

基于隔离等级的网络切片部署方法

潘启润, 黄开枝, 游伟

(战略支援部队信息工程大学, 河南 郑州 450001)

摘要: 为了兼顾网络切片的性能隔离需求和安全隔离需求, 提出了一种基于隔离等级的网络切片部署方法。该方法首先从性能隔离和安全隔离两方面确定了网络切片实例的隔离等级, 在合适的位置部署虚拟节点时, 不但保证所有网络切片实例均能达到各自的性能水平和安全水平, 而且从隔离等级差值入手对虚拟节点共存的条件进行限制; 然后利用整数线性规划方法对该问题进行建模, 把部署成本最小化作为目标函数; 最后使用基于遗传算法改进的粒子群算法求出最终的部署结果。仿真结果表明, 该方法具有较低的部署成本和较高的收益开销比, 并且能够同时保证性能和安全两方面的需求。

关键词: 网络切片; 隔离; 性能下降; 侧信道攻击

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2020003

Network slicing deployment method based on isolation level

PAN Qirun, HUANG Kaizhi, YOU Wei

Strategic Support Force Information Engineering University, Zhengzhou 450001, China

Abstract: In order to balance the performance isolation requirements and security isolation requirements of network slicing, a network slice deployment method based on isolation level was proposed. The method first determined the isolation level of the network slice instance from the aspects of performance isolation and security isolation. When deploying the virtual nodes in the appropriate location, not only ensured that all network slice instances could reach their respective performance levels and security levels, but also isolated them. The gradation difference started to limit the coexistence condition of the virtual node. Then the integer linear programming method was used to model the problem, and the deployment cost was minimized as the objective function. Finally, the particle swarm optimization algorithm based on genetic algorithm was used to find the final deployment result. The simulation results show that the method has lower deployment cost and higher benefit-to-cost ratio, and can guarantee both performance and security.

Key words: network slicing, isolation, performance degradation, side channel attack

收稿日期: 2019-04-02; 修回日期: 2019-06-06

通信作者: 潘启润, pangqirun03@163.com

基金项目: 国家重点研发计划网络空间安全专项基金资助项目 (No.2016YFB0801605); 国家自然科学基金创新群体基金资助项目 (No.61521003); 国家自然科学基金资助项目 (No.61801515)

Foundation Items: The National Key R&D Program Cyberspace Security Project (No.2016YFB0801605), The National Natural Science Foundation Innovation Group Project (No.61521003), The National Natural Science Foundation of China (No.61801515)

论文引用格式: 潘启润, 黄开枝, 游伟. 基于隔离等级的网络切片部署方法[J]. 网络与信息安全学报, 2019, 6(2): 96-105.
PAN Q R, HUANG K Z, YOU W. Network slicing deployment method based on isolation level[J]. Chinese Journal of Network and Information Security, 2019, 6(2): 96-105.

1 引言

5G 作为新一代移动通信技术的发展方向,不但要满足海量物联网的需求,而且要能够深度融合工业、医疗、交通等行业,实现真正的“万物互联”。5G 有三大应用场景——增强型移动带宽(eMBB)、海量机器类通信(mMTC)、超可靠低时延通信(uRLLC)。传统网络无法满足不同应用场景的多样化服务需求,故提出了网络切片(NS, network slicing)的概念。网络切片本质上是一个虚拟网络,多个虚拟网络部署在相同的物理基础设施上,每个虚拟网络为不同的用户提供端到端的定制化服务^[1],这个定制化的虚拟网络称之为网络切片实例(NSI, network slice instance)。每个网络切片实例是一个相对独立的网络域,具备各自的网络资源和业务信息。但是,不同的虚拟网络共享底层物理资源,使不同切片间可能存在无意或恶意的干扰行为,造成性能下降或信息泄露。

一方面,多个网络切片过度竞争底层硬件资源,造成虚拟节点(VM)的性能下降。为了应对该问题,文献[2]提出了一种基于层次聚类和优化搜索的功耗感知算法,从资源利用率的角度量化了 VM 间的资源使用情况。文献[3]基于 Xen IO 机制和 VCPU 调度机制设计了一个性能评估模型,根据该模型估计 VM 迁移前以及迁移后的性能下降程度,并提出了一种轻量级的干扰感知的 VM 在线迁移方法。文献[4]设计了一种 VM 整合方法,该方法不但可以保证 VM 的性能在可接受的范围内下降,而且最大限度减少耗费的资源。文献[5]对 VM 部署问题进行了研究,当使用提出的方法把多个 VM 放置到一个服务器上时,能够降低 VM 的性能下降程度。另一方面,恶意用户基于共享的物理资源可以发动侧信道攻击^[6],窃取用户隐私数据。为了防御侧信道攻击,文献[7]分析对比了 3 种在防御侧信道攻击时使用的 VM 分配策略,并提出了一种优先选择服务器的 VM 分配方法,即优先将同一用户的 VM 部署到相同服务器上。文献[8]基于博弈论研究了 VM “何时迁”的问题,不但给出了防御侧信道攻击的策略,还降低了迁移成本。文献[9]将中国墙的思想引入 VM 的部署和迁移中,将用户之间的关系分为联

盟关系和冲突关系,根据这两种关系进行 VM 的部署和迁移。文献[10]对所有物理服务器进行了分组,在部署 VM 时依次进行组选择和组内服务器选择,并分别进行安全优化和资源优化,该方法认为部署在同组服务器上的所有 VM 之间发生侧信道攻击的可能性很小。

但是,网络切片的部署本身就是一个很复杂的问题,在实际中往往会同时考虑多方面的因素。但现在的部署方法未能联合考虑虚拟节点上存在的性能下降和侧信道攻击,为解决该问题,本文提出了一种基于隔离等级的网络切片部署方法。该方法从资源供需比的角度评估 NSI 虚拟节点的性能隔离水平,计算与 NSI 共享资源的其他 NSI 的数量,用来评估安全隔离水平,并基于二者求出对应的隔离等级;在部署时,不但要保证所有的 NSI 均能满足隔离需求,而且仅允许固定隔离等级的 NSI 共享资源,并将 NSI 部署在成本开销最小的位置上;利用整数线性规划方法对该问题进行建模,而且使用基于遗传-离散粒子群算法求出最终的部署结果。仿真结果表明,该方法具有较低的部署成本和较高的收益开销比,并且能够同时保证性能隔离和安全隔离。

