基于 Q-learning 的跨域服务链映射机制

张红旗，黄睿，杨英杰，常德昱，张连成
（中国人民解放军战略支援部队信息工程大学密码工程学院，河南 郑州 450001）

摘 要：针对软件定义网络功能虚拟化环境下跨域服务链映射问题，提出了一种区域集中管理、全局协同调度的虚拟服务资源管控架构。在此基础上，建立了一种有效的跨域服务链映射框架，在此框架下将跨域服务链映射问题建模为以最小化映射开销为目标的整数线性规划问题，并基于 Q-learning 机制设计跨域服务链构建请求分割算法进行优化求解。仿真实验表明该方法在平均分割时间、平均映射开销和服务链构建请求接受率等方面相较传统方法具有更优的表现。

关键词：软件定义网络；网络功能虚拟化；服务链；映射；强化学习
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000-436x.2018283

Cross-domain service chain mapping mechanism based on Q-learning

ZHANG Hongqi, HUANG Rui, YANG Yingjie, CHANG Dexian, ZHANG Liancheng
Cryptography Engineering Institute of Information Engineering University, Zhengzhou 450001, China

Abstract: A virtual service resources controlling architecture with regional centralized management and global coordinated scheduling was proposed for the problem of cross-domain service chain mapping in SDNFV environment. On this basis, an effective mapping framework was built and the cross-domain mapping problem was modeled as an ILP with the purpose of minimizing mapping cost. A partitioning algorithm was designed to solve the problem based on Q-learning mechanism under this framework. Simulation results show that the performances of this method are better than other traditional methods on average partition time, average mapping cost, and acceptance ratioof service chain mapping request.

Key words: software defined network, network function virtualization, service chain, mapping, reinforcement learning

1 引言

软件定义网络（SDN, software defined network）将复杂的控制功能从传统网络设备中剥离出来，使数据平面仅负责高速转发，而网络的控制与管理由逻辑集中的控制平面基于全局网络视图与状态进行统一决策，有效简化了网络管理[1-2]。网络功能虚拟化（NFV，network function virtualization）是针对运营商网络中网络功能静态僵化问题而提出的 SDN 解决方案[3-4]。NFV 将以专有硬件形式部署的网络功能转变为在通用服务器上运行的虚拟网络功能（VNF, virtual network function），解耦传统网络设备的软件功能和硬件载体，减少了在网络特定位置部署中间件的开销，增加了网络设备部署的灵活性。由此，借助 SDN/NFV 的软件定义网络功能虚拟化（SDNFV, software defined network function

收稿日期：2018–05–24；**修回日期**：2018–09–11
通信作者：黄睿，xjhr1009@163.com
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.2012AA012704）；国家自然科学基金资助项目（No.61402526，No.61402525，No.61502528）；郑州市科技领军人才项目（No.131PLJRC644）
Foundation Items: The National High Technology Research and Development Program of China (863 Program) (No.2012AA012704), The National Natural Science Foundation of China (No.61402526, No.61402525, No.61502528), Zhengzhou Science and Technology Talents Project (No.131PLJRC644)

virtualization）技术应运而生[5]。

作为 SDNFV 技术的典型应用，服务链近年来颇受关注[6]。服务链在利用 NFV 技术将传统网络功能虚拟化并部署在服务节点的基础上，借助 SDN 的流量集中管控功能，依据用户/业务的服务请求引导流量按序经过服务节点上的 VNF 实例，进而为用户/业务提供可定制的网络服务。每条服务链可看作由一个或多个 VNF 实例按照既定次序连接而成的逻辑视图。因此，如何设计合理的服务链映射方法完成逻辑视图向物理拓扑的映射已成为该领域的研究热点。

文献[7]提出一种可重构的服务链映射机制，采用两阶段映射方法，分别基于物理节点的服务能力和物理链路的容量约束设计优化算法选择可映射的服务节点和路由路径，但未考虑分阶段映射过程中由于服务节点间距离过大而引入的时延开销。为此，文献[8]采用一阶段映射方法，以最大化平均资源利用率和最小化平均响应时间为目标建立服务链映射多目标优化模型，并根据网络情况和映射请求联合优化服务链映射问题。为进一步提高服务链的映射效率，文献[9]在文献[8]的基础上引入带有回溯机制的启发式算法联合优化 VNF 组合和服务链映射问题，有效降低了网络整体的带宽消耗。为实现传统网络功能向 VNF 的平滑过渡，文献[10]研究了硬件设备和虚拟机混合场景下的服务链映射问题，基于软件工程流水线的理念，提出一种定制化的服务链映射算法。文献[11]研究了多网络服务供应商条件下的服务链映射问题，并基于整数线性规划和图分割的方法给出了解决方案。由此可见，研究人员已对服务链映射问题进行了诸多有益的探索，但是相关研究仅局限于单域条件下的服务链映射，尚未对跨域条件下服务链的映射问题开展深入细致的讨论。

