方法能够抵御不同类型的因共享物理资源导致的攻击。文献[11-13]利用移动目标防御[14]的思想对同一物理节点上的虚拟机进行动态迁移,通过限制共存时间来保证目标虚拟机不被恶意主机窃取隐私信息,从而增加了侧信道攻击的难度。与第 1 类方法相比,第 2 类方法不需要修改物理服务器的软硬件,但虚拟机的动态迁移受制于迁移算法、迁移开销和网络环境,并且虚拟机动态迁移过程存在迁移失败的风险。文献[15]设计了一个简单的攻击实验,证明在虚拟化环境下虚拟机进行动态迁移的过程容易受到网络攻击。文献[16]指出虚拟机在迁移过程中会削弱或抵消迁出节点虚拟机入侵防御系统和入侵检测系统的作用。文献[17]介绍了在虚拟机迁移时攻击者针对系统控制平面、数据平面和迁移控制模块的非法攻击手段、过程以及后果。文献[18]指出了内部人员利用虚拟机迁移的相关漏洞获取用户数据的可能性。

现有虚拟机迁移方法大多忽略了迁移本身带来的风险,尤其在移动核心网中的高可靠通信场景下,频繁迁移引起的安全问题不容忽视。本文对迁移频率过快的虚拟节点采用冗余跳变的工作机制,以解决虚拟节点的信息泄露问题。在此基础上,建立虚拟机迁移频率计算模型并提出一种基于冗余跳变的虚拟机迁移方法,以抵御信息窃取攻击并最小化迁移资源开销。

1 信息泄露问题与解决方法

1.1 网络模型

基于虚拟化技术的网络切片部署过程可抽象为服务功能链(Service Function Chain, SFC)的映射过程[19-20],其中,实现网络服务的一组网络功能集合被称为 SFC[21]。服务提供商将 SFC 动态实例化至通用的底层物理设施网络上以向用户提供服务[22]。SFC 映射过程及网络结构组成如下:

1)底层物理网络。底层物理网络由通用的硬件设备平台组成,用一个赋权无向图 $G^s=(N^s,L^s)$ 表示,其中,N^s 表示物理服务器的集合,L^s 表示物理链路的集合。对于每一个物理服务器 $u\in N^s$,都有一个可用的 CPU 计算处理能力,其值大小用 $C(u)$ 表示。同样,对于每个物理链路 $(u,v)\in L^s$,都有一个可用的物理带宽资源,用 $B(u,v)$ 表示。

2)服务功能链。在虚拟化云环境中实现网络服务需要一组有序的网络功能 VNF 组成 SFC,采用赋权有向图 $G^v=(N^v,L^v)$ 表示 SFC 中全部 VNF 节点及其关系的逻辑视图,其中,N^v 表示 VNF 的逻辑节点集合,L^v 表示逻辑链路集合。

3)SFC 映射。系统的资源管理和编排模块根据 SFC 的请求信息和底层物理资源状况,完成映射 $f:G^v\rightarrow G^s$,根据映射过程中的指标需求设置约束条件和目标函数,然后设计相应的虚拟映射算法,求解并找到满足映射需求的 VNF 和虚拟链路对应的最优位置[23]。

4)虚拟机迁移。迁移是指将虚拟机从一个主机或存储位置移至另一个主机或存储位置的过程。虚拟节点 $\bar{u}$ 在 t 时刻部署在物理节点 u 上,在 $t+T$ 时刻,虚拟节点 $\bar{u}$ 需要迁移至另一物理节点 u' 工作,同时与虚拟节点 $\bar{u}$ 邻接的虚拟链路 $(\bar{u},\bar{v})$ 也要进行迁移。根据指定的目标函数设计相应的迁移算法 $M:G^v\rightarrow G^{s'}$,在空间维度找到 VNF 和虚拟链路所应迁移到的最优物理位置。

1.2 虚拟节点信息泄露问题描述

在虚拟化环境下,服务功能链共享底层物理基础设施,不同虚拟链路的虚拟节点可能映射到同一物理节点,如图 1 所示,左侧 2 条服务功能链 G_1^v、G_2^v 映射至右侧底层物理网络,虚拟节点 b、e 共用物理节点 B,虚拟节点 c、f 共用物理节点 D。在图 1 中,小方框中的数字表示节点的 CPU 计算处理能力值。