2 网络模型及问题描述

2.1 网络模型

网络切片利用软件定义网络(SDN, software defined network)和网络功能虚拟化(NFV, network function virtualization)技术实现按需组网,利用 SDN 技术将网元功能解耦成控制面功能和用户面功能^[11],并在它们之间定义可编程接口;利用 NFV 技术将物理网络根据场景需求虚拟化成多个相互独立的虚拟网络,按照不同的业务场景和话务模型定制裁剪网络功能,管理编排相应的网络资源,为不同的用户提供端到端的定制化服务。

当用户发起服务请求时,虚拟网络提供商首先根据用户的服务需求选择或设计虚拟网络功能,按照一定顺序将这些网络功能连接起来,形成提供专用服务的虚拟网络,如图 1 所示;然后将物理网络的资源分配给 NSI,激活所提供的服务;当用户设备与切片实例的服务关联成功后,

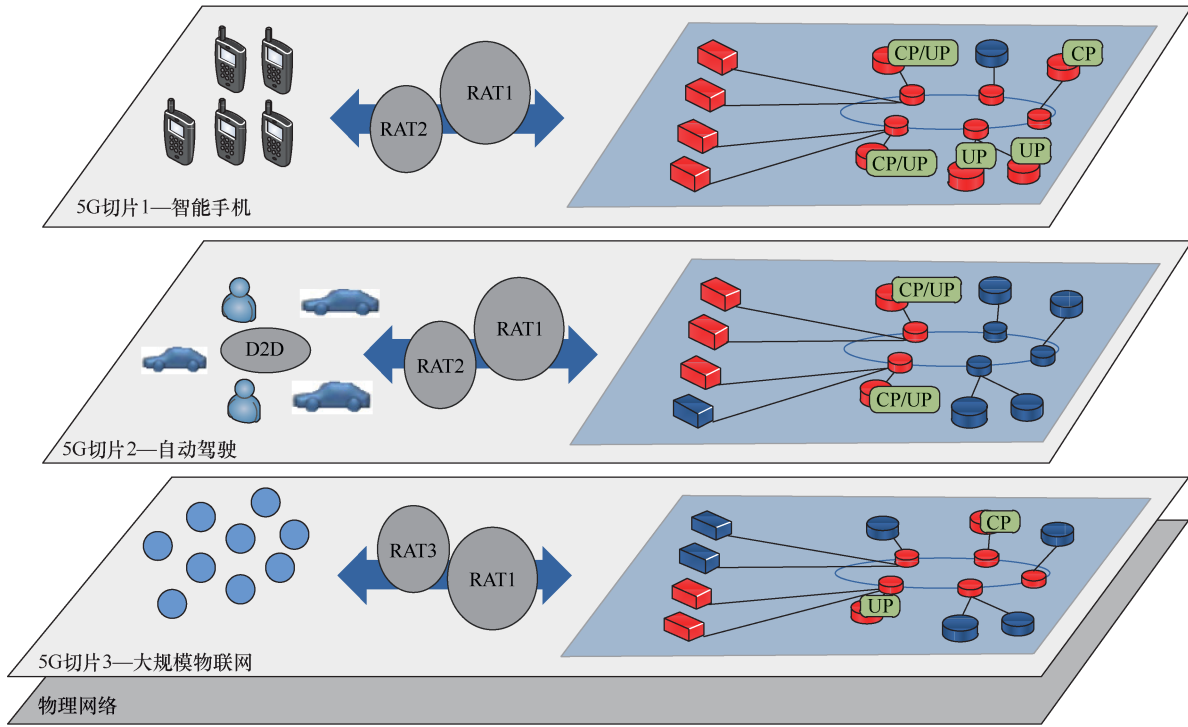


图1 网络切片实例
Figure 1 Network slicing instance

网络流量开始流过切片；系统会持续监控网络切片的相关指标（如资源利用率、端到端延迟和数据吞吐量等），按需重新配置网络切片^[12]。

网络切片实例到物理网络的映射被称为网络切片实例部署，实际是将虚拟节点映射到物理服务器上，将虚拟链路映射到一组物理链路上。随着 NSI 请求不断增多，在一个物理网络上会部署多个 NSI，具体细节如图 2 所示。物理网络 u_s 可以用一个有权无向图 $G_s = (N_s, L_s, C_N, C_L)$ 表示， N_s 和 L_s 分别是物理节点集合和物理链路集合， $n_s \in N_s$ 表示物理服务器； $l_s(C, D) \in L_s$ 是服务器 C 和服务器 D 间的物理链路。服务器可以提供多种物理资源， $C_N = \{c_{cpu}, c_{mey}, c_{net}\}$ 是物理服务器上剩余的物理资源集合， c_{cpu} 、 c_{mey} 和 c_{net} 分别表示计算资源、存储资源和网络资源， $C_L = \{c_{band}\}$ 表示物理链路的剩余带宽资源。假设有 K 个 NSI 等待部署，其中，第 j 个网络切片实例 v_j 可以表示为 $G_v^j = (N_v^j, L_v^j, R_N^j, R_L^j)$ ， N_v^j 和 L_v^j 表示 v_j 的虚拟节点集合和虚拟链路集合。同理， $R_N^j = \{r_{cpu}^j, r_{mey}^j, r_{net}^j\}$ 是虚拟节点所需的计算、存储

和网络资源的集合， $R_L^j = \{r_{band}^j\}$ 表示虚拟链路所需的带宽资源。

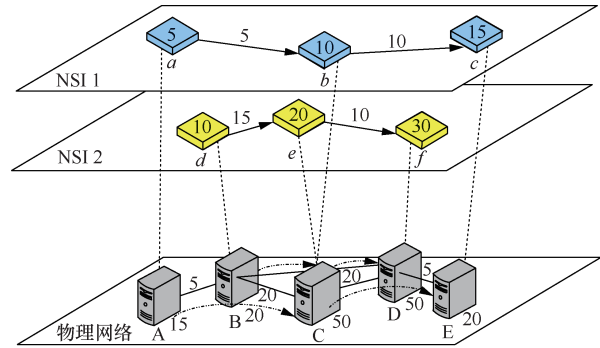


图2 网络切片实例部署细节
Figure 2 Network slicing instance deployment

2.2 隔离问题描述

不同的切片类型、不同租户的网络切片之间需要实施隔离，本文主要研究同一服务器上多个 VM 之间的隔离，当多个 NSI 部署到同一物理网络上时，必然存在多个 VM 部署到同一物理服务器上的情况。如图 3 所示，服务器上承载着 3 种类型的 NSI，分别是普通 NSI、目标 NSI 和恶意