国内清华大学毕军教授团队率先对 SDN“东西向”问题展开研究，提出一种协作式的域间 SDN 互联技术 WE-Bridge[12]，并基于 WE-Bridge 技术构建跨洲际的域间 SDN 实验床[13]，解决了目前 SDN 的研究集中在单个自治域内采用逻辑集中的控制平面而自治域的数量呈现出超线性增长的矛盾冲突[14]。受此启发，笔者认为服务链映射问题的研究不应局限于某一特定的 OpenFlow 自治域内。实际中受地理位置等客观条件的影响，VNF 往往由不同的供应商在各自治域内进行相应的维护和管理，而用户/业务的需求愈发呈现出多样化的趋势，因此，一条服务链可能需要跨越多个 OpenFlow 自治域进行映射。此时，由于各个 OpenFlow 自治域内网络拓扑连接信息及虚拟服务资源信息的独立性和封闭性，传统的单域服务链映射方法已不再适用。为此，本文基于 SDN/NFV 的典型实现方式，提出一种区域集中管理、全局协同调度的虚拟服务资源管控架构，在此基础上，建立一种有效的跨域服务链映射框架，对涉及跨域映射的服务链构建请求，以最小化映射开销为目标，设计基于 Q-learning 的跨域服务链构建请求分割算法进行优化求解，从而有效完成跨域服务链的映射。本文贡献主要包含以下 3 个方面。

1) 基于 SDN 的典型实现方式，提出一种分层分域的虚拟服务资源管控架构，纵向包含基础设施平面、逻辑控制平面和应用管理平面，横向包含各 OpenFlow 自治域、区域服务代理和全局服务代理，共同实现虚拟服务资源的区域集中管理和全局协同调度。

2) 在上述虚拟服务资源管控架构的基础上，建立一种基于轮询机制的跨域服务链映射框架。该框架下，全局服务代理采用轮询方式接受各区域服务代理上报的跨域服务链映射请求，在保证系统公平性的同时有效避免基础设施平面资源分配的冲突。

3) 针对需要进行跨域映射的服务链构建请求，设计一种基于 Q-learning 机制的跨域服务链构建请求分割算法，以最小化映射开销为目标，采用智能化的方法进行优化求解，实现跨域服务请求的及时响应及跨域服务链的高效映射。

2　虚拟服务资源管控架构

SDNFV 环境下网络功能的硬件载体和软件功能改变了以往紧密耦合的特点，实现了虚拟服务资源的灵活管控，分离形成基础设施平面、逻辑控制平面和应用管理平面，如图 1 所示。

2.1　基础设施平面

传统物理网络的基础设施平面是由各种异构网络体系相互融合形成的复杂巨网络，其具有覆盖范围广、组成结构复杂的特点，要实现如此大规模的资源管理和维护是相当困难的。本文提出了虚拟服务资源管控架构，将基础设施平面划分为多个 OpenFlow 自治域（如图 1 中 OPNFV-D1~OPNFV-D3），每个自治域由各自的 SDN 控制器和 NFV 编排器进行区域控制，管理和维护区域内的网络通信资源和虚拟服务资源。各个自治域之间相对独立，共同协作以实现为用户/业务的多样化服务链映射请求提供及时有效的响应。

