

# 基于强化学习的服务链映射算法

魏亮, 黄韬, 张娇, 王泽南, 刘江, 刘韵洁

(北京邮电大学网络与交换技术国家重点实验室, 北京 100876)

**摘要:** 提出基于人工智能技术的多智能体服务链资源调度架构, 设计一种基于强化学习的服务链映射算法。通过 Q-learning 的机制, 根据系统状态、执行部署动作后的奖惩反馈来决定服务链中各虚拟网元的部署位置。实验结果表明, 与经典算法相比, 该算法有效降低了业务的平均传输延时, 提升了系统的负载均衡情况。

**关键词:** 网络功能虚拟化; 人工智能; 服务链; 强化学习

**中图分类号:** TP302

**文献标识码:** A

**doi:** 10.11959/j.issn.1000-436x.2018002

## Service chain mapping algorithm based on reinforcement learning

WEI Liang, HUANG Tao, ZHANG Jiao, WANG Zenan, LIU Jiang, LIU Yunjie

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

**Abstract:** A service chain resource scheduling architecture of multi-agent based on artificial intelligence technology was proposed. Meanwhile, a service chain mapping algorithm based on reinforcement learning was designed. Through the Q-learning mechanism, the location of each virtual network element in the service chain was determined according to the system status and the reward and punishment feedback after the deployment. The experimental results show that compared with the classical algorithms, the algorithm effectively reduces the average transmission delay of the service and improves the load balance of the system.

**Key words:** network function virtualization, artificial intelligence, service chain, reinforcement learning

### 1 引言

网络功能虚拟化<sup>[1]</sup> (NFV, network function virtualization) 是未来网络领域的重要研究方向。NFV 技术将网络功能软件整合部署到行业标准的服务器、交换机和存储硬件上, 取代传统专用物理设备, 从而达到降低成本、能耗和复杂性的目的。NFV 同时具备软件与硬件解耦、灵活的网络功能部署、动态的服务提供等能力。以运营商为主导的 NFV 组织就是希望将 NFV 引入到运营商网络中,

使传统的运营商网络逐渐演进为面向业务的新型网络, 进而能够快速将新业务带给网络用户。因此, NFV 具有硬件通用化、业务软件化、资源弹性伸缩等特征, 可以实现网络低成本建设、高效率运营和灵活业务提供。NFV MANO (NFV management and orchestration)<sup>[2]</sup> 在 ETSI ISG NFV 架构中被定义为由多个功能实体组合而成, 主要负责管理和编排基础设施、资源以及服务等。通过组合不同虚拟网络功能来实现网络服务的自动化部署、生命周期管理和端到端的管理。

收稿日期: 2017-09-07; 修回日期: 2017-11-29

基金项目: 国家高技术研究发展计划 (“863” 计划) 基金资助项目 (No.2015AA016101); 国家自然科学基金资助项目 (No.61501042); 北京科技新星基金资助项目 (No.Z151100000315078)

**Foundation Items:** The National High Technology Research and Development Program of China (863 Program) (No.2015AA016101), The National Natural Science Foundation of China (No.61501042), Beijing New-Star Plan of Science and Technology (No.Z151100000315078)

在 NFV 环境中，虚拟网络功能（VNF, virtual network function）运行在一个或多个虚拟机（也可以是容器）上，负责执行处理特定的网络功能<sup>[3]</sup>。单个虚拟网络功能可以连接或组合在一起构建全面的网络通信服务。原先实体化网络组件可以被相应的虚拟网络功能组件所替代，由 NFVI 来承载提供所需的虚拟化资源<sup>[4]</sup>，由 MANO 进行统一地编排管理。常见的 VNF 包括虚拟的交换机（vSwitch）、虚拟的路由器（vRouter）、虚拟的防火墙（vFW）、虚拟的负载均衡（vLB）等网元。

网络功能虚拟化可以和软件定义网络<sup>[5]</sup>（SDN, software-defined networking）技术有机结合。充分利用 SDN 设计理念将网络的数据平面和控制平面进行分离，并实现可编程控制<sup>[6]</sup>，形成应用、控制、转发 3 层架构。SDN 通过集中控制实现网络资源的灵活调度、按需调控，而 NFV 可通过虚拟化技术实现网络功能的灵活部署与动态调整。NFV 和 SDN 两者可以相互补充，共同构建智能化的网络系统。

服务链（service chain）借助软件定义网络技术自动创建一个连接网络服务的虚拟服务链<sup>[7]</sup>，网络流量可以按照业务逻辑要求的既定顺序穿过 VNF。服务链基于软件的方式来创建虚拟链路连接，因此，这些连接可以通过 NFV 编排层来创建和释放。NFV 编排层不但能够将多个 VNF 与物理网络组件组合成服务链，同时还提供 VNF 放置和链接以及动态扩展和弹性服务。为了实现各种业务逻辑，服务链需要面向业务流来定义所需要的网络功能以及处理方式。服务链技术可以广泛应用于数据中心、运营商网络、企业网络等。

服务链定义了特定顺序的网络功能集合，其放置将影响业务编排、服务提供以及物理资源的使用效率。服务链需要面临资源分配优化、动态流量调控、动态服务映射、服务策略实施、服务可靠性及可用性、安全等方面的挑战。它常常以最小化网络延时、最小化能耗、最小化代价、最大化资源利用率等为目标进行建模优化服务链映射问题<sup>[8]</sup>。

## 2 相关研究

服务链的映射放置能够灵活利用基础设施的资源，动态编排网络功能服务。因此，此方向领域是网络功能虚拟化中的研究热点。Kuo 等<sup>[9]</sup>研究 VNF 放置和路径选择联合优化问题，提出了一种根据链路长度和服务器的使用情况来进行服务链部署的算