NSI。不同的 NSI 间存在两种类型的隔离问题。

- ① 性能隔离，在共享物理资源的 NSI 间，可能由于误操作或过载而占用大量物理资源，造成性能严重下降，影响用户体验^[13]。在服务器 D 上部署了较多 VM，故 VM 间过度竞争底层硬件资源的可能性很大，由此带来 VM 的性能下降。
- ② 安全隔离，恶意 NSI 为窃取目标 NSI 的隐私信息，首先针对目标 NSI 所属区域和类型，启动大量相同区域和相同类型的 VM 实例；然后进行 VM 同驻检测，查看是否和目标 VM 同驻成功；最后在同驻成功的服务器上实施侧信道攻击。在服务器 C 上，VM b 会故意向 VM e 发送特定服务请求，VM e 在处理该请求时，会对底层物理资源产生一定影响，之后虚拟机监视器（Hypervisor）切换到 VM b 运行，因 Hypervisor 并不会立即清空物理资源中的数据，VM b 经过评估和测量，可以推断出 VM e 的隐私信息，“侧信道”便建立起来^[14]。

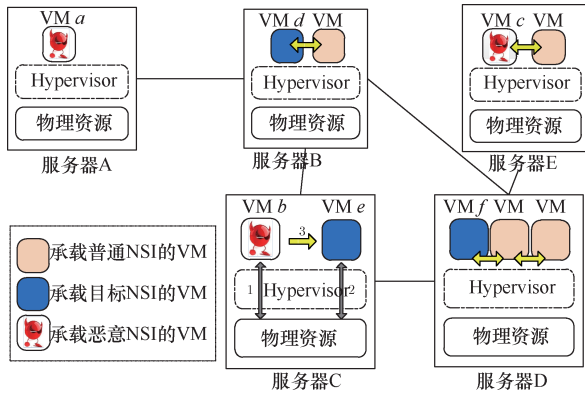


图3 底层物理网络的部署
Figure 3 VM deployment on physical network

3 部署方法描述

3.1 隔离等级的确定

网络切片实例 v_j 的隔离等级 H_v^j 是性能隔离等级 $Pref_v^j$ 和安全隔离等级 Sec_v^j 的加权和，即 $H_v^j = \kappa_1 Pref_v^j + \kappa_2 Sec_v^j$ ，其中 $\kappa_1 + \kappa_2 = 1$ ，可以根据用户需求调整 κ_1 和 κ_2 的取值。将性能隔离等级和安全隔离等级划分为5级，分别是1级（较高）、2级（高）、3级（中）、4级（普通）、5级（较低）。下面分别介绍性能隔离等级和安全隔离等级的求法。

性能隔离等级：在物理资源超额分配的情况下，多个 VM 过度竞争一种或多种物理资源，因物理资源和虚拟资源的供需不匹配，带来 VM 的性能下降。将 VM 需求的虚拟资源与服务器剩余的物理资源的比值定义为性能隔离因子。假设 v_j 共 m 个节点，当第 x 个虚拟节点 $n_v^{j,x}$ 部署在服务器 n_s^x 上时， $n_v^{j,x}$ 的性能隔离因子 $\eta_v^{j,x}$ 如式(1)所示。

$$\eta_v^{j,x} = \varepsilon_1 \frac{r_{cpu}^{j,x}}{c_{cpu}} + \varepsilon_2 \frac{r_{mey}^{j,x}}{c_{mey}} + \varepsilon_3 \frac{r_{net}^{j,x}}{c_{net}} \quad (1)$$

因不同节点所需的不同类型的资源大小存在差异，采用权重系数 ε 进行调整，且 $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 1$ 。 $\eta_v^{j,x} \in (0,1]$ ， $\eta_v^{j,x}$ 越接近 0，其性能水平越高。因 m 个节点的虚拟资源的需求不同，映射至服务器的剩余物理资源也不同，故求出的性能隔离因子不同，将所有节点中 $\eta_v^{j,x}$ 最大的值作为 v_j 整体的性能隔离水平，即 $\eta_v^j = \max\{\eta_v^{j,1}, \dots, \eta_v^{j,m}\}$ 。根据 η_v^j 值将 v_j 的性能隔离等级 $Perf_v^j$ 分为 1~5 级，如表 1 所示。

表1 隔离等级
Table 1 Isolation level

隔离等级	η_v^j 数值范围	σ_v^j 数值范围
1 级（较高）	(0,0.2]	$\left[0, m \cdot \frac{\varphi}{5}\right]$
2 级（高）	(0.2,0.4]	$\left(2m \cdot \frac{\varphi}{5}, 4m \cdot \frac{\varphi}{5}\right]$
3 级（中）	(0.4,0.6]	$\left(4m \cdot \frac{\varphi}{5}, 6m \cdot \frac{\varphi}{5}\right]$
4 级（普通）	(0.6,0.8]	$\left(6m \cdot \frac{\varphi}{5}, 8m \cdot \frac{\varphi}{5}\right]$
5 级（较低）	(0.8,1]	$\left(8m \cdot \frac{\varphi}{5}, m\varphi\right]$

安全隔离等级：在发动侧信道攻击时，攻击者通过分析目标 NSI 部署策略中存在的漏洞，可以确定目标 NSI 的 VM 的部署位置，从而将恶意 VM 部署在相同的服务器上。对于一个 NSI 来说，与其共享物理资源的 NSI 的数量越多，受到侧信道攻击的可能性越大。对于网络切片实例 v_j 的 m 个虚拟节点，将 m 个虚拟节点部署在不同服务器上，假设基础设施运营商允许在服务器上最多部署 φ 个虚拟节点，故与实例 v_j 共存的 NSI 数量 σ_v^j 介于 0 到 $m\varphi$ 之间，根据 σ_v^j 的值将 n_v^j 的安全隔离等级