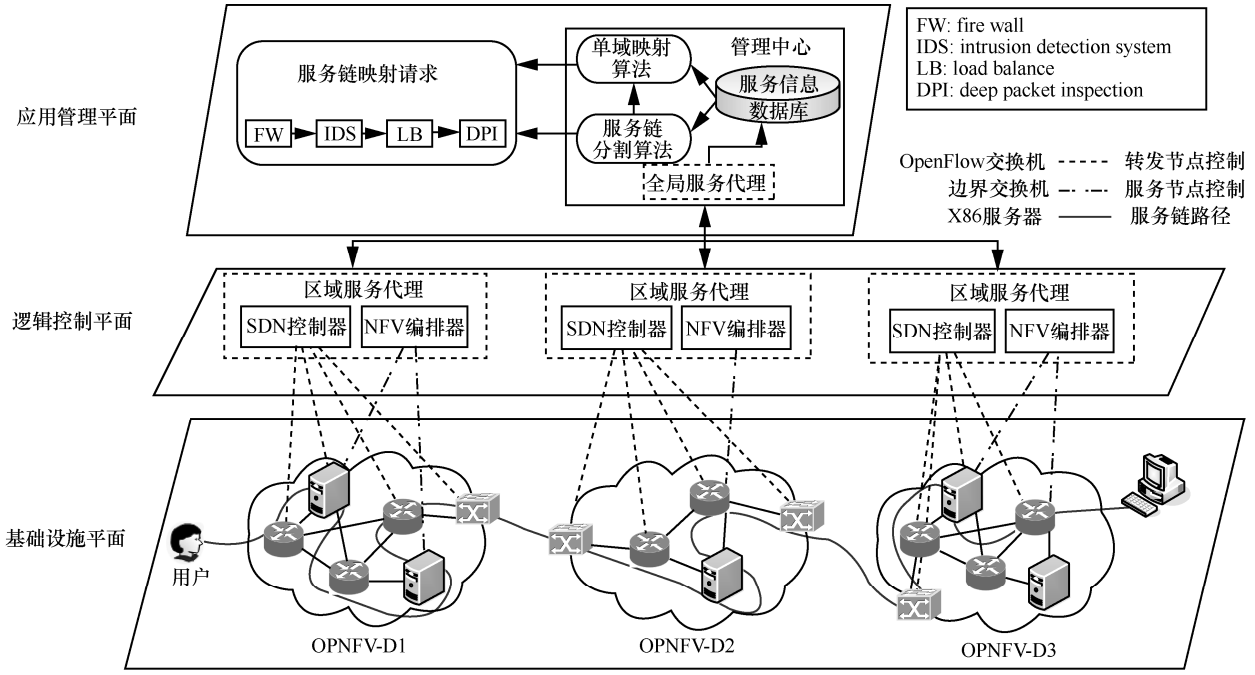


图 1 虚拟服务资源管控架构

2.2 逻辑控制平面

逻辑控制平面是上层应用管理平面和底层基础设施平面之间的“桥梁”。本文构建的 SDNFV 环境下的虚拟服务资源管控架构旨在实现虚拟服务资源的区域集中管理、全局协同调度。区域服务代理实时掌握本区域的状态信息（包括拓扑、虚拟资源的使用情况等），同时，接受区域内用户/业务的服务链映射请求，完成其向底层物理资源的映射。按照各自实现的功能和职责的不同，可将区域服务代理划分为 SDN 控制器和 NFV 编排器 2 种类型。SDN 控制器负责管理对应 OpenFlow 自治域内的交换机，实现流量的动态牵引。NFV 编排器负责管理对应 OpenFlow 自治域内提供 VNF 的 X86 服务器（也称服务节点），实现虚拟服务资源的动态调度。二者结合，为构建灵活的基础设施平面提供了保证。

2.3 应用管理平面

应用管理平面是位于逻辑控制平面之上的扩展平面，可根据需要进行扩充。本文构建了由全局服务代理、服务信息数据库、服务链分割算法和单域映射算法组成的管理中心以满足跨域服务链映射的需求。全局服务代理是为了实现分布式协同调度的需要，负责管理区域服务代理的加入或退出，并实时掌握域内边界节点以及域间链路的拓扑连接关系及使用情况。当出现跨域服务链映射请求时，统一协调各区域服务代理进行协同处理。服务信息数据库负责存储域内的虚拟服务资源信息及域间的状态连接信息，为算法提供计算依据。服务链分割算法是解决跨域服务链映射问题的核心，当服务链不需要进行分割时即转化为单域服务链映射问题，可采用相应的单域映射算法进行求解。

3 跨域服务链映射问题描述

在第 2 节的架构中，基础设施平面为满足区域集中管理、全局协同调度的需求被划分为不同的 OpenFlow 自治域，本节的跨域服务链映射问题研究正是基于此进行的。服务代理依据用户/业务的服务链映射需求及网络资源状态完成服务链的构建，从而实现逻辑映射需求与物理网络资源之间的有效匹配，如图 2 所示。

3.1 底层物理网络

如图 2 所示，底层物理网络被划分为 DN 个不同的 OpenFlow 自治域（此处 $DN=3$），可用有权无向图 $G^S=\left(N^S,L^S\right)=\bigcup_{k=1}^{RN}G_k^S\left(1\leqslant k\leqslant DN\right)$ 表示。其中，N^S 代表底层物理节点集合，包括域内节点集合 N^{Si} 和边界节点集合 N^{Sb}，域内节点集合 N^{Si} 又可细分为服务节点集合 N^{Si_S} 和转发节点集合 N^{Si_F}，对 $\forall n^s\in N^{Si\_S}$，用 $vol^I\left(n^s\right)$ 表示服务节点 n^s 的 I 类资源