法。Mechtri 等<sup>[10]</sup>提出一种基于基础设施拓扑图和请求的 VNF 转发图的邻接矩阵特征分解的算法来实现分布式云环境下寻找服务链的最佳放置方案。Wang 等<sup>[11]</sup>在考虑网络代价和服务性能的基础上提出了基于一跳流量调度和基于贪婪算法搜索多条路径的两阶段服务链映射算法。Ye 等<sup>[12]</sup>提出了基于关键子拓扑映射反馈的闭环算法来联合优化 VNF 组合和服务链映射，从而达到最小化整体带宽消耗的目的。Reddy 等<sup>[13]</sup>提出了基于混合整数线性规划的可变邻域搜索算法来实现流量变化和延时受限情况下的服务链映射。Mehraghdam 等<sup>[14]</sup>在考虑最小化延时等目标的基础上将服务链映射问题建模成混合整数二次约束规划问题求解。Zhang 等<sup>[15]</sup>提出根据网络和请求来联合优化调度服务链放置问题，实现最大化平均资源利用率和最小化平均响应时间。Hirwe 等<sup>[16]</sup>通过有向无环图来表征服务链，对拓扑排序以最短路径放置服务链，达到最小化服务链跳数、减少 SDN 交换机上流规则条目、消除 ping-pong 流量的目的。Zhang 等<sup>[17]</sup>针对流量增长情况下服务链的按需扩展问题提出了两步选择的启发式算法。Sahhaf 等<sup>[18]</sup>提出了基于服务分解的服务链映射算法，指出基于资源感知的分解策略选择将显著提升算法性能。Moens 等<sup>[19]</sup>研究了基于物理硬件和虚拟机混合场景下部署服务链的问题，借鉴软件生产线工程理念提出了定制化的服务链放置算法。Fei 等<sup>[20]</sup>针对分布式网络功能虚拟化场景下端局选择和 VNF 分配问题给出优于启发式的算法。Dietrich 等<sup>[21]</sup>提出了基于线性规划和图分割的方法来解决多网络服务提供商的服务链映射问题。

Habib 等<sup>[22]</sup>提出了使用强化学习算法来解决云间资源分配和虚拟机管理问题，Groléat 等<sup>[23]</sup>提出了基于分布式强化学习的方式来解决端到端服务质量保证的问题，Lin 等<sup>[24]</sup>提出了在多级层次化 SDN 中，使用强化学习的手段来构建 QoS 感知的自适应路由机制。

综上所述，尽管目前在服务链放置等问题上已经取得了很多研究成果，但是在动态场景下资源智能调度研究方面依然欠缺。因此，在利用人工智能领域强化学习方法来提升服务链映射效果方面还存在进一步深入研究的必要性。

## 3 问题描述

一般服务链映射问题可以分解成 2 个子问题，

分别是 VNF 放置和 VNF 组链。VNF 放置主要是考虑如何选择合适的物理节点部署 VNF 虚拟机，而 VNF 组链是考虑如何利用 SDN 技术进行流量调度构建服务链。

将用于部署服务链的基础设施建模成为一张有向图，用  $G = (V, E)$  来表示。这里，服务器节点集合用  $V$  表示，其中一台服务器用  $v$  表示。同样地，链路集合用  $E$  表示，其中一条链路用  $e$  表示。对于每一台服务器  $v_i \in V$  而言，其可以虚拟出的 vCPU 的数量为  $C_{v_i}$ ，假设一个 vCPU 能承载一个 VNF，所以每台服务器上能承载  $C_{v_i}$  个 VNF。假设 2 台服务器之间的流量采用最短路径算法调度，将 2 台服务器之间最短路径的延时用  $D(v_i, v_j)$  表示。

系统中支持的 VNF 种类集合用  $F$  表示， $F = (f_1, f_2, f_3, \dots)$ ，其中， $f_i$  为不同类型的 VNF，如防火墙、负载均衡器等。网络业务的请求  $S_i$  由一系列的网络功能按次序组成， $S_i = (s_{i,1}, s_{i,2}, s_{i,3}, \dots)$ ，其中， $s_{i,j}$  为  $S_i$  业务的第  $j$  个网络功能。这个网络服务请求会被映射成一条服务链， $|S_i|$  为每个服务链的长度。每个网络功能对应一种 VNF 来实现，网络功能  $s_{i,j}$  所需要的 VNF 用  $f(s_{i,j})$  表示，部署网络功能  $s_{i,j}$  的节点位置用  $v(s_{i,j})$  来表示。假设任意 2 个节点上部署的 VNF 之间的流量都按照最短路径策略进行路由转发，表 1 为本文的变量及意义。

将服务链的平均链路延时和服务器的负载作为优化目标。

**优化目标 1** 最小化服务链的平均链路延时

$$\bar{D} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{|S_i|-1} D(v(s_{i,j}), v(s_{i,j+1})) \quad (1)$$

**优化目标 2** 使服务器的负载最均衡

$$\widetilde{C}_{v_i} = \frac{1}{|V|} \sum_{i=1}^{|V|} \widetilde{C}_{v_i} \quad (2)$$

$$\sigma^2 = \frac{1}{|V|} \sum_{i=1}^{|V|} (\widetilde{C}_{v_i} - \overline{C}_{v_i})^2 \quad (3)$$

使用简单的线性加权法将 2 个优化目标进行组合，得到最终的优化目标

$$\text{Minimize: } \alpha \bar{D} + \beta \sigma^2$$

$$\text{Subject to: } \alpha + \beta = 1, \alpha > 0, \beta > 0$$

$$\widetilde{C}_{v_i} \leq \overline{C}_{v_i}, \forall v_i \in V$$

$$D(v_i, v_j) < \infty, \text{ 即 } v_i \text{ 和 } v_j \text{ 节点之间有路径}$$

## 4 系统设计

服务链技术将在 5G、物联网、工业互联网等新兴应用场景中大显身手，可以利用 SDN/NFV 技术和人工智能技术实现网络资源的智能调度和服务链的映射。SDN 控制器作为网络的大脑，可以赋予强化学习的能力。能够实时收集网络的状态和行为数据，捕获路由信息、流量信息、拓扑链路，实时遥测、分析、计算、优化、控制。同样，NFV MANO 平台也可以集成智能管理的能力，能够统一化、精细化、智能化管理计算、存储、网络三维资源。基于两者的协同控制就可以形成自动化闭环管理，可以构建自学习、自愈、自优化、自服务的智能网络。

表 1

变量及意义

变量	意义
$G = (V, E)$	网络拓扑
$v_i \in V$	其中一个节点
$e_i \in E$	其中一条链路
$C_{v_i}$	服务器节点总的 vCPU 数量
$\widetilde{C}_{v_i}$	服务器节点已经使用的 vCPU 数量
$D(v_i, v_j)$	节点 $v_i$ 和节点 $v_j$ 之间的最短路径的链路延时
$F = (f_1, f_2, f_3, \dots)$	系统支持的 VNF 的类型
$S_i = (s_{i,1}, s_{i,2}, s_{i,3}, \dots)$	服务链请求，由不同的网络功能按次序组成
$f(s_{i,j})$	网络功能 $s_{i,j}$ 所需要的 VNF 类型
$ S_i $	服务链的长度
$v(s_{i,j})$	$s_{i,j}$ 部署的服务器节点位置