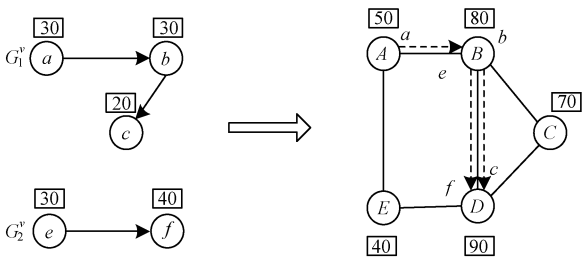

图 1 服务功能链共享物理资源示例

Fig. 1 Example of shared physical resources of service function chain

攻击者利用运行在同一物理节点上的虚拟机,可以发动攻击窃取用户信息,其中,主要攻击类型是侧信道攻击和虚拟节点溢出攻击。将图 1 中的物理

节点 B、D 放大,分别示例 2 种攻击过程。侧信道攻击过程如图 2(a)所示:1)目标节点 b 对共用的 CPU cache、内存总线等物理节点资源发生作用;2)恶意虚拟节点 e 通过共用的物理节点资源对虚拟节点 b 进行探测;3)根据探测结果完成 1 bit 信息的隐蔽传输,然后不断重复上述过程。虚拟节点溢出攻击过程如图 2(b)所示:1)目标节点 f 上运行的自定义协议存在漏洞,恶意用户向其发送特定报文;2)目标节点 f 崩溃退出,向管理程序发送事件消息,进而触发漏洞或注入恶意代码;3)管理程序被控制,从而泄露其他虚拟节点的信息。

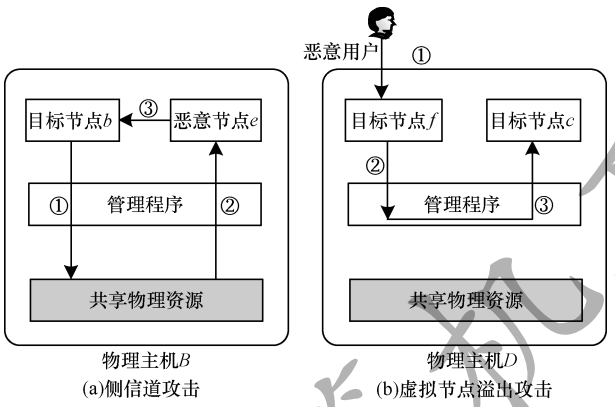

图 2　2 种虚拟节点信息泄露过程示例

Fig. 2　Example of information disclosure process of 2 kinds of virtual nodes

在虚拟节点信息被攻击者窃取的过程中,主要存在如表 1 所示的 3 个安全影响因素[24]。

表 1　虚拟节点安全影响因素

Table 1　Influencing factors of virtual node security

符号	含义
S	虚拟节点信息泄露速率
ε	最小时间间隔
Γ	共存时间间隔数量

因此,为了避免攻击者发动侧信道攻击并成功窃取用户隐私信息,系统需要满足式(1)。

$$S\Gamma\varepsilon \leqslant I \tag{1}$$

其中,I 表示信息被攻击者成功窃取所需的最小信息量。

1.3　基于冗余跳变的虚拟机迁移方法

随着服务功能链中虚拟机数量的增多,部分虚拟机的 CPU 资源和隐私信息承载量较大,迁移需要达到较高频率才能有效抵御信息窃取攻击。但是,对所有虚拟机不断迁移会带来开销过大和算法收敛时间过长的问题,这不仅增大了网络资源消耗,同时迁移失败概率和迁移安全风险也会加大。在移动通信网络中,需要对用户隐私进行保护,但同时应该保证移动通信网络的高可靠性要求。本文考虑迁移本身的安全风险,提出一种基于冗余跳变的虚拟机迁移方法。在现有迁移方法的基础上,以承载隐私信息量和虚拟机 CPU 资源大小作为评判标准,将虚拟机可能泄露信息的速率进行排序,根据用户需求按等级分类处理,从而缩短虚拟机迁移算法的收敛时间并降低迁移开销。

以承载隐私信息量和虚拟机 CPU 资源大小作为评判标准,将虚拟机可能泄露信息的速率进行排序并根据用户需求按等级分类处理。将虚拟机隐私信息泄露速率定义为:

$$D = kIC \tag{2}$$

其中,k 为常数,I 为虚拟机承载的隐私信息量,C 为虚拟机调用的 CPU 资源。根据 D 的大小对不同虚拟机采用不同的迁移频率,虚拟机的迁移周期为:

$$T = \frac{1}{D} \tag{3}$$

对于共享物理基础设施资源的虚拟机,为避免用户隐私信息泄露,在达到限制的共存时间后即进行迁移,参数 D 越大,迁移的周期越短,迁移频率也越高,网络中可能存在一部分虚拟机的参数 D 过大,需要迁移的周期很短,但迁移频率过高会给网络带来较高的安全风险。对于这部分虚拟机,如图 3 中的 b 节点,本文在映射阶段先将其冗余备份为节点 b',在网络部署完成后,相同功能的节点以迁移周期 T 进行交替跳变工作,既避免了因共存时间过长引起的隐私泄露问题,也解决了虚拟机频繁迁移带来的开销和风险。在图 3 中,箭头上的数字表示链路带宽资源值。