Sec_v^j 分为 1~5 级, 如表 1 所示。

3.2 部署位置的选择

不同网络切片实例的服务需求不同, 基础设施运营商会根据 NSI 的服务需求选择合适的位置进行部署。设网络切片实例 v_j 的服务需求是 $\Theta_v^j = \{R_v^j, \eta_{\max}^j, \sigma_{\max}^j, H_{\max}^j, \Gamma_j\}$, 其中 $R_v^j = \{r_{\text{cpu}}^j, r_{\text{mey}}^j, r_{\text{net}}^j\}$ 表示虚拟节点所需的多种虚拟资源的集合; $\eta_{\max}^j$ 是 v_j 所有虚拟节点可接受的性能隔离因子的最大值; $\sigma_{\max}^j$ 是 v_j 允许与其他 NSI 共存的最大数量; 由 $\eta_{\max}^j$ 和 $\sigma_{\max}^j$ 可求出对应的性能隔离等级 $Pref_v^j$ 和安全隔离等级 Sec_v^j , 进而求出隔离等级 $H_{\max}^j$; 此外, 为了保证 v_j 的隔离水平, 只允许固定隔离等级差值的 NSI 共享物理资源, 该差值即为 Γ_j 。

在选择部署位置时, 主要考虑以下几个方面。① 尽量降低部署成本, 主要考虑租赁物理节点和物理链路所带来的成本支出。② 性能和安全的隔离水平不是一成不变的, 是随着 NSI 的不断部署而不断变化的。所以在部署新请求时, 不但要保证“新”到来的 NSI 满足隔离需求, 还要保证已经部署的“老”NSI 仍满足隔离需求, 要保证“新”“老”NSI 的性能隔离因子值均不超过各自规定的 $\eta_{\max}$, 与其他 NSI 共存的数量不超过各自规定的 $\sigma_{\max}$ ($\eta_{\max}$ 和 $\sigma_{\max}$ 都是值越小, 对应的隔离水平越高, 隔离等级越低)。③ 为了防止隔离等级差距很大的 NSI 共存, 要求部署在相同服务器上的多个 NSI 的隔离等级的差值不超过 Γ_j 。

4 部署模型建立

4.1 变量设定

现将第 j 个网络切片实例 v_j 部署在物理网络 u_s 上, v_j 共有 m 个虚拟节点, u_s 共有 n 个物理节点, 定义两个变量。① $\alpha_{j,x,y}$: 当 v_j 的第 x 个虚拟节点 $n_v^{j,x}$ 成功映射到 u_s 的第 y 个服务器 n_s^y 上时, $\alpha_{j,x,y}$ 值为 1, 否则值为 0。② $\beta_{j,ef,\zeta}$: 当 v_j 的虚拟链路 $l_v^{j,ef}$ 成功映射到 u_s 的物理链路 ζ 上时, $\beta_{j,ef,\zeta}$ 值为 1, 否则值为 0。

4.2 目标函数

部署 NSI 时, 主要存在节点资源和链路资源两方面的费用。假设部署 v_j 单位 CPU 资源、单位内存资源和单位网络资源所需要的费用分别是 ω_{cpu}^j 、 ω_{mey}^j 和 ω_{net}^j , 则部署虚拟节点 $n_v^{j,x}$ 所耗费费用是 $\Omega_v^{j,x} = \omega_{\text{cpu}}^j r_{\text{cpu}}^{j,x} + \omega_{\text{mey}}^j r_{\text{mey}}^{j,x} + \omega_{\text{net}}^j r_{\text{net}}^{j,x}$ 。在部署 v_j 的虚拟链路时, 所耗费的费用与实际部署到物理链路的跳数(条数)及带宽资源的需求量有关。对于虚拟节点 e 和 f 间的虚拟链路 l_{ef}^j , 带宽资源需求量是 $r_{\text{band}}^{j,ef}$, 部署时经过的物理链路的跳数是 hop_{ef}^j , 部署单位带宽资源需要的费用是 ω_{band}^j , 部署虚拟链路 l_{ef}^j 的总费用是 $J_{ef}^j = \omega_{\text{band}}^j r_{\text{band}}^{j,ef} hop_{ef}^j$ 。部署网络切片实例 v_j 的总成本为 $\text{Total}_V^j = \sum_{x=1}^m \Omega_v^{j,x} + \sum_{l_{ef}^j \in L_v^j} J_{ef}^j$ 。为节省部署成本, 将部署成本最小化作为目标函数, 如式(2)所示。

$$\min \text{Total}_V^j = \min \left\{ \sum_{x=1}^m \alpha_{j,x,y} \Omega_v^{j,x} + \sum_{l_{ef}^j \in L_v^j} \beta_{j,ef,\zeta} J_{ef}^j \right\} \quad (2)$$

4.3 约束条件

当 v_j 的第 x 个虚拟节点 $n_v^{j,x}$ 映射到 u_s 的第 y 个服务器 n_s^y 上时, n_s^y 的剩余 CPU 资源、剩余存储资源及剩余网络资源不小于 $n_v^{j,x}$ 所需的对应的虚拟资源, 如式(3)所示。

$$\begin{cases} \sum_{x=1}^m \alpha_{j,x,y} (c_{\text{cpu}}^y - r_{\text{cpu}}^{j,x}) \geq 0 \\ \sum_{x=1}^m \alpha_{j,x,y} (c_{\text{mey}}^y - r_{\text{mey}}^{j,x}) \geq 0, \forall n_s^y \in N_s \\ \sum_{x=1}^m \alpha_{j,x,y} (c_{\text{net}}^y - r_{\text{net}}^{j,x}) \geq 0 \end{cases} \quad (3)$$

式(4)表示一个虚拟节点只能映射到一个服务器上, 不能对其进行分割; 式(5)表示请求部署的 NSI 的所有虚拟节点全都部署到物理网络上, 式(4)和式(5)即唯一性约束。

$$\sum_{x=1}^m \alpha_{j,x,y} \leq 1, \forall n_s^y \in N_s \quad (4)$$

$$\sum_{y=1}^n \alpha_{j,x,y} = 1, \forall n_v^{j,x} \in N_v^j \quad (5)$$

当 $n_v^{j,e} \mapsto n_s^a$, $n_v^{j,f} \mapsto n_s^b$ 时, 服务器 n_s^a 经过 hop 跳到达服务器 n_s^b , 即 $\zeta = \{l_s(n_s^a, n_s^1), \dots, l_s(n_s^{\text{hop}-1}, n_s^b)\}$, ζ 所有单条物理链路中的带宽资源最小值不小于虚拟链路 $l_v(n_v^e, n_v^f)$ 所需的带宽资源, 如式(6)所示。

$$\sum_{l_v^{j,ef} \in L_v^j} \beta_{j,ef,\zeta} (\min\{c_{\text{band}}(n_s^a, n_s^1), \dots, c_{\text{band}}(n_s^{\text{hop}-1}, n_s^b)\} - r_{\text{band}}^j(n_v^{j,e}, n_v^{j,f})) \geq 0, \forall \zeta \in L_s \quad (6)$$