基于 SDN/NFV 技术构建的新型网络毫无疑问具有高度复杂性和内在分布性，而且处于开放变化的动态环境中。多 agent 技术主要研究一组自治的 agent 在分布式开放的动态环境下，通过交互、合作、竞争、协商等智能行为完成复杂的控制。本文提出一种基于多智能体强化学习技术的服务链资源调度架构。

#### 4.1 系统架构

在 NFV MANO 架构中，SDN orchestrator (SDN-O) 负责端到端的网络连接服务、服务资源编排、流量工程等；一般会管理多个域内控制器或超级控制器。NFV orchestrator (NFV-O) 负责网络服务的生命周期管理、VNF 部署等；一般部署在数据中心或企业网等。Global service orchestrator (GS-O) 负责跨域之间的协同编排、端到端的服务编排等。一般会管理一个 SDN-O 以及多个 NFV-O，同时一个 SDN-O 会管理一个或多个 SDN 域 (SDN domain)。每个 SDN 域有相应的控制器侧重实现网络资源抽象、路径计算等功能。NFV-O 一般部署在数据中心或企业网，而 SDN 控制器依据其应用在数据中心、广域网、光传输网等不同场景进行细分。

本文系统架构在基础设施平面、控制管理平面的基础上增加一个智能服务平面，在智能服务平面上构建由 SDN domain agent、NFV-O agent、GS-O agent 等部件组成的多 agent 智能协同管理系统。根据架构设计的需要，定义几个新的功能实体，如下所示。

1) SDN domain agent。基于 SDN 控制器进行网络状态数据的收集、分析、策略执行。具体包含监控引擎、分析引擎、策略引擎。其中，监控引擎主要通过南向接口协议的统计分组、发送探测分组等手段获得链路拓扑、质量、延时、利用率等网络的全局监控信息。

2) NFV-O agent。通过 NFV-O、VNF manager、VIM 等系统部件完成网络服务、VNF 等信息的收集、分析、策略执行。NFV-O agent 亦包含监控引擎、分析引擎、策略引擎。其中，监控引擎主要通过 VIM、VNF manager 等渠道实时掌握基础设施资源使用情况、网络服务的运行情况等。

3) GS-O agent。负责全局资源及业务信息的收集、分析、策略执行。根据 SDN domain agent 以及 NFV-O agent 搜集的信息进行全局的精细化智能，实现 SDN 域和 NFV 域的有机协同。其同样包含监

控引擎、分析引擎、策略引擎。

由于 SDN domain agent、NFV-O agent、GS-O agent 的分析引擎和策略引擎需要根据业务场景等灵活构建，涉及技术手段繁多，本文简要介绍其功能作用。分析引擎先将监控引擎存储到数据库的信息进行聚合和关联，如数据规范化、删除冗余信息等。然后全面分析预测网络的未来状态，通过数据挖掘和人工智能等技术手段对未来趋势进行研究判断。策略引擎通过对网络状态的诊断，决策产生主动或被动的措施或策略，避免出现已经发现或可能出现的网络问题，并通过 SDN 控制器或 NFV-O 来执行具体的策略。

本文系统架构将构建一个层次化的多 agent 强化学习管理体系，该体系中的每一个 agent 都代表一定的角色。其中，GS-O agent 是全局管理者，主要担当网络服务的总体控制决策功能，根据网络服务特征、拓扑结构、需求预测及工作负载等因素对整个基础设施的运行状况做出评估，其目的是寻求总体控制效果最优。SDN domain agent 或 NFV-O agent 作为局域控制 agent，由若干区域协调控制系统构成，接收局部信息和限制条件，在接收 GS-O agent 综合评估的基础上，自主决定相应的控制策略，进而达到最优化控制本域基础设施运行的目标，并及时将业务需求预测和控制效果传送给 GS-O agent，作为后续评估的参考。该架构可以支持多个域之间协同编排，针对不同应用场景的跨域服务链。具体系统架构如图 1 所示。

#### 4.2 系统控制及流程

由于多 agent 学习离不开 agent 之间的通信，因此基于层次化的多 agent 强化学习通过梳理不同职能的 agent 之间的管理关系以及共享智能平面的信息数据，大大减少了各 agent 之间的通信开销。各自 agent 主要采用强化学习的方式进行学习，通过不断与基础设施和业务服务环境的接触获取控制经验并改善控制行为。通过外部环境提供的强化信号（奖励或惩罚）对控制动作的好坏做出客观评价。通过这种方式，各 agent 在行动—评价的体系中获得知识，改进行动方案以适应外部环境需求。同时每一个 agent 强化学习框架结构包含监控引擎、分析引擎、策略引擎，分别对应状态感知器、学习器、动作选择器等能力。状态感知器把环境状态映射成 agent 的内部感知；学习器根据环境状态的奖励值以及内部感知，更新 agent 的策略知识；动作