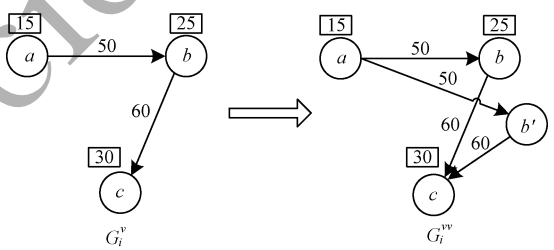

图 3　节点冗余备份示例

Fig. 3　Example of node redundant backup

2　基于冗余跳变的虚拟机映射与迁移模型

本节基于上述服务功能链部署网络模型,对虚拟机设计具体的映射约束条件和映射目标约束,对虚拟机迁移设置具体的迁移条件和迁移目标约束,并提出可在计算机上实现的虚拟机映射算法。

2.1　虚拟机映射模型与算法

2.1.1　备份映射模型

虚拟机映射即将服务功能链部署到物理底层网络以提供网络服务的过程,此过程需要考虑 3 个因素:采用冗余跳变工作方式的虚拟机的冗余备份,映射过程需满足底层物理节点和链路资源约束,映射

的目标函数。变量说明如下：

1）$\alpha_u^{\bar{u}_i}$：当虚拟节点 $\bar{u}_i$ 映射到物理节点 u 上时，$\alpha_u^{\bar{u}_i}$ 为 1，否则为 0。

2）$\beta_{(u,v)}^{(\bar{u}_i,\bar{v}_i)}$：当虚拟链路映射到物理链路 (u,v) 上时，$\beta_{(u,v)}^{(\bar{u}_i,\bar{v}_i)}$ 为 1，否则为 0。

约束条件如下：

1）虚拟节点映射约束：

$$\sum_{u\in N^s}\alpha_u^{\bar{u}_i}=1,\ \forall\ \bar{u}_i\in N_i^{vv},\ \forall\ G_i^{vv}\in G^{vv} \tag{4}$$

$$\sum_{\bar{u}_i\in N_i^{vv}}\alpha_u^{\bar{u}_i}\leqslant 1,\ \forall\ u\in N^s,\ \forall\ G_i^{vv}\in G^{vv} \tag{5}$$

$$\sum_{G_i^{vv}\in G^{vv}}\sum_{\bar{u}_i\in N_i^{vv}}G(\bar{u}_i)\times\alpha_u^{\bar{u}_i}\leqslant C(u),\ \forall\ u\in N^s \tag{6}$$

式(4)表示 SFC 中一个虚拟节点只能被映射至一个物理节点；式(5)表示一个 SFC 中的不同虚拟节点不能被映射至一个物理节点；式(6)表示一个物理节点上承载的所有虚拟节点的 CPU 资源需求之和不能大于该物理节点的 CPU 资源。

2）虚拟链路映射约束：

$$\sum_{G_i^{vv}\in G^{vv}}\sum_{(\bar{u}_i,\bar{v}_i)\in L_i^{vv}}B(\bar{u}_i,\bar{v}_i)\times\beta_{(u,v)}^{(\bar{u}_i,\bar{v}_i)}\leqslant B(u,v)$$
$$\forall\ (u,v)\in L^s \tag{7}$$

$$\sum_{v\in N(u)}\beta_{(u,v)}^{(\bar{u}_i,\bar{v}_i)}-\sum_{v\in N(u)}\beta_{(v,u)}^{(\bar{u}_i,\bar{v}_i)}=\alpha_u^{\bar{u}_i}-\alpha_u^{\bar{v}_i}$$
$$\forall\ \bar{u}_i,\bar{v}_i\in N_i^{vv},\ \forall\ u\in N^s,\ \forall\ G_i^{vv}\in G^s \tag{8}$$

$$\sum_{(\bar{u}_i',\bar{v}')\in L_i'^{vv}}\sum_{v\in N(u)}\beta_{(u,v)}^{(\bar{u}_i',\bar{v}')}=0$$

$$\alpha_u^{\bar{u}_i}=1,\ \forall\ u\in N^s,\ \forall\ G_i^{vv}\in G^v \tag{9}$$

式(7)表示部署至一条物理链路上的虚拟链路带宽总和小于该目标物理链路负载量；式(8)表示虚拟链路应满足流约束限制；式(9)表示与备份节点 $\bar{u}_i'$ 相邻的虚拟链路 $(\bar{u}_i',\bar{v}_i')$ 不能经过虚拟节点 $\bar{u}_i$ 所在的物理节点。

在满足上述 SFC 资源请求的条件下，目标函数为映射成本最小，具体如下：

$$\min P(G_i^v)=\delta_t\sum_{\bar{u}_i\in N_i^{vv}}\alpha_{u_i}^{\bar{u}_i}C(\bar{u}_i)+\varphi_B\sum_{(\bar{u}_i,\bar{v}_i)\in L_i^{vv}}\beta_{(u_i,v_i)}^{(\bar{u}_i,\bar{v}_i)}B(u_i,v_i) \tag{10}$$