假设 $n_v^{j,x}$ 映射到服务器 n_s^y 上时, 还存在 w 个已经部署的 NSI 的虚拟节点, 任意节点表示为 $n_v^{\theta,z}$, θ 是该节点隶属的 NSI, $n_v^{j,x}$ 和 $n_v^{\theta,z}$ 应同时满足式(7)~式(9)的约束条件。式(7)说明部署时“新”“老”节点的性能隔离因子不大于各自可接受的最大性能隔离因子的值, 式(8)说明与“新”“老”NSI 共存的其他 NSI 的数量都不超过各自可接受数量的最大值。除此之外, 为了防止隔离等级相差较大的 NSI 共存, 还要保证“新”“老”NSI 与其他 NSI 的隔离等级差值不超过各自的规定值。

$$\begin{cases} \eta_v^{j,x} \leq \eta_{\max}^j, \forall n_v^{j,x} \in N_V^j \\ \eta_v^{\theta,z} \leq \eta_{\max}^\theta, \forall n_v^{\theta,z} \in N_V^\theta \end{cases} \quad (7)$$

$$\begin{cases} \sigma_v^j \leq \sigma_{\max}^j \\ \sigma_v^\theta \leq \sigma_{\max}^\theta \end{cases} \quad (8)$$

$$\begin{cases} |H_v^j - H_{\max}^j| \leq \Gamma_j \\ |H_v^\theta - H_{\max}^\theta| \leq \Gamma_\theta \end{cases} \quad (9)$$

5 算法描述

网络切片的映射问题是一个 NP 难问题, 在多项式时间内无法利用传统方法进行求解, 为解决该问题, 采用基于遗传算法 (GA, genetic algorithm) 优化的离散粒子群 (DPSO, discrete particle swarm optimization) 算法^[15]进行求解。在粒子群算法的每个种群更新速度和位置后, 进行交叉和变异操作, 改进后的算法有效克服了粒子群算法在求极值时存在的早熟问题, 增强了算法全局寻优的能力, 提高了求解的精度。

为了减少基础设施运营商的部署成本, 把部署成本作为适应度函数, 如式(10)所示。为了求出映射结果, 首先随机初始化一个含有多个粒子组成的种群, 其规模为 $Scope$, 每一个粒子都是 v_j 可能的部署位置, 向量 $\vec{x}_\gamma = (x_{\gamma 1}, x_{\gamma 2}, \dots, x_{\gamma A})$ 表示第 γ 个粒子, 其中, γ 是 $[1, Scope]$ 上的整数, A 是 NSI 请求的节点总个数, $x_{\gamma a}$ 表示第 γ 个粒子中把第 a 个虚拟节点映射到服务器的序号; 将 $\vec{x}_\gamma$ 代入适应度函数中, 求出第 γ 个粒子的部署成本。然后更新当前的个体最优解 $\vec{p}_{\gamma a}$ 和全局最优解 $\vec{p}_{ga}$ 。为了使粒子快速而准确地飞向最优解, 每个粒子根据飞行速度更新当前的部署位置, 第 γ 个粒子的飞行速度可以用向量 $\vec{v}_\gamma = (v_{\gamma 1}, v_{\gamma 2}, \dots, v_{\gamma A})$ 表示。速度更新公式如式(11)所示, 其中 $a \in [1, A]$, 学习因子 c_1 和 c_2 分别代表粒子本身获得的经验以及其他粒子获得的经验, 调整 c_1 和 c_2 的值可以控制粒子在全局探测和局部开采的有效平衡, r_1 和 r_2 是相互独立的伪随机数, 服从 $[0, 1]$ 上的均匀分布, 使用式(12)更新位置。

$$\text{Fitness}(v_j, \vec{x}_\gamma) = \sum_{x=1}^m \alpha_{j,x,y} \Omega_v^{j,x} + \sum_{l_v^{j,ef} \in L_v^j} \beta_{j,ef,\zeta} J_{ef}^j \quad (10)$$

$$\begin{aligned} v_{\gamma a}(t+1) &= v_{\gamma a}(t) + c_1 r_{1a}(p_{\gamma a}(t) - x_{\gamma a}(t)) + \\ &c_2 r_{2a}(p_{ga}(t) - x_{\gamma a}(t)) \end{aligned} \quad (11)$$

$$x_{\gamma a}(t+1) = x_{\gamma a}(t) + v_{\gamma a}(t+1) \quad (12)$$

在更新粒子位置后, 算法不会立即更新个体最优解和全局最优解, 而是利用遗传算法对当前的粒子位置进行再更新。其中, 每一个粒子都是一条染色体, 所部署的服务器序号即染色体的基因。首先对当前粒子的位置信息进行交叉操作, 交叉概率是 pc , 随机选择两个粒子, 随机选择两个虚拟节点所映射的服务器序号作为交叉的位置, 交叉结束后, 再验证两个染色体的可行性, 即是否满足唯一性约束条件、是否超过服务器序号范围。然后进行变异操作, 随机选择一个个体, 并以 pm 的概率改变某个基因的值, 之后验证染色体的可行性, 如果满足, 更新个体最优解 $\vec{p}_{\gamma a}$ 和全局最优解 $\vec{p}_{ga}$ 。该算法流程如图 4 所示, 具体的流程如算法 1 和算法 2 所示。