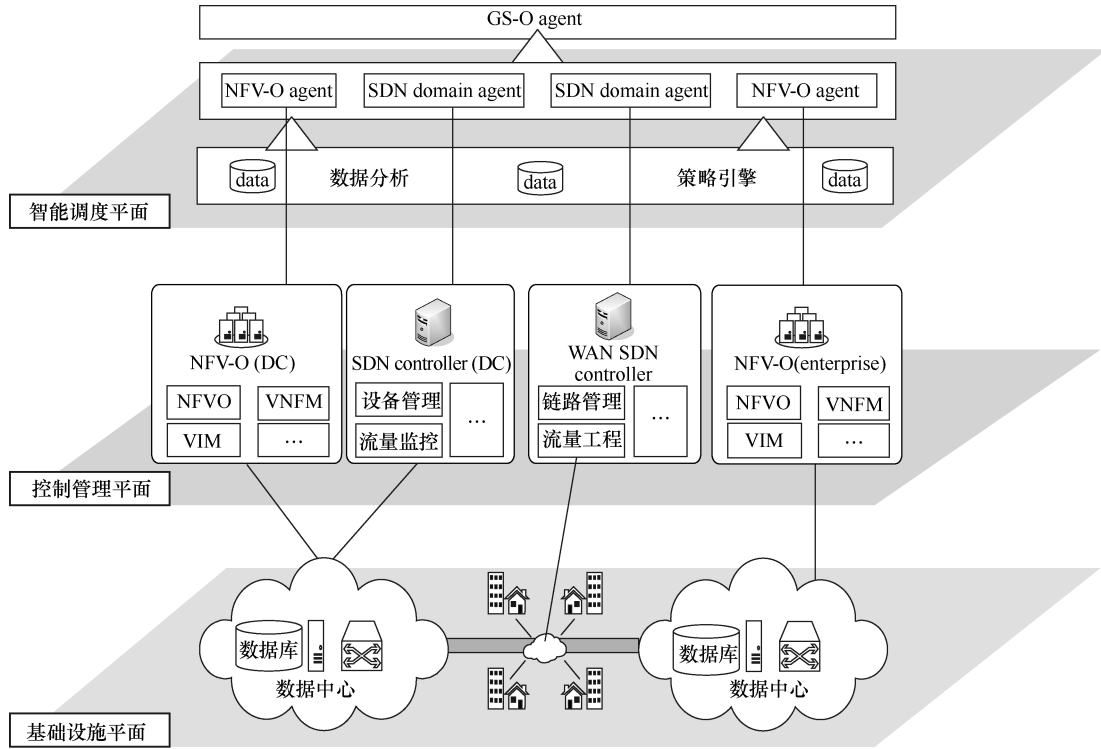


图 1 基于多 agent 强化学习的资源调度系统架构

选择器根据当前策略选择动作作用于环境。这样整个系统控制如图 2 所示，非常适合以强化学习的算法框架进行建模处理。

在这个多 agent 强化学习服务链调度体系架构

中存在 agent 之间的协作。如一般 SDN controller 和 NfV-O 部署在同一物理场所（如数据中心等）。因此，SDN domain agent 和 NfV-O agent 之间可以充分协作，交互同步虚拟网元、网络接入端口、网

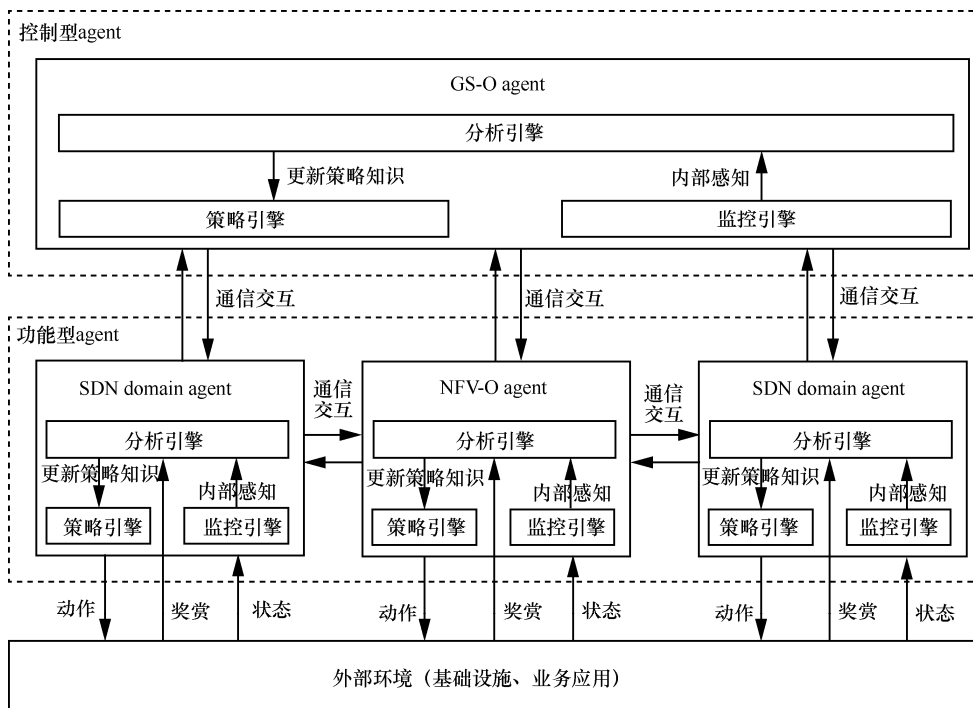


图 2 基于多 agent 强化学习的资源调度控制架构

络流量等信息。此外，2 个 SDN 域之间也存在协同调度，如一个数据中心 SDN 控制器和一个广域网 SDN 控制器等。同样 2 个 NFV 域之间有时也需要动态交换计算、存储、网络的资源分配情况以及虚拟机的生成销毁情况等。根据任务类型分类，可以认为各 agent 之间是充分合作的，可以双向通知感知对方状态。

系统可以承载虚拟化核心网、虚拟化内容分发网络、云安全防护等多种业务，动态智能调度底层计算、存储、网络资源并依据业务特性进行编排。当需要进行服务链映射时，其处理流程如图 3 所示。

- 1) 用户向 GS-O 发起服务链构建请求，GS-O agent 根据请求描述输出放置决策。
- 2) GS-O 进行响应检查可用资源并进行资源预留，通知 NFV-O 和 SDN controller 执行具体资源分配。
- 3) NFV-O 向 NFV-O agent 请求资源分配策略，实例化 VNF。
- 4) SDN controller 向 SDN domain agent 请求 SDN 组链策略，下发流表进行流量调度构建服务链。
- 5) GS-O 给用户反馈服务链构建成功信息。

### 5 基于强化学习的服务链映射算法

强化学习是人工智能领域机器学习技术中的重要方法之一，它区别于监督学习、无监督学习的地方在于其与环境交互的特点。强化学习是一种交互式的学习方法，强调在与环境的交互中学习获得

评价性的反馈信号，在不需要给定各种状态下输入信号的期望输出情况下，通过最大化未来回报为学习目标。因此，强化学习具有自学习和在线学习的优点，在求解先验信息较少的复杂优化决策问题中具有广泛的应用场景。