其中，δ_t 和 ϕ_B 分别表示虚拟机和虚拟链路的单位资源映射开销。SFC 映射成本主要有虚拟节点映射的资源开销和虚拟链路映射的物理链路带宽资源开销两部分。

2.1.2 备份映射算法

本文基于冗余跳变的虚拟机映射算法描述如下：

算法 1 基于冗余跳变的虚拟机映射算法

输入 服务功能链 $G_i^v(N_i^v,L_i^v)$，底层物理网络 $G^s(N^s,L^s)$

输出 虚拟节点映射方案 $f:G^v\rightarrow G^s$

1. 根据隐私信息泄露速率 D 的大小对虚拟节点进行降序排序，用集合 N_i^v 表示；

2. 根据用户和系统对网络稳定性的需求，选取集合 N_i^v 序列前合适比例虚拟机节点 N_i^{v-d}，冗余备份为 $N_i^{v-d'}$，得到服务功能链 $G_i^{vv}(N_i^{vv},L_i^{vv})$；

3. for $\bar{u}\in N_i^{vv}$ do

$\tilde{u}$. mapping；//对隐私信息泄露速率 D 大的虚拟机优先//进行映射

4. if u. ability $\geqslant\bar{u}_i$. requirement, then //物理节点是否//满足虚拟机映射资源约束条件，物理资源多的节点优先//进行映射

5. continue

6. end if

7. $\bar{u}_i$. mapping = true and break

8. if $\bar{u}_i$. mapping = true then reject G_i^v and return //虚拟//节点映射失败，重新映射

9. end if

10. end for

11. for each $(\bar{u}_i,\bar{v}_i)\in L_i^{vv}$ do

12. $(\bar{u}_i,\bar{v}_i)$. mapping；

13. for each $(u,v)\in L^s$ do

14. P = {path | (u, v). ability $\geqslant(\bar{u}_i,\bar{v}_i)$. requirement, $(u,v)\in L^s$}//找到满足虚拟链路 $\bar{u}_i$ 的所有物理路径的集合

15. $\beta_{P[i]}^{(\bar{u}_i,\bar{v}_i)}$//满足目标函数的底层物理映射最优路径 P[i]

16. $(\bar{u}_i,\bar{v}_i)$. mapping = true and break

17. if $(\bar{u}_i,\bar{v}_i)$. mapping = false then reject G_i^i and return

18. end if

19. end for

20. end for

在映射过程中，将隐私信息泄露速率 D 较大的虚拟节点优先映射到物理资源大的物理节点上，以提高节点的映射成功率，该过程利用贪婪算法实现。为使映射资源开销最小，在对虚拟链路进行映射时本文采用最短路径算法选择目标物理链路。

2.2 虚拟机迁移模型与算法

2.2.1 虚拟机迁移模型

工作在同一个物理主机上的虚拟机在达到共存时间阈值后需要迁移至另一个物理服务器工作，迁移过程中主要考虑 3 个因素：虚拟机的共存时间是否超过阈值，迁移过程中底层物理节点和链路的资源约束，迁移的目标函数。变量说明如下：

1）$\alpha_{G^v}^t(\bar{u},u)$：当服务功能链 G^v 的虚拟节点 $\bar{u}$ 在 t 时刻映射在物理节点 u 上时，$\alpha_{G^v}^t(\bar{u},u)$ 为 1，否则为 0。

2）$\beta_{G^v}^t[(\bar{u},\bar{v}),(u,v)]$：在 t 时刻如果服务功能链 G^v 的虚拟链路 $(\bar{u},\bar{v})$ 映射在物理链路 (u,v) 上，$\beta_{G^v}^t[(\bar{u},\bar{v}),(u,v)]$ 为 1，否则为 0。

虚拟机进行迁移时需要满足式(4)~式(8)的约束条件,此外,为了保护用户隐私信息安全,需要虚拟机之间满足共存的时间约束,服务功能链 G_i^v 的虚拟机 $\bar{u}_i$ 和服务功能链 G_j^v 中的虚拟机 $\bar{u}_j$ 共存的时间应小于允许的共存时间阈值 T:

$$\varepsilon\sum_{\lambda=1}^{\Gamma}\sum_{u\in N^s}\alpha_{G_i^v}^{\lambda\varepsilon}(\bar{u}_i,u)\alpha_{G_j^v}^{\lambda\varepsilon}(\bar{u}_j,u)<T$$
$$\forall G_i^v,G_j^v\in G^v,i\neq j,\bar{u}_i\in N_i^v,\bar{u}_j\in N_j^v \tag{11}$$