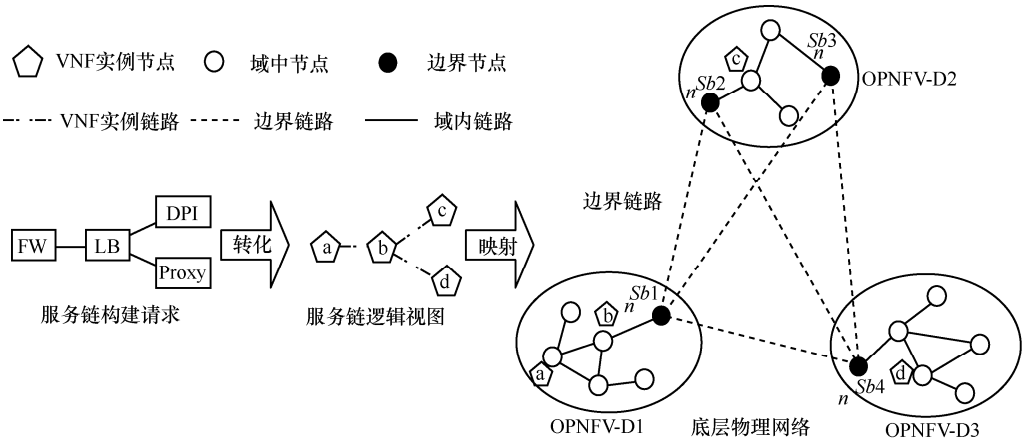


图 2　跨域服务链映射问题描述

的容量，$I=\{CPU,Memory,Throughput\}$；$L^S$ 代表底层物理链路集合，包括域内链路集合 L^{Si} 和域间链路集合 L^{Sb}，对 $\forall l^s \in L^{Sb}$，用 $bw\left(l^s\right)$ 表示其带宽资源容量。依据第 2 节的虚拟服务资源管控架构，可将整个底层物理网络视图划分为全局视图和区域视图。全局视图用 $G^{Sb}=\left(N^{Sb},L^{Sb}\right)$ 表示，描述了全网边界节点和域间链路的分布状况，而区域视图 G_k^S 描述了每个 OpenFlow 自治域内的物理拓扑连接情况。

3.2　服务链构建请求

对于包含 p 个原子服务功能的服务链映射请求 $SMR_i=\left(f^1,f^2,\cdots,f^p\right)$，可将其抽象为一个有权无向图 $G^R=\left(N^R,L^R\right)$。其中，N^R 为服务链中 VNF 实例节点构成的集合，L^R 为 VNF 实例链路构成的集合。对 $\forall n^r \in N^R$，用 $rv^I\left(n^r\right)$ 表示 VNF 实例节点 n^r 的 I 类资源需求；对 $\forall l^r \in L^R$，用 $rb\left(l^r\right)$ 表示 VNF 实例链路 l^r 的带宽资源需求。

3.3　服务链映射

服务链映射问题可形式化表示如下。*Mapping*：$G^R\left(N^R,L^R\right)\rightarrow G^S\left(N^S,L^S\right)$，按照映射区域的不同可将其划分为单域映射和跨域映射。单域服务链映射是将 VNF 实例和 VNF 链路根据一定的约束条件及目标函数映射到某一 OpenFlow 自治域范围内的底层物理网络上，而跨域服务链映射不同于单域服务链映射。因为在单域映射问题中，域内的拓扑连接情况及虚拟资源状态都是已知的，而对于跨域映射，不同的 OpenFlow 自治域之间信息都是不对外公开的。因此，需要采取分布式映射机制，实现域间映射和域内映射的协同，有效满足映射需求，并实现网络资源的充分使用。

4　跨域服务链映射策略

针对跨域服务链映射问题的特殊性，本文提出一种基于请求分割的分布式跨域服务链映射策略。首先，在第 2 节虚拟服务资源管控架构的基础上建立跨域服务链映射框架，描述服务链映射的具体流程。在此基础上，基于 Q-learning 机制设计跨域服务链构建请求分割算法，以实现跨域服务链构建请求在域间和域内的协同映射。

4.1　跨域服务链映射框架

如图 3 所示，为提高服务链构建请求的映射成功率，简化处理流程，可将映射框架设置如下。

1) 用户/业务依据就近原则向区域服务代理发出服务链构建请求，区域服务代理将其转化为服务链逻辑视图，判断其需要进行单域映射还是跨域映射，并将其划归于相应的集合 $S\_D\{\ \}$ 和 $C\_D\{\ \}$ 中。

2) 单域映射请求集合 $S\_D\{\ \}$ 按照先进先出（FIFO，first in first out）的原则由各个区域服务代理进行独立处理，依据域内拓扑连接情况及虚拟资源使用状况进行单域服务链映射。各区域服务代理可以并行处理所属域内的映射请求以提高映射效率。

3) 由于各个 OpenFlow 自治域中的网络信息是不对外公开的，因此跨域映射请求需要由全局服务代理完成。全局服务代理根据其掌握的全局资源信息及域间资源约束进行规划，将跨域服务链映射请求分割为多个单域服务链映射请求，并交由各区域服务代理进行处理。

4) 全局服务代理周期性地对各个区域服务代理进行轮询，只有轮询到的区域服务代理才能向全

局服务代理发出跨域映射请求，以避免跨域映射请求处理出现冲突和全局服务代理发生过载。

5) 跨域映射请求集合 $C\_D\{\}$ 按照 FIFO 的原则交由全局服务代理进行处理，由全局服务代理向涉及的区域服务代理发出协同处理指令。当映射完成后，各区域服务代理向全局服务代理反馈映射成功的信息。