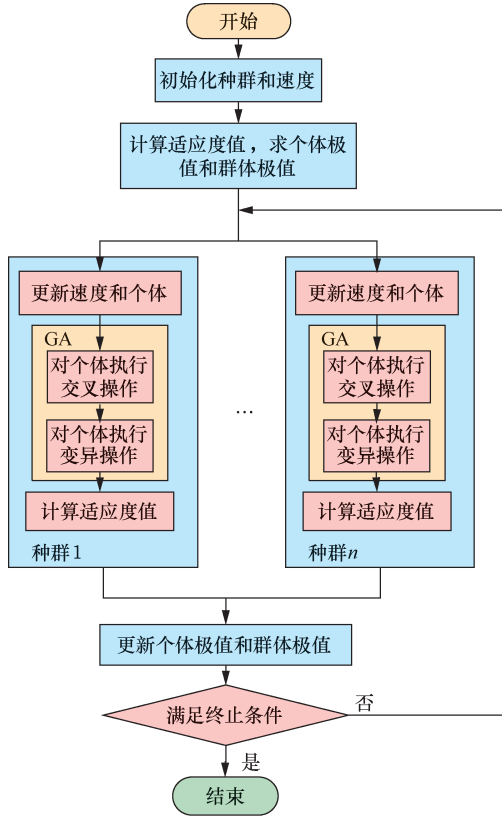


图4 GA-PSO 算法流程
Figure 4 GA-PSO algorithm flow

算法1 GA-PSO(G_S, G_V, Θ_V)

输入 G_S ; 网络切片实例请求 G_V^j 及其服务需求 Θ_V^j

输出 部署结果 $\Phi_j = \{M_N^j, M_L^j\}$

- 1) 设置种群规模 $Scope$ 、最大迭代次数 $MAXg$ 及 c_1 和 c_2 ;
- 2) 为所有粒子初始化位置向量 $\vec{x}$ 、速度向量 $\vec{v}$;
- 3) 用算法2求初始种群所有粒子适应度值, 并求出个体最优和全局最优值;
- 4) for $1 \leq i \leq MAXg$ do
- 5) for $1 \leq \gamma \leq Scope$ do
- 6) 利用式(11)及式(12)更新速度向量 $\vec{v}_\gamma$ 、位置向量 $\vec{x}_\gamma$;
- 7) 对位置向量 $\vec{x}_\gamma$ 依概率 pc 进行交叉操作;
- 8) 对位置向量 $\vec{x}_\gamma$ 依概率 pm 进行变异操作;
- 9) 使用算法2计算适应度值, 更新个体

最优值、全局最优值;

10) end for

11) end for

12) 依据全局最优值, 输出 G_V^j 的部署结果

$$\Phi_j = \{M_N^j, M_L^j\}$$

算法2 适应度求解函数 $FIT(G_S, G_V^j, \Theta_V^j, \vec{x}_\gamma)$

输入 G_S ; G_V^j ; Θ_V^j ; $\vec{x}_\gamma$

输出 $Fitness(v_j, \vec{x}_\gamma)$

- 1) 设置 ω_{cpu}^j 、 ω_{mey}^j 、 ω_{net}^j 及 ω_{band}^j ;
- 2) if $\vec{x}_\gamma$ 满足式(3) && 满足式(4) && 满足式(5) && 满足式(7) && 满足式(8) && 满足式(9)
- 3) for $1 \leq k \leq length(\vec{x}_\gamma) - 1$ do
- 4) 获取虚拟节点 $n_v^{j,k}$ 和 $n_v^{j,k+1}$ 的映射位置, 分别为 n_s^a 和 n_s^b ;
- 5) 最短路径算法求得 n_s^a 到 n_s^b 的最短路径的跳数为 hop ;
- 6) while n_s^a 到 n_s^b 间存在有效路径 do
- 7) 用 K 短路径算法求出 n_s^a 到 n_s^b 间跳数为 hop 的路径;
- 8) if $\min\{c_{band}(n_s^a, n_s^1), \dots, c_{band}(n_s^{hop-1}, n_s^b)\} - r_{band}^j(n_v^{j,k}, n_v^{j,k+1}) \geq 0$ do
- 9) 将 $l(n_v^{j,k}, n_v^{j,k+1})$ 映射到 $\zeta_{a \rightarrow b}$ 上, 用式(10)求 $Fitness(v_j, \vec{x}_\gamma)$;
- 10) else $hop = hop + 1$, go to (7); end if
- 11) end while
- 12) if n_s^a 到 n_s^b 间不存在有效路径 do
- 13) $Fitness(v_j, \vec{x}_\gamma) = U$, U 是一个足够大的正数; break;
- 14) end if
- 15) end for
- 16) else $Fitness(v_j, \vec{x}_\gamma) = U$ end if
- 17) 返回 $Fitness(v_j, \vec{x}_\gamma)$ 的值

6 仿真分析

6.1 实验环境

在配置为 Intel(R) Core(TM) i5-2450 2.5 GHz CPU, 4 GB 内存的 PC 机上仿真, 用 GT-ITM 生成物理网络和 NSI, 并使用 Matlab 对结果进行分

析。网络参数设置如表2所示, 仿真参数设置如表3所示。切片实例请求的最大性能隔离因子 $\eta_{\max}$ 服从 $[0.5, 1]$ 上的均匀分布, 当该切片实例请求的节点数量是 m 时, 与其他切片实例共存的最大数量 $\sigma_{\max}$ 服从 $[4, 4m]$ 上的均匀分布, 根据 $\eta_{\max}$ 和 $\sigma_{\max}$ 可以求出对应的隔离等级 $H_{\max}$, 设定 Γ 值为1, 即仅允许相邻隔离等级的切片实例共享资源, $\omega_{\text{cpu}}=0.2$, $\omega_{\text{mey}}=\omega_{\text{net}}=0.15$, $\omega_{\text{band}}=0.5$ 。

表2 网络参数
Table 2 Network parameters

参数	物理网络	网络切片实例
网络模型	纯随机	纯随机
节点连接概率 p	0.2	0.5
节点个数	50	服从 $U[3, 5]$
节点和链路资源容量	服从 $U[50, 100]$	服从 $U[5, 20]$
请求到达	无	服从每100个时间单元到达5个泊松分布
请求生命周期	无	服从均值为2000个时间单元的指数分布

表3 仿真参数
Table 3 Simulation parameters

网络参数	取值
最大迭代次数	60
种群规模	40
c_1, c_2	1.494 45
λ	0.6
pc	0.7
pm	0.3

6.2 结果分析

将本文提出的方法(方法1)与文献[16](方法2)及最小化共存度的方法(方法3)进行比较。方法2是为了应对侧信道攻击提出的, 该方法将VM分成高中低3个安全等级, 在映射时仅允许同安全等级的虚拟节点共享资源。方法3是为了应对性能下降提出的, 该方法把最小化与其他网络切片实例共存的数量作为目标函数。从部署成本、请求接受率、收益开销比、性能等级未达标NSI数量的占比、安全等级未达标NSI数量的占比和隔离等级未达标NSI数量的占比这几方面对比这3种方法。