其中,$t=\lambda\varepsilon,\lambda\in[1,\Gamma]$,$\varepsilon$ 为最小时间间隔。

在对虚拟机进行迁移时,在虚拟节点之间共存时间约束下,本文以最小化迁移资源开销作为目标函数。迁移开销为迁移过程中虚拟节点和虚拟链路的迁移开销总和:

$$\min\left\{\delta\sum_{i=1}^{k}\sum_{\bar{u}\in N^v}\sum_{u\in N^s}C(\bar{u})\alpha_{G_i^v}^{t}(\bar{u},u)(1-\alpha_{G_i^v}^{t+\Gamma}(\bar{u},u))+\varphi\sum_{i=1}^{k}\sum_{(\bar{u},\bar{v})\in L^v}\sum_{(u,v)\in L^s}B(\bar{u},\bar{v})\beta_{G_i^v}^{t}\cdot[(\bar{u},\bar{v}),(u,v)](1-\beta_{G_i^v}^{t+\Gamma}[(\bar{u},\bar{v}),(u,v)])\right\} \tag{12}$$

其中,δ 为虚拟节点单位资源迁移开销,φ 为虚拟链路单位资源迁移开销。式(12)的第 1 项是虚拟节点迁移的资源开销,第 2 项是虚拟链路迁移的资源开销。

2.2.2　虚拟机迁移算法

本文基于冗余跳变的虚拟机迁移算法描述如下:

算法 2　基于冗余跳变的虚拟机迁移算法

输入　服务功能链 $G_i^v(N_i^v,L_i^v)$,底层物理网络 $G^s(N^s,L^s)$,目前映射状态 $f:G^v\to G^s$

输出　虚拟机迁移方案 $M:G^{v1}\to G^s$

1. for each $\bar{u}\in N_i^v$ do
2. if $\bar{u}\in N_s^{vd},N_i^{vd}=N_i^{v-d}\cup N_i^{v-d'}$, do//对采用冗余跳变方//案的节点,检查其共存时间,以共存约束时间 T 为周期进行//跳变工作
3. if $\varepsilon\sum_{\lambda=1}^{\Gamma}\sum_{u\in N^s}\alpha_{G_i^v}^{\lambda\varepsilon}(\bar{u}_i,u)\alpha^{\lambda\varepsilon}(\bar{u}_j,u)\geqslant T$ then//虚拟机之间//共存的时间达到 $\bar{u}$ 的阈值 T
4. hop($\bar{u}$); //虚拟机 $\bar{u}$ 跳变到备份物理节点工作
5. end if
6. end if
7. else if $\varepsilon\sum_{\lambda=1}^{\Gamma}\sum_{u\in N^s}\alpha_{G_i^v}^{\lambda\varepsilon}(\bar{u}_i,u)\alpha^{\lambda\varepsilon}(\bar{u}_j,u)\geqslant T$ then
8. migrate($\bar{u}$); //对虚拟节点 $\bar{u}$ 进行迁移
9. 将所有满足迁移条件的物理节点放入集合 N 中
10. for(i=0;i≤length(N);i++)do
11. if migrate_node(N[i])==true&migrate_links(N[i])==true then//物理节点、物理链路是否满足资源//需求
12. migratest_cost($\bar{u}$)=min{migrate_cost(N)}
//迁移到产生最小开销的物理节点上
13. end if
14. end for
15. end if
16. end for

在对服务功能链迁移的过程中,首先遍历所有已映射的虚拟节点,检查虚拟机在物理主机已工作时间是否超过该虚拟节点的共存时间约束值 T,超过时标记该节点时间共存值为 1,否则为 0。对所有节点检查完后,在满足资源需求的条件下将时间共存值为 1 的虚拟机迁移至开销最小的物理节点上。

3　实验结果与分析

3.1　实验环境设置

本文实验在 Inter(R) Core(TM) i7-8750 CPU 2.2 GHz、8 GB RAM 的计算机上进行,利用 GT-ITM 工具生成服务功能链和物理网络拓扑,使用 Matlab 软件编程实现服务功能链映射算法和迁移算法,最后分析仿真结果。

底层网络物理主机设为 15 个,物理链路为 24 条。物理网络中的节点计算资源($300\leqslant C\leqslant 500$)、链路的带宽资源($150\leqslant B\leqslant 350$)为固定值。每条 SFC 中虚拟节点数目服从[4,6]的整数随机均匀分布。SFC 平均每 10 个单位时间发送一次映射请求,SFC 的生命周期服从参数为 40 的指数分布。虚拟节点的计算资源需求服从[40,140]区间的均匀分布,带宽资源需求服从[20,120]区间的均匀分布,迁移节点的平均共存时间阈值设为 10 个单位时间。