4.2 基于 Q-learning 的跨域服务链构建请求分割

单域服务链构建请求的映射算法很多，这里不再赘述。本文关注的重点在于跨域服务链构建请求的映射。依据图 3 提出的跨域服务链映射框架，跨域映射需要被分割为多个不同的单域映射进行处理，而如何进行合理分割正是本文研究的重点。文献[15]从理论上证明了虚拟网络分割问题是非确定性多项式（NP，non-deterministic polynomial）难问题，而跨域服务链构建请求分割问题可以看作是虚拟网络分割问题的特例，因此，也难以在多项式时间内得到解决。为此，本文首先建立跨域服务链构建请求分割问题模型，然后设计基于 Q-learning 的智能优化算法求取问题的近似最优解。

4.2.1 模型建立

跨域服务链构建请求分割问题是指以降低跨域服务链映射开销为目标，根据各 OpenFlow 自治域的资源匹配状况和边界节点的相关信息，将跨域服务链映射请求分割为多个单域服务链映射请求，从而形成一套有效的服务链映射方案。

跨域服务链映射开销用 *Cost* 表示，主要包含 3 个部分：节点映射开销（ *Node_cost* ）、域间链路映射开销（ *Link_cost* ）和域内链路映射开销。对于一条固定的服务链，节点映射开销是一定的，不同的是承载路径不同而导致的不同链路映射开销。当分别位于两个 OpenFlow 自治域中的 VNF 实例节点间需要建立域间链路时，选择不同的边界节点会产生不同的 *Link_cost* 。因此，在某一服务链分割方案中，既需要为每个 VNF 实例节点指明承担其映射的 OpenFlow 自治域，还需要指明经由域中具体哪一个边界节点完成域间链路的连接。由于各 OpenFlow 自治域的链路连接信息不会完全对外公开，且域间链路映射开销与域内链路映射开销相差较大。因此，本文着重考虑节点映射开销和域间链路开销。将跨域服务链映射的总开销 *Cost* 表示为

$$Cost = Node\_cost + Link\_cost \tag{1}$$

跨域服务链构建请求分割问题可看作是满足以下映射条件，以最小化跨域服务链映射开销为目标的最优化问题。

$$f_n : n^r \to MS\left(n^r\right), n^r \in N^R \tag{2}$$

$$f_l : l^r\left(i,j\right) \to Path\left(i',j'\right), l^r\left(i,j\right) \in L^R,$$
$$i' = f_n\left(i\right), j' = f_n\left(j\right) \tag{3}$$

式(2)中的 f_n 表示 VNF 实例节点映射，即把 VNF 实例节点映射到某个 OpenFlow 自治域的边界节点上，满足 VNF 实例节点 n^r 映射条件的所有边界节点构成的集合用 $MS\left(n^r\right)$ 表示。值得注意的是，边界节点不承载具体的 VNF 实例节点映射，本文提到的将某一 VNF 实例节点映射到某一边界节点上仅表示将该 VNF 实例节点划分到该边界节点所在的 OpenFlow 自治域；式(3)中的 f_l 表示 VNF 实例链路映射，有权无向图 G^{Sb} 中边界节点 i' 和 j' 间的所有可行映射路径构成的集合用 $Path\left(i',j'\right)$ 表示。当 VNF 实例链路 $l^r\left(i,j\right)$ 的两个端点 i 和 j 分别映射到边界节点 i' 和 j' 上时，$l^r\left(i,j\right)$ 必须映射到集合 $Path\left(i',j'\right)$ 中的路径上以完成服务链映射请求的有效分割。

1) 变量定义

定义 1 分割方案矩阵 $\boldsymbol{H}_{m\times n}$ 。服务链构建请求中 VNF 实例节点的数目用 m 表示，物理网络中边界节点的数目用 n 表示。如式(4)所示，矩阵项 $\boldsymbol{H}[i][j]$ 的取值表示 VNF 实例节点 i 和边界节点 j 之间的映射关系。

$$\boldsymbol{H}[i][j] = \begin{cases} 1, \text{VNF实例节点}i\text{映射到边界节点}j \\ 0, \text{VNF实例节点}i\text{未映射到边界节点}j \\ -1, \text{边界节点}j\text{不在VNF实例节点}i \\ \text{的映射范围内} \end{cases} \tag{4}$$

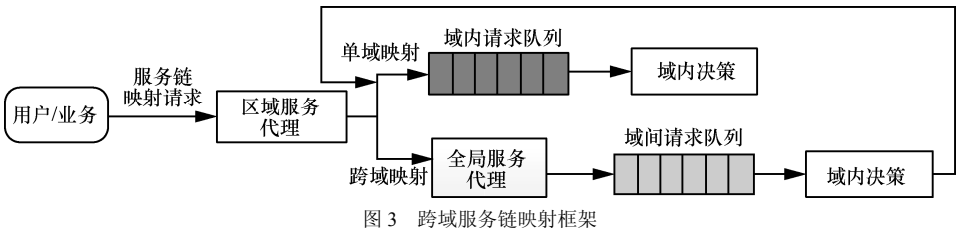


图 3 跨域服务链映射框架

以图 2 中的请求分割方案为例，可将其分割方案矩阵表示为表 1。

表 1　　请求分割方案的矩阵表示

	n^{Sb1}	n^{Sb2}	n^{Sb3}	n^{Sb4}
a	1	−1	−1	−1
b	1	−1	−1	−1
c	−1	1	0	−1
d	−1	−1	−1	1