强化学习运用智能体对环境做出主动试探，不断地试错交互来进行学习。智能体首先感知环境状态，并根据环境当前状态和内部控制策略选择一个动作并执行。环境迁移到新状态后，给出一个新的强化信号对智能体选择动作的好坏做出评价。智能体根据环境反馈的奖赏值计算回报并更新内部控制策略。然后根据新的强化信号和当前环境状态选择下一个动作。智能体选择动作的原则是使从环境中获得正强化信号的概率增大，受到惩罚的概率变小。智能体应选择能最大化长期奖赏之和的动作。因此，在利用强化学习解决现实问题过程中，最重要的是将一个实际的问题转化成为强化学习的模型并利用相关的算法来得到优化的策略结果。即根据所要解决的实际问题，定义环境中的状态集、动作集和反馈函数。在服务链映射问题中，定义如下。

**定义 1** 状态集。为服务链中各 VNF 选择物理服务器节点进行部署的关键在于根据各个节点的实时状态选择合适的放置节点。首先，基于物理服务器节点已经使用的 vCPU 数量定义单个节点的状态，为每个节点的 vCPU 使用量设置一个门限值，

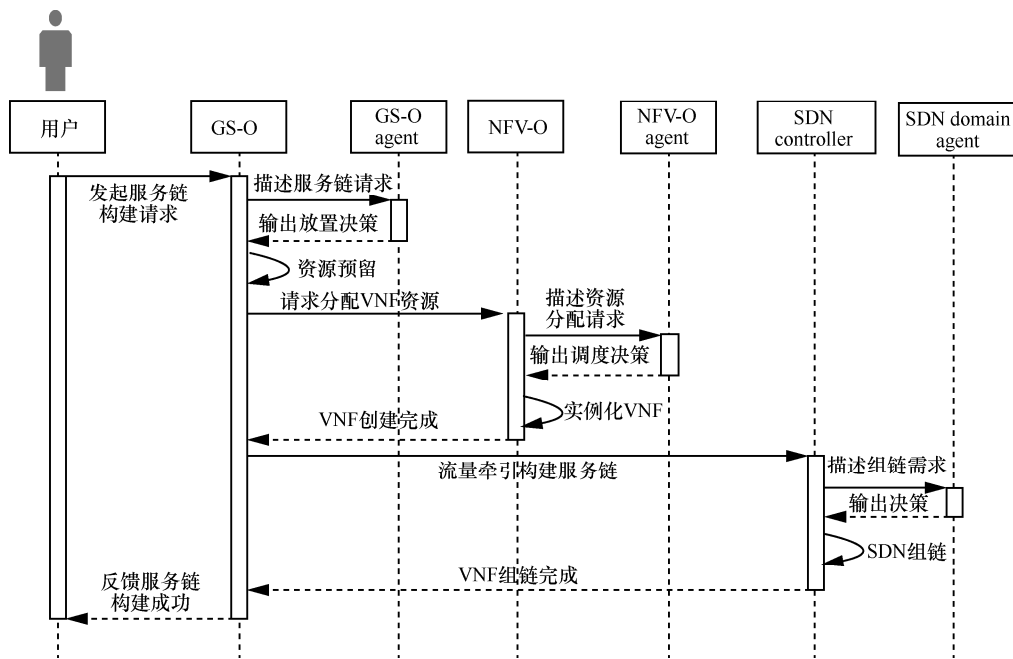


图 3 基于多 agent 的服务链映射资源调度流程

假设超过该门限值, 则该节点状态为 1, 否则, 节点状态为 0。然后考虑全局节点的状态, 假设有  $N$  个节点, 每个节点有 0 和 1 这 2 种状态, 则全局有  $2^N$  种状态。如用二进制表示状态值为“0010”, 则表示其中一个节点的资源使用超过门限, 其余节点没有超过门限。除了全局节点的状态, 由于业务流量需要按顺序经过服务链中的 VNF, 因此, 当前 VNF 的部署位置与其在服务链中的上一个 VNF 的部署位置有一定的联系, 流量会在这 2 个相邻的 VNF 节点位置之间进行转发。所以本文将上一个 VNF 部署的节点位置引入状态信息中, 假设有  $N$  个节点, 则有  $N$  种可能。综上, 状态集中的状态个数为  $2^N N$ 。状态空间表示为

$$S = \{s_1, s_2, s_3, \dots, s_n\}$$

其中,  $n = 2^N N$ 。

**定义 2** 动作集。根据每个时刻的状态信息, 需要为当前需要部署的 VNF 选择一个物理服务器节点进行部署, 假设有  $N$  个节点, 则动作集中的每个动作对应一个节点, 则有  $N$  种动作。动作集为

$$A = \{a_1, a_2, a_3, \dots, a_N\}$$

**定义 3** 反馈函数。针对不同的情况, 设置不同的反馈值, 当动作选取的节点资源使用已经超过门限值时, 设置反馈值为固定的 -50; 当动作选取的节点资源没有超过门限值时, 反馈函数为

$$R = 0 - A[L_r(s_t) - L_r(s_{t+1})] - BD_r \quad (4)$$

其中,  $L_r(s)$  表示状态  $s$  时系统的负载均衡性, 计算方法为全部节点已用 vCPU 的方差,  $D_r$  为该动作选择的节点与上一个 VNF 部署节点之间路径的最小延时。

此外, 由于强化学习算法存在维数灾难、收敛速度慢、探索和利用平衡等问题, 本文算法将重点考虑在一个数据中心或企业网内部署服务链的情形, 暂时不考虑需要同时在多个域部署某服务链的情况。即暂时不考虑多个 SDN 域和 NFV-O 域, 仅考虑服务链在某一个基础设施上的映射。另外, 在服务链的组链阶段基本通过 SDN 控制器下发流表控制不同 VNF 之间的流量走向, 一般都按最短路径策略进行选路。

### 5.1 基于 Q-learning 的服务链映射

强化学习任务通常用马尔可夫决策过程 (MDP, Markov decision process) 来描述。由于系统建模的马尔可夫决策过程中模型的状态转移概率未知, 本

文在模型中采用 Q-learning 算法来进行学习控制。Q-learning 算法是一种模型无关的值迭代方法, 不需要提前知道状态转移矩阵, 以求解具有延时回报的序贯优化决策问题。