3.2　结果分析

将本文基于冗余跳变的虚拟机迁移方法与文献[12]提出的一般虚拟机迁移方法作对比,本文方法记为 Backup,一般虚拟机迁移方法记为 General。由上文可知,本文方法中所有虚拟机与现有一般动态迁移方法中所有虚拟机具有相同的共存时间约束,即与一般动态迁移方法相比,本文方法可取得相同的信息安全防护效果。下文在虚拟机映射和迁移的资源开销、映射成功率、迁移算法收敛时间和迁移失效率等 5 个方面进行仿真,并对比分析仿真结果。

1)虚拟机映射资源开销。如图 4 所示,相比一般迁移方法,本文 20% 冗余跳变的方法(指选取迁移频率前 20% 的虚拟节点进行冗余备份)映射平均资源消耗增加了 21.81% ,10% 冗余跳变的方法(指选取迁移频率前 10% 的虚拟节点进行冗余备份)映射平均资源消耗增加了 10.91% 。本文方法对部分频繁迁移的节点采取冗余跳变的工作方式,且冗余备份节点的映射增加了虚拟机映射资源的开销。

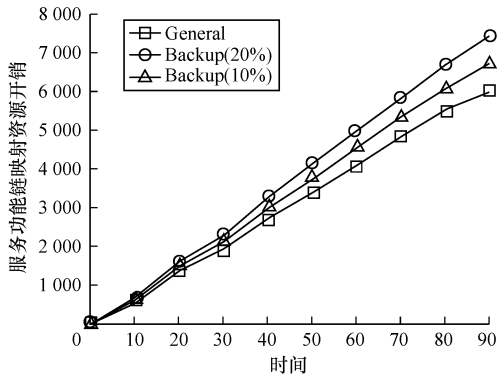

图 4 服务功能链映射开销对比

Fig. 4 Cost comparison of service function chain mapping

2)虚拟机迁移资源开销。如图 5 所示,相比一般迁移方法,本文 20% 冗余跳变的方法迁移平均资源消耗减少了 25.27% ,10% 冗余跳变的方法迁移平均资源消耗减少了 12.63% ,原因是进行跳变工作的虚拟节点不需要再寻找物理节点进行迁移,降低了虚拟机迁移开销。

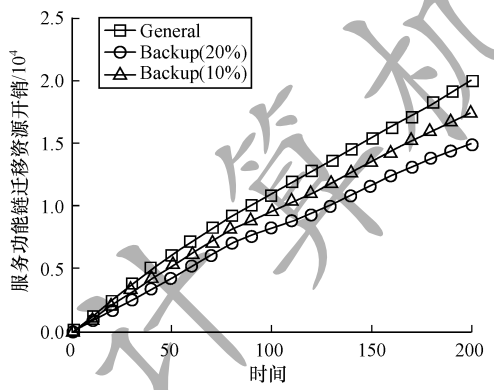

图 5 服务功能链迁移资源开销

Fig. 5 Resource overhead of service function chain migration

3)映射成功率。图 6 显示了不同迁移方式下服务功能链映射成功率随时间的变化情况。从图 6 可以看出,相比一般迁移映射方法,本文 10% 冗余跳变的方法备份成功率平均降低 4.22% ,20% 冗余跳变的方法平均降低 9.73% 。由于基于冗余跳变的迁移方法有较多备份节点需要映射,因此映射成功率较低。

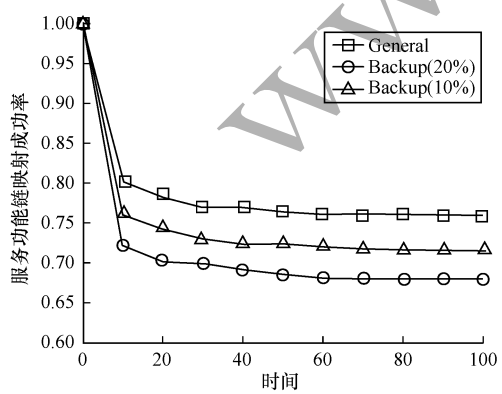

图 6 服务功能链映射成功率

Fig. 6 Success rate of service function chain mapping

4)迁移算法收敛时间。图 7 所示为一般迁移方法和 20% 冗余跳变方法的迁移算法收敛时间对比,从图 7 可以看出,20% 冗余跳变方法的迁移算法收敛时间比一般迁移方法平均减少了 24.79% 。由于冗余跳变方法部分节点不需要迁移算法进行节点的迁移,因此本文方法可明显减少迁移算法的收敛时间。在实际网络中,因为频繁迁移需要跳变工作的虚拟机占比较小,所以不会对网络映射资源开销、迁移资源开销、映射成功率以及迁移算法收敛时间产生太大影响。