图5显示了不同方法的部署成本总和的变化情况。可以看出, 随着NSI请求数量增多, 这3种方法的部署成本总和均呈现增长的趋势, 但本文方法的部署成本总和增长速率低于方法2和方法3, 方法3部署成本增长速率最高。图6反映了不同方法的收益开销比的变化情况。随着NSI请求数量增多, 这3种方法的收益开销比最终稳定在不同的区间上, 本文方法收益开销比高于其他两种方法, 方法3收益开销比最低。相较于对比的其他方法, 本文方法在部署成本和收益开销比两方面, 均表现出了较好的效果, 这是因为本文方法在部署时不但保证服务器上虚拟节点的共存数量与虚拟资源和物理资源的供需比都满足节点部署要求, 而且保证“新”“老”虚拟节点的部署需求, 使底层物理网络的碎片资源数量相对较少, 降低了部署成本且提高了收益开销比。

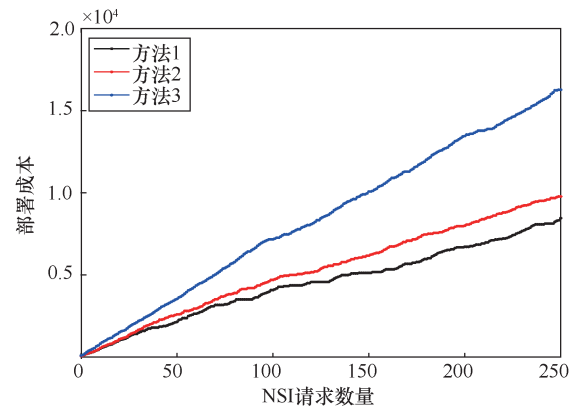


图5 部署成本
Figure 5 Deployment cost

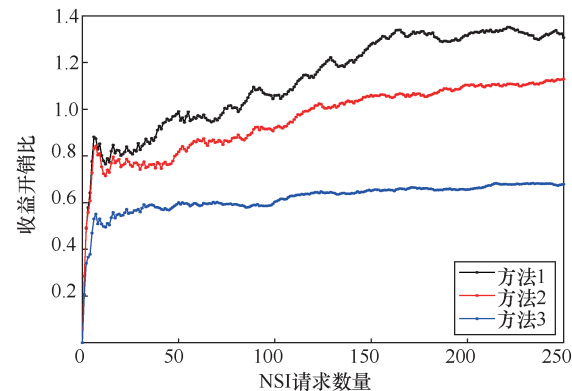


图6 收益开销比
Figure 6 Revenue-to-cost ratio

图7显示了不同方法的请求接受率的变化情况。随着 NSI 请求数量增多,这3种方法的请求接受率都存在不同程度的下降,而且随着 NSI 请求数量的增多都趋于稳定。在刚开始部署时,物理网上承载的 NSI 数量很少,所以3种方法在开始时请求接受率都比较高,随着物理网络资源被占用增多,请求接受率产生较为明显的下降。但本文方法请求接受率最低,主要是因为本文方法在部署时需要兼顾安全和性能两方面的约束条件,使部署成功的概率变得更小,即用较低请求接受率换取了较高的隔离水平。

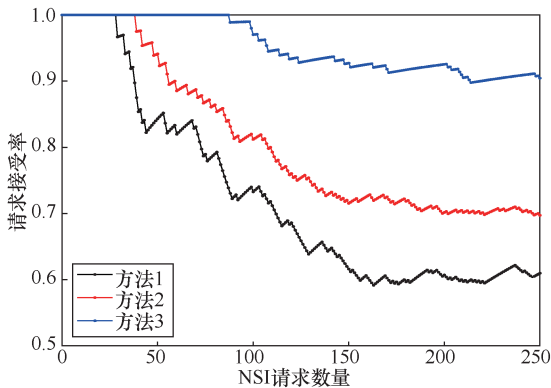


图7 请求接受率
Figure 7 Request acceptance rate

为了进一步证明本文方法的有效性,从性能等级、安全等级和隔离等级3方面对这3种方法进行比较。假设当前共部署了 N 个NSI请求,对于仍部署在物理网络上的每一个NSI,其中超过其可接受的最低性能隔离因子的NSI数量是 N_η ,超过其可接受的最低基本安全水平的NSI数量是 N_σ ,超过其可接受的最低隔离等级的NSI数量是 N_{ISO} ,图8~图10分别表示 $\frac{N_\eta}{N}$ 、 $\frac{N_\sigma}{N}$ 和 $\frac{N_{ISO}}{N}$ 随NSI请求数量的变化情况。可以看出,本文方法在这3方面均处于很低的水平,说明使用本文方法部署NSI,可以同时保证性能等级、安全等级和隔离等级3方面的要求。其他两种方法在性能或是安全等级未达标的NSI数量上明显高于本文方法,因此,仿真结果表明本文方法在一定程度上可以同时保证用户在安全和性能两方面的隔离需求。

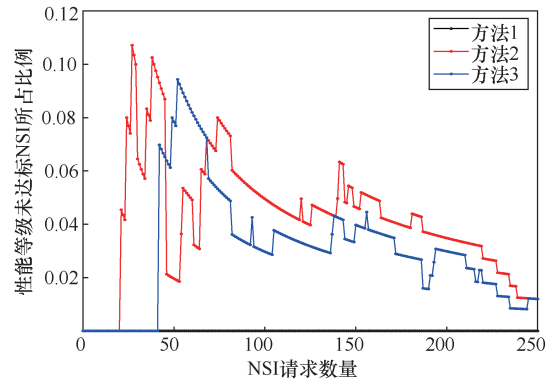


图8 性能等级未达标 NSI 所占比例
Figure 8 Proportion of NSI whose performance level is not up to standard