Q-learning 算法中, 关键在于定义一个状态—动作矩阵  $Q(s, a)$ , 矩阵中的每一个值表示一个“状态—动作”对的值。每次在选取动作时, 根据  $Q(s, a)$  矩阵来选取, 假设当前的状态为  $s_t$ , 则选取的动作  $\tilde{a}$  需要满足

$$Q(s_t, \tilde{a}) = \max_a \{Q(s_t, a)\} \quad (5)$$

在本文中, 采用  $\epsilon$ -贪婪的探索方式来选取动作。这一机制以小概率  $\epsilon$  均匀随机选择一个动作进行探索, 以概率  $1 - \epsilon$  来根据  $Q$  矩阵选取当前最佳动作, 这一机制主要是为了防止算法落入局部最优的状态中, 通过小概率的探索机制, 跳出局部最优的状态。在算法设计中,  $\epsilon$  并不是固定的, 而是动态变化的,  $\epsilon = \frac{10}{K}$ ,  $K$  为学习循环的次数。这样做的意义在于在学习的前期动作选取上采用随机方式的概率更大, 能更好地去探索。而在学习的后期, 则动作选取上更偏向于根据  $Q$  矩阵计算生成。

当动作选定后, 执行该动作, 系统进入下一状态  $s_{t+1}$ , 同时也能得到系统的一个反馈  $r(s_t, a_t)$ 。根据式(6)对  $Q$  矩阵进行迭代更新。

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha_t [r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (6)$$

其中,  $(s_t, a_t)$  为马尔可夫决策过程在时刻  $t$  的“状态—动作”对;  $s_{t+1}$  为时刻  $t+1$  的状态;  $r(s_t, a_t)$  为时刻  $t$  的回报;  $\alpha_t > 0$  为学习因子。

### 5.2 算法描述

结合 5.1 节给出具体的算法实现描述, 算法 1 侧重描述整个服务链的映射算法过程, 如下所示。

#### 算法 1 基于 Q-learning 服务链映射算法

接收到服务链放置请求

检查基础设施计算存储网络资源是否满足服务链请求资源总量需求

if (满足资源总量需求)

    将服务链的头节点放置在负载最小的节点上

    for (后续服务链需要放置的 VNF)

        根据 Q-learning 算法选择合适的 VNF 放置节点

else

拒绝服务链放置请求

算法 2 侧重描述基于 Q-learning 的 VNF 放置过程，如下所示。

**算法 2** 基于 Q-learning 的 VNF 放置算法

接收到 VNF 放置请求

产生一个(0,1)之间的随机数  $\gamma$

if( $\gamma < \epsilon$ )

随机选择动作对 VNF 进行放置，即随机选择节点对 VNF 进行放置

else

根据  $Q$  矩阵计算选择可选动作将 VNF 放置在合适节点

计算 VNF 放置后的回报  $r(s_t, a_t)$

根据式(6)更新  $Q$  矩阵用于下一次计算放置策略

## 6 算法仿真与分析

### 6.1 仿真环境

本文基于 Java 语言编写测试验证工具来实现算法的仿真。算法的仿真实现分为 3 个模块，首先是底层网络环境构建，包括网络拓扑节点和物理服务器节点资源的模拟等；然后是业务的生成模块，每个业务包含一个服务链，业务以泊松流到达，在系统中完成的时间服从负指数分布，所以在系统中停留一定的时间后也以泊松分布离去。最后是基于增强学习的服务器映射算法模块，该模块运行着 Q-learning 算法，为业务中服务链的部署做决策，然后在每次部署后查看系统的下一状态和反馈，更新  $Q$  矩阵。

#### 1) 服务链的产生

假设网络中存在的 VNF 种类的个数为 10 种。服务链的长度从 [1, 8] 中按均匀分布随机产生。服务链中的每个 VNF 的产生从集合中随机选取，并保证选取过的 VNF 种类不再选取，使一条服务链中的 VNF 种类都不相同。

服务链请求按泊松过程产生，即产生的时间间隔服从指数分布  $e^{-\lambda t}$ ，其中， $\lambda=1$ ；同时服务链在系统中持续的时间服从指数分布  $e^{-\mu t}$ ，其中， $\mu=1$ 。系统根据上述概率分布规律进行服务链生命周期的管理。

#### 2) 实验拓扑产生

网络拓扑使用 Brite 拓扑产生器产生，一共选取产生了 10 个节点网络拓扑，节点之间的连接关

系和链路的时延都由拓扑产生器产生。假设每个服务器节点有 10 个 vCPU，本文假设每个 vCPU 能承载一个 VNF，所以每个服务器节点上可以放置 10 个 VNF。

#### 3) 基于 Q-learning 的服务链映射算法

Q-learning 算法的核心在于  $Q$  矩阵，由于实验的网络拓扑中为 10 个节点，所以，状态集中状态的个数为 10 240，动作集中的动作个数为 10， $Q$  矩阵中“状态—动作”对的个数为 10 240×10。每到来一个服务链的请求，就会进行一次部署，然后将  $Q$  矩阵进行一个更新。对算法进行评估时，是当  $Q$  矩阵收敛后，利用收敛后的  $Q$  矩阵来为服务链的部署做决策，以测试算法的性能。

### 6.2 对比算法

为了对比验证本文提出算法的有效性，选择了以下几种经典的服务链映射算法进行比较分析。

RL (reinforcement learning)。强化学习算法，即本文算法。

SPG (short path greedy)。如果上一个 VNF 部署的节点有可用的资源，则在上一个 VNF 部署的节点上放置当前 VNF。如果上一个 VNF 部署的节点上没有可用的资源，则在链路传输延时最短的节点上放置，即在距离上一个 VNF 部署节点传输延时最小的节点上放置，SPG 的目标是为了减少业务数据分组在服务链中转发的时延。

LLG (least loaded greedy)。每个 VNF 都选择在负载最小的服务器节点上放置。

RD (random)。随机选择服务器节点放置 VNF。

### 6.3 评价指标

由于服务链放置涉及资源利用率、网络延时、系统能耗等指标，暂时先选取 2 个重要的评价指标来衡量本文提出算法的有效性：1) 每个业务的服务链的平均传输延时；2) 系统中服务器节点的负载均衡性。本文算法适用于优化不同的目标，只需要更改反馈值即可。

#### 1) 业务的平均传输延时

记录每个业务在映射完成后业务链整体在链路上的传输延时，然后取平均值。如某个业务链的形式为  $VNF_1 \rightarrow VNF_2 \rightarrow VNF_3$ ，则该业务的传输时延为  $VNF_1$  所在节点  $\rightarrow VNF_2$  所在节点  $\rightarrow VNF_3$  所在节点的链路传输时延。其中，如果 2 个 VNF 部署在同一节点上，则 2 个 VNF 间的传输时延极小，可以忽略不计，即  $D(V(s_{ij}), v(s_{i,j+1}))=0$ 。如果 2 个 VNF 部