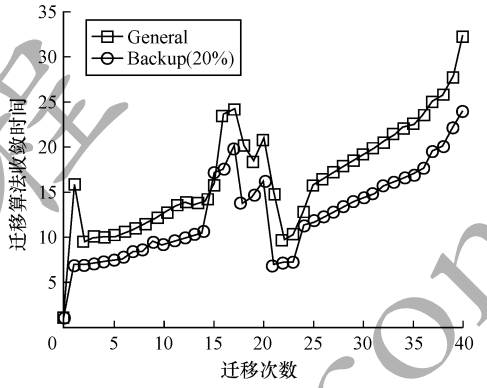

图 7 迁移算法收敛时间

Fig. 7 Convergence time of migration algorithm

5)迁移成功率。在仿真中设 SFC 所有节点、30% 冗余备份节点、20% 冗余备份节点和 10% 冗余备份节点的平均节点迁移成功率分别为 99.9% 、98.0% 、99.0% 、99.5% ,其中,迁移失败包括迁移过程失败和迁移过程中系统被攻击成功 2 种情况。仿真结果如图 8 所示,从图 8 可以看出,在 100 个单位时间内,服务功能链采用一般迁移方法、10% 节点冗余跳变方法、20% 节点冗余跳变方法、30% 节点冗余跳变方法的迁移成功率分别为 65.53% 、80.20% 、88.68% 、93.24% ;在 200 个单位时间内,各方法服务功能链迁移成功率分别为 42.94% 、64.32% 、78.64% 、86.93% 。因此,可得出如下结论:对迁移高风险虚拟机进行跳变工作,迁移失效概率将显著降低。在实际网络中,虚拟机迁移失败风险主要由少部分的高风险虚拟机引发,由实验结果可知,本文方法可明显降低网络中虚拟机迁移失败的概率。

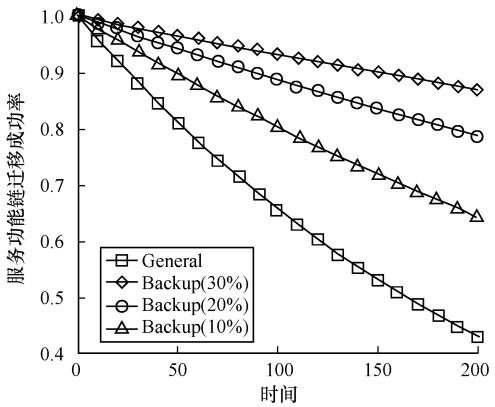

图 8 服务功能链迁移成功率

Fig. 8 Success rate of service function chain migration

4　结束语

核心网引入虚拟化技术使得未来电信设备不再使用专用硬件设备,而是通过软件下发到统一的硬件上实现网络功能,这种方式能够提高资源利用率,但虚拟节点共享物理资源也带来了信息泄露的安全隐患。本文考虑主动防御技术的动态迁移方法中迁移本身存在的安全风险,提出一种基于冗余跳变的虚拟机迁移方法。通过建立虚拟机隐私信息泄露速率计算模型,得到各虚拟机的迁移周期。根据用户需求和网络环境选择不同比例的虚拟机进行冗余备份映射,采取冗余跳变的工作方式解决虚拟机之间共存时间过长带来的信息泄露问题,剩余虚拟机则在达到共存时间阈值后进行动态迁移。实验结果表明,该方法虽然增加了映射开销并降低了映射成功率,但减少了迁移资源开销和迁移算法收敛时间,并且在满足 SFC 服务性能要求的同时显著降低了网络中虚拟机迁移失败的概率。但是,如果频繁迁移的虚拟机占比较大,本文方法将存在资源开销较大的问题,因此,下一步将研究一种兼顾资源开销与网络安全性的防护方法。

参考文献

[1] HERRERA J G, BOTERO J F. Resource allocation in NFV:a comprehensive survey[J]. IEEE Transactions on Network and Service Management,2016,13(3):518-532.

[2] ORDONEZ-LUCENA J,AMEIGEIRAS P,LOPEZ D,et al. Network slicing for 5G with SDN/NFV: concepts, architectures,and challenges[J]. IEEE Communications Magazine,2017,55(5):80-87.

[3] LIU F F,YAROM Y,GE Q,et al. Last-level cache side-channel attacks are practical[C]//Proceedings of 2015 IEEE Symposium on Security and Privacy. Washington D. C., USA:IEEE Press,2015:605-622.

[4] IRAZOQUI G, EISENBARTH T, SUNAR B. S$A: a shared cache attack that works across cores and defies VM sandboxing-and its application to AES[C]//Proceedings of 2015 IEEE Symposium on Security and Privacy. Washington D. C.,USA:IEEE Press,2015:591-604.

[5] KELLER E, SZEFER J, REXFORD J, et al. NoHype: virtualized cloud infrastructure without the virtualization[J]. ACM SIGARCH Computer Architecture News,2010,38(3): 350-361.

[6] PATTUK E, KANTARCIOGLU M, LIN Z Q, et al. Preventing cryptographic key leakage in cloud virtual machines [C]//Proceedings of USENIX Security Symposium. San Diego, USA: USENIX Association, 2014:703-718.