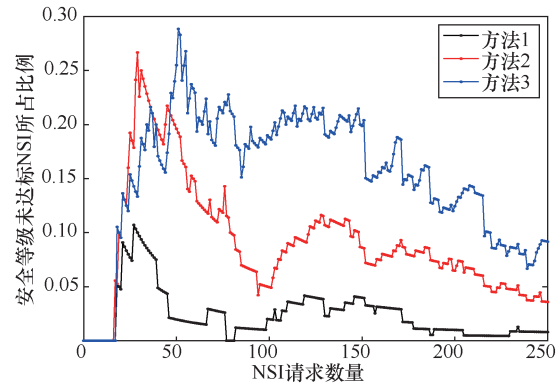


图9 安全等级未达标 NSI 所占比例
Figure 9 Proportion of NSI whose security level is not up to standard

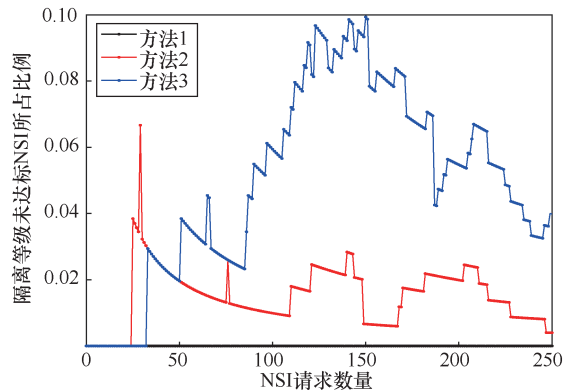


图10 隔离等级未达标 NSI 所占比例
Figure 10 Proportion of NSI whose isolation level is not up to standard

7 结束语

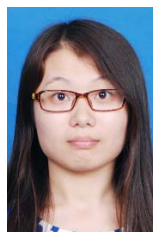
网络切片是基于虚拟化技术实现的,虚拟化技术的引入使多个NSI可以部署到同一物理基础设施上,虽然提高了系统的灵活性和效率并降低了成本,但其共享底层资源的特点也带来了性能下降和侧信道攻击的问题,故本文提出了一种基于隔离等级的网络切片部署方法。在部署时,

将部署成本最小作为目标函数, 最终的部署位置一方面保证“新”“老”NSI 均能达到各自的隔离要求, 另一方面按需求将处于不同隔离等级的 NSI 划分到不同分组, 避免了隔离等级差距很大的 NSI 共存。采用改进型的离散粒子群算法进行迭代寻优, 求出最终的部署结果。最终仿真结果证明本文方法可以同时保证用户在安全和性能两方面的隔离需求, 但网络请求接受率降低, 在未来的研究中将对该方法做进一步的改进。

参考文献:

- [1] FOUKAS X, PATOUNA S G, ELMOKASHFIA, et al. Network slicing in 5G: survey and challenges[J]. IEEE Communications Magazine, 2017, 55(5): 94-100.
- [2] ZHU Q, ZHU J D, AGRAWAL G. Power-aware consolidation of scientific workflows in virtualized environments[C]//High Performance Computing, Networking, Storage & Analysis. 2010: 1-12.
- [3] XU F, LIU F M, LIU L H, et al. iAware: making live migration of virtual machines interference-aware in the cloud[J]. IEEE Transactions on Computers, 2014, 63(12): 3012-3025.
- [4] ALAN R, AMANK, SRIRAM G, et al. PACMan: performance aware virtual machine consolidation[C]//The 10th International Conference on Autonomic Computing. 2013: 83-94.
- [5] ROOZBEHM, ALBERTY Z. Managing performance degradation of collocated virtual machines in private cloud[C]//The 18th International Conference on High Performance Computing and Communications. 2016: 128-135.
- [6] GULMEZOGLUB, MEHMET S İ, IRAZOQUIG, et al. Cross-VM cache attacks on AES[J]. IEEE Transactions on Multi-Scale Computing Systems, 2017, 2(3): 211-222.
- [7] 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.
- [8] AHMED H A, GEORGE A, MINA G, et al. It's time to migrate! A game-theoretic framework for protecting a multi-tenant cloud against collocation attacks[C]//IEEE International Conference on Cloud Computing. 2018: 725-731.
- [9] YU S, GUI X, LIN J, et al. A security-awareness virtual machine management scheme based on Chinese wall policy in cloud computing[J]. The Scientific World Journal, 2014(5): 805-923.
- [10] LIANG X, GUI X L, JIANA N, et al. Mitigating cloud co-resident attacks via grouping-based virtual machine placement strategy[C]//IEEE International Performance Computing and Communications Conference. 2017: 1-8.
- [11] SAMAM R, AN X L, WEI Q, et al. Reshaping the mobile core network via function decomposition and network slicing for the 5G era[C]//IEEE Wireless Communications and Networking Conference Workshops. 2016: 90-96.
- [12] SAMEER KUMARS, RAYMOND M, ANDREA F. A cloud-native approach to 5G network slicing[J]. IEEE Journals & Magazines, 2017, 55(8): 120-127.
- [13] TOMAS L, LAKEWE B, ELMROTHE. Service level and performance aware dynamic resource allocation in overbooked data centers[C]//IEEE/ACM International Symposium on Cluster. 2016.
- [14] 梁鑫, 桂小林, 戴慧珺, 等. 云环境中跨虚拟机的 Cache 侧信道攻击技术研究[J]. 计算机学报, 2017, 40(2): 317-336.
- LIANG X, GUI X L, DAI H J, et al. Research on Cache side channel attack technology of virtual machines in cloud environment [J]. Chinese Journal of Computers, 2017, 40(2): 317-336.
- [15] 潘勇, 郭晓东. 一种基于遗传算法改进的粒子群优化算法[J]. 计算机应用与软件, 2011, 28(9): 222-224.
- PAN Y, GUO X D. An improved particle swarm optimization algorithm based on genetic algorithm[J]. Computer Applications and Software, 2011, 28(9): 222-224.
- [16] 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.
- ZHAO S, JI X S, MAO Y X, et al. Research on dynamic migration of virtual machine based on security level[J]. Journal on Communications, 2017, 38(7): 165-174.

[作者简介]



潘启润(1993—), 女, 河北石家庄人, 战略支援部队信息工程大学硕士生, 主要研究方向为新一代移动通信技术、网络切片。

黄开枝(1973—), 女, 安徽滁州人, 战略支援部队信息工程大学教授、博士生导师, 主要研究方向为移动通信、无线物理层安全。

游伟(1984—), 男, 江西丰城人, 战略支援部队信息工程大学讲师, 主要研究方向为密码学、5G 网络安全。