署在不同节点上,  $D(V(s_{ij}), v(s_{ij+1}))$ 为 2 个节点之间路径的最小时延。

### 2) 服务器节点负载均衡

在系统运行过程中, 在不同时刻采集不同物理服务器节点上的负载情况, 然后, 计算不同服务器节点上负载情况的方差来衡量系统的负载均衡。方差越小表明系统负载越均衡; 方差越大表明系统负载波动越大。

## 6.4 实验结果

### 1) 业务的平均传输延时结果

按照收敛后的  $Q$  矩阵来部署服务链, 对算法性能进行评估。实验中随机生成 500 条服务链并按照不同的算法进行部署实验, 然后统计服务链承载业务的平均链路延时, 业务平均链路延时指标对比如图 4 所示。

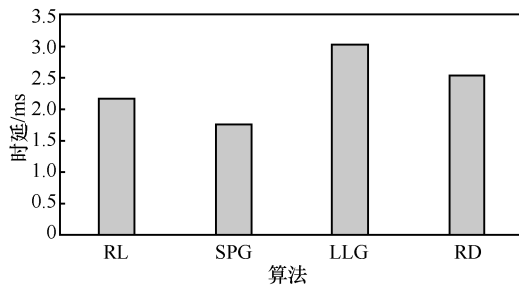


图 4 业务平均链路延时指标对比

如图 4 所示, SPG 在业务平均链路延时上最小, 这是因为每次都选择在同一个节点上部署同一条服务链上相邻的 2 个 VNF, 或在链路时延最小的 2 个节点上去部署相邻的 2 个 VNF, 这样可以有效地降低业务的时延。RL 算法业务平均链路时延上的表现仅次于 SPG, 这是因为该算法在强化学习的奖惩反馈中加入了关于链路延时优化的激励, 当链路延时越小, 则得到的奖赏越大, 所以策略选择上会选择延时较小的节点去部署 VNF。LLG 的表现最差, 这是因为每次 LLG 都选择负载最小的节点去

部署 VNF, 导致同一条服务链中的 VNF 可能分散部署在多台服务器节点上, 使链路时延增大。

### 2) 系统负载均衡结果

当仿真平台运行过程中, 业务按泊松流到达, 按泊松流离去, 则在系统中会存在一些正在处理的业务链, 这些业务链的部署情况, 系统的负载情况是否比较均衡, 也是系统需要评估的重要因素。

在系统运行过程中, 按一定间隔采集系统当前的负载均衡状况, 即监测节点 vCPU 资源使用量, 然后统计不同节点资源使用量的方差值, 共采集 20 组数据来计算方差, 然后取 20 次方差的平均值。本文用方差的平均值来描述系统的均衡性, 如图 5 所示。

统计应用不同算法的系统负载均衡情况, 计算其平均值。系统均衡性的指标对比如图 5 所示。

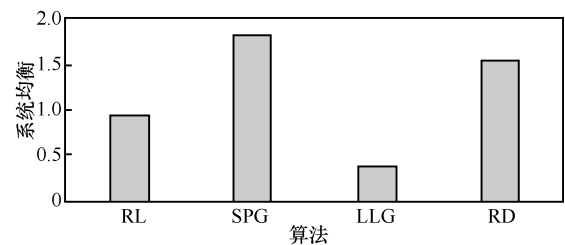


图 5 系统均衡性的指标对比

如图 5 所示, LLG 的波动小且系统的负载方差值小, 说明系统负载比较均衡, 表现最佳, 这是因为 LLG 每次都将在服务链中各 VNF 部署在负载最小的节点上。RL 算法的表现比 LLG 稍差, 这是因为在算法奖惩反馈中考虑了系统负载的因素, 策略每次选择会将服务链中各 VNF 部署在负载较小的节点上, 使系统的负载能够比较均衡。SPG 算法表现得最差, 这是因为 SPG 为了减少链路的时延, 常常将相邻的 2 个 VNF 部署在同一个节点上, 导致系统的负载均衡性表现较差。

为了更加直观地反映底层网络的负载均衡性, 将这 20 次的方差数据按时序表示为图 6, LLG 在不同

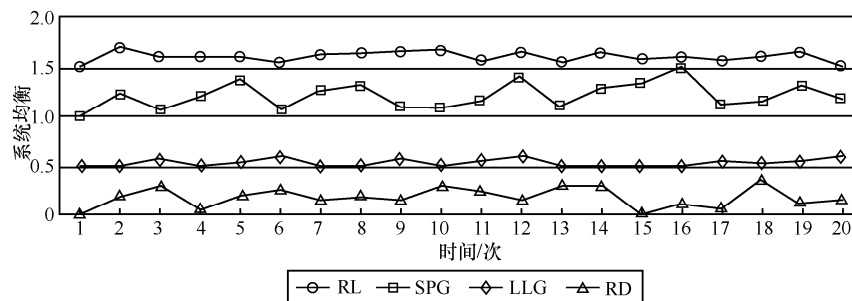


图 6 20 次系统负载均衡方差数据对比

时刻的节点负载方差值基本都接近 0, 说明在不同时刻, 不同节点上的负载基本是相同的, RL 算法在不同时刻的方差值也比较小。

## 7 讨论

根据互联网公司和电信运营商的实操经验, 一般单一网络业务链部署所需服务器数目为几台到几百台不等, 具体视负载均衡、安全防护、内容分发等不同业务需求而定, 本文在实验中选择 10 个节点的拓扑进行验证性实验, 具有一定的参考价值。

本实验中选择策略矩阵作为 Q-learning 算法的策略表示。由于策略矩阵存在着天然的可拓展性问题, 导致该算法不能支持大规模的网络拓扑。因此, 针对算法的可拓展性问题, 本文提出以下改进的思路作为未来的研究工作。首先, 当节点数量较多时, 可以选择根据经验将策略矩阵进行初始化, 使策略矩阵接近最优的结果, 来加速策略矩阵的收敛。其次, 可以使用深度神经网络来表述状态—动作集合, 业界已经在游戏场景中使用深度神经网络表征状态—动作集合, 证明了深度神经网络可以适用于复杂的场景。