[7] NAHAPETIAN A. Side-channel attacks on mobile and wearable systems[C]//Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference. Washington D. C. ,USA:IEEE Press,2016: 243-247.

[8] VATTIKONDA B C,DAS S,SHACHAM H. Eliminating fine grained timers in Xen[C]//Proceedings of the 3rd ACM Workshop on Cloud Computing Security. New York, USA:ACM Press,2011:41-46.

[9] WU Jingzheng, DING Liping, LIN Yuqi, et al. Xenpump:a new method to mitigate timing channel in cloud computing [C]//Proceedings of 2012 IEEE International Conference on Cloud Computing. Washington D. C. ,USA:IEEE Press,2012:678-685.

[10] VARADARAJAN V, RISTENPART T, SWIFT M M. Scheduler-based defenses against cross-VM side-channels [C]//Proceedings of USENIX Security Symposium. San Diego, USA: USENIX Association, 2014:687-702.

[11] HAN Y, CHAN J, ALPCAN T, et al. Using virtual machine allocation policies to defend against co-resident attacks in cloud computing[J]. IEEE Transactions on Dependable and Secure Computing, 2017, 14 (1): 95-108.

[12] MOON S J,SEKAR V,REITER M K. Nomad:mitigating arbitrary cloud side channels via provider-assisted migration[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York,USA:ACM Press,2015:1595-1606.

[13] ZHANG Yulong, LI Min, BAI Kun, et al. Incentive compatible moving target defense against VM-colocation attacks in clouds[C]//Proceedings of IFIP International Information Security Conference. Berlin, Germany: Springer,2012:388-399.

[14] EVANS D, NGUYEN-TUONG A, KNIGHT J. Effectiveness of moving target defenses[M]. Berlin, Germany: Springer,2011:29-48.

[15] CHEN Yidan, LI Taoshen. Security problem analysis of virtual machine live migration in cloud computing environment[J]. Computer Technology and Development, 2015,25(12):114-117. (in Chinese)
陈怡丹,李陶深. 云计算环境下虚拟机动态迁移的安全问题分析[J]. 计算机技术与发展,2015,25(12): 114-117.

[16] NAVAMANI B,YUE C,ZHOU X B,et al. An analysis of the virtual machine migration incurred security problems in the cloud[J]. Practical Radiation Oncology, 2015(1):71-79.

[17] OBERHEIDE J, COOKE E, JAHANIAN F. Empirical exploitation of live virtual machine migration[EB/OL]. [2019-03-02]. http://www. orkspace. net/secdocs/Conferences/BlackHat/Federal/2008/Exploiting%20Live%20Virtual %20Machine% 20Migration-paper. pdf.

[18] DUNCAN A,CREESE S,GOLDSMITH M,et al. Cloud computing: insider attacks on virtual machines during migration [C]//Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Washington D. C., USA:IEEE Press,2013:493-500.

[19] CHOYI V K, ABDEL-HAMID A, SHAH Y, et al. Network slice selection, assignment and routing within 5G networks [C]//Proceedings of 2016 IEEE Conference on Standards for Communications and Networking. Washington D. C. ,USA:IEEE Press,2016: 1-7.

(下转第 34 页)

(上接第 27 页)

[20] CHATRAS B, KWONG U S T, BIHANNIC N. NFV enabling network slicing for 5G[C]//Proceedings of IEEE Innovations in Clouds, Internet and Networks. Washington D. C., USA: IEEE Press, 2017: 219-225.

[21] HAN B, GOPALAKRISHNAN V, JI L S, et al. Network function virtualization: challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2015, 53(2): 90-97.

[22] BASTA A, KELLERER W, HOFFMANN M, et al. A virtual SDN-enabled LTE EPC architecture: a case study for S-/P-gateways functions[C]//Proceedings of 2013 IEEE SDN for Future Networks and Services. Washington D. C., USA: IEEE Press, 2013: 1-7.

[23] WAN Yue, CHEN Xiuhong, HE Jiajia. Local spectral clustering mapping algorithm using sparse autoencoders[J]. Transducer and Microsystem Technologies, 2018, 37(1): 145-148, 153. (in Chinese)
万月, 陈秀宏, 何佳佳. 利用稀疏自编码的局部谱聚类映射算法[J]. 传感器与微系统, 2018, 37(1): 145-148, 153.

[24] ZHAO Shuo, JI Xinsheng, MAO Yuxing, et al. Research on dynamic migration of virtual machine based on security level[J]. Journal on Communications, 2017, 38(7): 165-174. (in Chinese)
赵硕, 季新生, 毛宇星, 等. 基于安全等级的虚拟机动态迁移方法[J]. 通信学报, 2017, 38(7): 165-174.

编辑　吴云芳