定义 2　链路类型变量 $X_{i,j}$。其中，i 和 j 为 VNF 实例链路 $l^r(i,j)$ 的 2 个端点，可以根据矩阵 $\boldsymbol{H}$ 的值判断 VNF 实例链路 $l^r(i,j)$ 的类型，如式(5)所示，变量 $X_{i,j}$ 的值代表链路的不同类型。

$$X_{i,j}=\begin{cases}0,\text{链路}l^r(i,j)\text{为域内链路}\\1,\text{链路}l^r(i,j)\text{为域间链路}\end{cases} \tag{5}$$

2) 约束条件

跨域服务链构建请求分割问题需满足以下约束条件。

$$\sum_{j=1}^{n}\boldsymbol{H}[i][j]\leqslant 1,\forall i\in N^R \tag{6}$$

$$rb\left[l^r(i,j)\right]X_{i,j}\leqslant bw\left(l^s\right),\forall i,j\in N^R \tag{7}$$

式(6)表示矩阵 $\boldsymbol{H}$ 中的每一行之和小于或等于 1，因为每个 VNF 实例节点只能被映射到一个 OpenFlow 自治域。式(7)确保了 VNF 实例链路的带宽资源需求不超过物理链路的带宽资源容量。

3) 目标函数

跨域服务链构建请求分割问题的求解目标找到使得服务链映射开销最小的分割方案，其目标函数 Obj 可表示为

$$\begin{aligned}\text{Min}:Obj\left(\boldsymbol{H}_{m\times n},\boldsymbol{FM}_{m\times m},\boldsymbol{BM}_{n\times n}\right)\\=\alpha Obj_n+\beta Obj_l\end{aligned} \tag{8}$$

目标函数 Obj 中的 3 个参数具体含义如下。

$\boldsymbol{H}_{m\times n}$：VNF 实例节点与物理网络边界节点的映射关系矩阵。

$\boldsymbol{FM}_{m\times m}$：服务链构建请求的流量矩阵，$\boldsymbol{FM}[i][j]$ 表示 VNF 实例节点 i 和 VNF 实例节点 j 之间的带宽资源需求。

$\boldsymbol{BM}_{n\times n}$：连接边界节点间链路的最小单位开销矩阵，$\boldsymbol{BM}[i][j]$ 表示通过 Floyd 算法[16]计算得出的边界节点 i 和 j 之间所有链路的单位开销的最小值。

为便于目标函数 Obj 的计算，本文定义一种特殊的运算，用符号 $\odot$ 表示。

$$a\odot b=\begin{cases}a\times b,a\times b>0\\0,a\times b\leqslant 0\end{cases} \tag{9}$$

由于服务链的节点映射开销是一定的，而服务链的域间链路映射开销随边界节点选取的不同而不同。因此，用常数 C 表示节点映射开销 Obj_n，用 $\sum_{i=1}^{m}\sum_{j=1}^{n}\boldsymbol{FM}[i][j]\times\boldsymbol{BM}[u][v]$ 表示域间链路映射开销 Obj_l，其中，$u=\sum_{q=1}^{n}\boldsymbol{H}[i][q]\odot q$，$v=\sum_{q=1}^{n}\boldsymbol{H}[j][q]\odot q$，表示 VNF 实例节点 i 映射到了边界节点 u，而 VNF 实例节点 j 映射到了边界节点 v。目标函数 Obj 中的系数 α 和 β 用于调节节点映射开销 Obj_n 和域间链路映射开销 Obj_l 的权重。

4.2.2　算法准备

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。作为一种交互式学习方法，其强调在与环境的作用中学习获得评价性的反馈信号，并据此改进行动方案以适应环境，从而达到预期的目的。

作为强化学习的典型实现方式，Q-learning 算法常用于求解先验知识较少的复杂决策优化问题。如图 4 所示，Q-learning 算法模型是一个自适应闭环控制系统。智能体 Q-Agent 首先通过感知环境状态，在当前状态 s 的基础上依据 Q 值函数选择一个动作 a 并执行，当迁移到下一状态 s' 时，智能体 Q-Agent 依据环境的反馈计算收益函数 $R(s,a)$ 并据此更新 Q 值函数 $Q(s,a)$，然后依据新的 Q 值和当前环境状态选择下一个动作，迭代进行直至得到问题的最优策略。