## 8 结束语

服务链的映射是 NFV 架构体系中的核心, 重点解决网络服务资源的动态提供和灵活编排问题。本文设计了基于多智能体强化学习技术的服务链资源调度架构, 提出了一种基于 Q-learning 强化学习的服务链映射算法, 通过仿真验证表明在业务的平均传输延时和服务器节点负载均衡性上取得了很好的优化效果。后续工作中将针对虚拟化核心网、虚拟内容分发网络等不同业务特征驱动下的服务链映射方式展开深入研究。

### 参考文献:

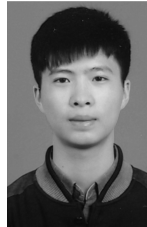
- [1] HAN B, GOPALAKRISHNAN V, JI L, et al. Network function virtualization: challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2015, 53(2): 90-97.
- [2] MIJUMBI R, SERRAT J, GORRICO J L, et al. Network function virtualization: state-of-the-art and research challenges[J]. IEEE Communications Surveys & Tutorials, 2016, 18(1): 236-262.
- [3] RIERA J F, ESCALONA E, BATALLÉ J, et al. Virtual network function scheduling: concept and challenges[C]//2014 International Conference on Smart Communications in Network Technologies (SaCoNeT). 2014: 1-5.
- [4] MIJUMBI R, SERRAT J, GORRICO J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions [C]//2015 1st IEEE Conference on Network Softwarization (NetSoft). 2015: 1-9.
- [5] KREUTZ D, RAMOS F M V, VERÍSSIMO P E, et al. Software-defined networking: a comprehensive survey[J]. Proceedings of the IEEE, 2015, 103(1):14-76.
- [6] NUNES B A A, MENDONCA M, NGUYEN X N, et al. A survey of software-defined networking: past, present, and future of programmable networks[J]. IEEE Communications Surveys & Tutorials, 2014, 16(3): 1617-1634.
- [7] LI Y, CHEN M. Software-defined network function virtualization: a survey[J]. IEEE Access, 2015, 3: 2542-2553.
- [8] BHAMARE D, JAIN R, SAMAKA M, et al. A survey on service function chaining[J]. Journal of Network and Computer Applications, 2016, 75:138-155.
- [9] KUO T W, LIOU B H, LIN K C J, et al. Deploying chains of virtual network functions: on the relation between link and server usage[C]//IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications.2016: 1-9.
- [10] MECHTRI M, GHRIBI C, ZEGHLACHE D. A scalable algorithm for the placement of service function chains[J]. IEEE Transactions on Network and Service Management, 2016, 13(3): 533-546.
- [11] WANG L, LU Z, WEN X, et al. Joint optimization of service function chaining and resource allocation in network function virtualization[J]. IEEE Access, 2016, 4: 8084-8094.
- [12] YE Z, CAO X, WANG J, et al. Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization[J].IEEE Network, 2016, 30(3): 81-87.
- [13] REDDY V S, BAUMGARTNER A, BAUSCHERT T. Robust embedding of VNF/service chains with delay bounds[C]//2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). 2016: 93-99.
- [14] MEHRAGHDAM S, KELLER M, KARL H. Specifying and placing chains of virtual network functions[C]// 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet). 2014: 7-13.
- [15] ZHANG Q, XIAO Y, LIU F, et al. Joint optimization of chain placement and request scheduling for network function virtualization[C]//2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). 2017:731-741.
- [16] HIRWE A, KATAOKA K. LightChain: a lightweight optimisation of VNF placement for service chaining in NFV[C]// 2016 IEEE NetSoft Conference and Workshops (NetSoft). 2016: 33-37.
- [17] ZHANG B, ZHANG P, ZHAO Y. Co-scaler: cooperative scaling of software-defined NFV service function chain[C]//2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). 2016: 33-38.
- [18] SAHHAF S, TAVERNIER W, COLLE D, et al. Network service chaining with efficient network function mapping based on service decompositions[C]//2015 1st IEEE Conference on Network Softwarization (NetSoft). 2015: 1-5.
- [19] MOENS H, TURCK F D. Customizable function chains: managing service chain variability in hybrid NFV networks[J].IEEE Transactions on Network and Service Management, 2016,13(4): 711-724.
- [20] FEI X, LIU F, XU H, et al. Towards load-balanced VNF assignment in geo-distributed NFV Infrastructure[C]//IEEE/ACM 25th International

Symposium on Quality of Service (IWQoS).2017: 1-10.

- [21] DIETRICH D, ABUJODA A, RIZK A, et al. Multi-provider service chain embedding with nestor[J]. IEEE Transactions on Network and Service Management, 2017,14(1): 91-105.
- [22] HABIB A, KHAN M I. Reinforcement learning based autonomic virtual machine management in clouds[C]//2016 5th International Conference on Informatics, Electronics and Vision (ICIEV). 2016: 1083-1088.
- [23] GROLEAT T, POUYLLAU H. Distributed inter-domain SLA negotiation using Reinforcement Learning[C]//12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops. 2011: 33-40.
- [24] LIN S C, AKYILDIZ I F, WANG P, et al. QoS-aware adaptive routing in multi-layer hierarchical software defined networks: a reinforcement learning approach[C]//2016 IEEE International Conference on Services Computing (SCC).2016: 25-33.

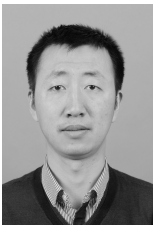


张娇（1986-），女，河北保定人，北京邮电大学副教授，主要研究方向为数据中心网络、网络功能虚拟化、软件定义网络、未来网络体系架构等。

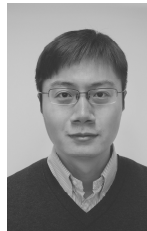


王泽南（1994-），男，浙江湖州人，北京邮电大学博士生，主要研究方向为网络功能虚拟化、网络智能等。

[作者简介]



魏亮（1981-），男，江苏扬州人，北京邮电大学博士生，主要研究方向为未来网络、软件定义网络、网络功能虚拟化等。



刘江（1983-），男，河南郑州人，博士，北京邮电大学副教授，主要研究方向为网络体系架构、网络虚拟化、软件定义网络、信息中心网络等。



黄韬（1980-），男，重庆人，博士，北京邮电大学教授，主要研究方向为路由与交换、软件定义网络、内容分发网络等。



刘韵洁（1943-），男，山东烟台人，中国工程院院士，北京邮电大学教授，主要研究方向为未来网络技术、网络体系架构、网络融合与演进等。