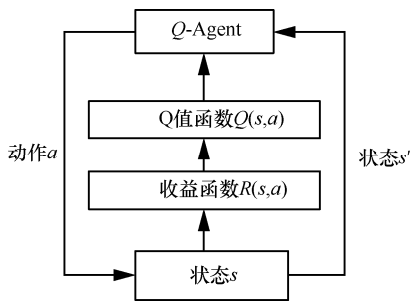


图 4　Q-learning 算法模型

Q-learning 算法中 Q 值函数是状态和行为的评价值，可用瞬时回报和未来回报来表示。

$$Q(s_t,a_t)\leftarrow R(s_t,a_t)+\gamma\max\{Q(s_{t+1},a_{t+1})\} \quad (10)$$

其中，s_t 和 a_t 表示当前状态和行为，s_{t+1} 和 a_{t+1} 表示下一个状态和行为，折扣系数 γ 是满足 $0<\gamma<1$ 的常数，用于调节智能体 Q-Agent 对未来回报的关注程度。

当系统处于状态 s_t 时，依据式(11)来选择行为 a_t。

$$a_t\leftarrow\arg\max_{a\in A}\{Q(s_t,a)\} \quad (11)$$

为了避免算法落入局部最优，本文采取 $\varepsilon-\text{greedy}$ [17]的动作选取方式，即以小概率 ε 随机选择一个动作进行探索，以 $1-\varepsilon$ 的概率根据 Q 值选取当前最佳动作，ε 可动态改变，将其设置为 $\varepsilon=\frac{E}{10}$，其中 E 表示学习循环的次数。其目的在于，智能体 Q-Agent 在学习前期可采用随机方式进行更好地探索，而在学习后期更偏向于根据 Q 值指导下一步的动作选择。

当选定并执行该动作后，系统进入下一状态 a_{t+1}，同时系统也得到相应的收益函数 $R(s_t,a_t)$，可依据式(12)对 Q 值函数进行迭代更新。

$$\begin{aligned}Q_{t+1}(s_t,a_t)=&(1-\lambda)Q_t(s_t,a_t)+\\&\lambda\left(R(s_t,a_t)+\gamma\max Q_t(s_{t+1},a_{t+1})\right)\end{aligned} \quad (12)$$

其中，$\lambda(0<\lambda\leqslant1)$ 是智能体 Q-Agent 的学习因子，可表示为 $\lambda=\frac{1}{(1+\Phi_t(num))}$，$\Phi_t(num)$ 表示经过 num 次行为选择后，行为 a_t 被选择的次数。如果在迭代过程中，某个行为被选中次数少，则在接下来的选择中偏向选择该行为，从而确保智能体 Q-Agent 拥有较好的探索特性。

在利用 Q-learning 算法解决现实问题的过程中，最重要的是将一个实际的问题转化成为 Q-learning 算法模型并以此得到优化的策略结果。即根据所要解决的实际问题，定义环境中的状态空间、动作集合和收益函数。结合本文所提的跨域服务链构建请求分割问题，可分别将其定义如下。

定义 3 状态空间 S。跨域服务链构建请求分割问题的关键在于确定 VNF 实例节点和边界节点的映射关系。因此，将 VNF 实例节点 i 和边界节点 j 组成的二元组 (i,j) 定义为一个状态 s，如果服务链中 VNF 实例节点数目为 m，物理网络中边界节点数目为 n，则共有 $m\times n$ 个状态，记为 SN。因此，状态空间可表示为 $S=\{s_1,s_2,\cdots,s_{SN}\}$。

定义 4 动作集合 A。对每一个 VNF 实例节点 i 和边界节点 j 组成的状态二元组 $s(i,j)$，可对其进行映射、不映射和不在映射范围 3 种操作，分别对应动作 a_1、a_0 和 a_{-1}。因此，动作集合可表示为 $A=\{a_1,a_0,a_{-1}\}$。

定义 5 收益函数 $R(s,a)$。对某状态 $s(i,j)$ 执行不同的动作 a 将会获得不同的收益，此处，利的式(8)计算当前方案的映射开销，取其相反数作为对应的收益，即 $R(s,a)=-(\alpha Obj_n+\beta Obj_l)$。

Q-learning 算法收敛的本质在于 Q 值函数的收敛[18]。此处对算法的收敛性进行分析，将最优 Q 值表示为 $Q^*(s_t,a_t)$。

定理 1 由式(8)定义的收益函数 R 的值在不同系统状态下有界。

证明见附录 A。

定理 2 对于有界收益函数 R 的 Q 值迭代问题，学习因子 $0<\lambda\leqslant1$，且满足

$$\sum_{t=1}^{\infty}\lambda_{\Phi_t(num)}=\infty,\sum_{t=1}^{\infty}\lambda_{\Phi_t(num)}^2<\infty,\forall s,a \quad (13)$$

则当 $t\to\infty$ 时，有

$$\lim_{t\to\infty}Q_t(s_t,a_t)=Q^*(s_t,a_t) \quad (14)$$

证明见附录 B[19]。

4.2.3 算法描述

如表 2 所示，在上述准备工作的基础上，可将基于 Q-learning 的跨域服务链构建请求分割算法描述如下。需要注意的是该算法将针对某一特定的跨域服务链构建请求中 VNF 实例节点与边界节点的映射关系进行说明。

算法 1 基于 Q-learning 的跨域服务链构建请求分割算法

输入 跨域服务链构建请求中 VNF 实例节点集合 $gv\{\}$ 和网络中边界节点集合 $gb\{\}$

输出 最优分割方案矩阵 $\boldsymbol{H}_{opt}$

初始化 Q 值函数矩阵、折扣系数 γ、学习因子 λ、计数器 $\Phi_t(num)$;

for each episode do

初始化状态空间 S;

for each step in episode do

对初始状态 s 采取 $\varepsilon-\text{greedy}$ [17]

方法执行动作 a 并依据式(8)计算收益函数 $R(s,a)$;

更新计数器 $\Phi_t(num)$;

更新学习因子 λ;

根据式(12)计算当前 Q 值函数并更新 Q 值函数矩阵;

$s \leftarrow s'$;

until Q is convergent;

end for

end for

算法 1 首先对相关参数进行初始化，随后利用 $\varepsilon-$greedy [17]方法指导智能体 Q-Agent 的行为选择，并依据式(12)进行 Q 值函数的更新，迭代进行，直至 Q 值函数收敛，从而据此为跨域服务链构建请求分割方案做出决策。

5　方案分析与评估

为全面评估本文所提跨域服务链映射策略的有效性，本文从以下两方面开展仿真实验。

1) 由于现有研究仅对单域服务链映射问题进行探讨，而未对跨域服务链映射问题开展深入研究，故现有算法都不便与本文方法直接进行比较。虚拟网络映射（VNE）问题中，有研究者专门对跨域条件下的虚拟网络映射问题进行了研究，其中，2 种较为经典的算法分别为基于二元整数规划的跨域虚拟网络映射算法（LID-partition）[20]和基于人工蜂群算法的跨域虚拟网络映射算法（ABC- partition）[21]。本文分别对这两种算法进行改造，使其适用于跨域服务链映射问题，并将它们与本文所提的基于 Q-learning 的跨域服务链构建请求分割算法（Q-partition）进行比较，以验证本文方法的性能优势。

2) 在不同物理网络拓扑连接条件下，比较本文算法在平均分割时间和平均映射开销方面的变化，以验证本文方法（Q-partition）的适应性。

5.1　实验设置

仿真实验在配置 Ubuntu 14.04 64 位操作系统的 Intel Core i7-6300HQ @3.40 GHz、8 GB 内存的 PC 机上进行。以 Mininet 模拟器为基础，采用开源的 Floodlight 控制器，使用 Java 语言编写实验代码并利用 Matlab 工具对实验结果进行分析。每次实验运行 50 000 个时间单位。

将底层物理网络划分为 10 个 OpenFlow 自治域，每个域内用 GT-ITM[22]工具生成物理网络拓扑，拓扑由 50 个节点组成，包括 6 个服务节点、40 个转发节点和 4 个边界节点，边界节点之间采用全连接的方式，域内节点以 0.5 的概率相连。为评估算法的自适应性，假设跨域服务链构建请求动态到达，且服从时间单位为 1 000，强度为 p_A 的泊松过程，请求的生命周期服从均值为 60 个时间单位的指数分布。假设网络中有 8 种 VNF，每种 VNF 有 3 种不同的 VNF 实例，每个构建请求由一个或多个 VNF 实例组成，数量服从 $[4,8]$ 的均匀分布。为便于讨论，目标函数 Obj 中的系数 α 和 β 均取 1，折扣系数 γ 取 0.8。

5.2　实验结果分析

5.2.1　算法性能对比

将本文算法（Q-partition）与 LID-partition 算法和 ABC-partition 算法进行对比，几种算法域内映射统一采用文献[23]中的算法，从平均分割时间、平均映射开销和服务链构建请求接受率 3 个方面进行分析。

1) 对比 3 种算法的平均分割时间。观察当服务链构建请求长度增加时，各算法平均分割时间的变化趋势。如图 5 所示，3 种算法的平均分割时间都随着服务链构建请求长度的增加而增加。LID-partition 算法的平均分割时间要明显高于其他 2 种算法，且随着问题规模的增大呈指数增长，在服务链构建请求长度为 8 时，平均分割时间高达 3.8×10^4 ms。这是因为 LID-partition 算法的二元整数规划中还涉及虚拟链路的分割，其中二元变量的数目要多于另外 2 种算法，故计算时间较长。Q-partition 算法和 ABC-partition 算法的平均分割时间要明显小于 LID-partition 算法，且随着服务链构建请求长度的增加呈线性增长，即使在问题规模较大时，也能够在可接受的时间范围内收敛。图 5(b)中，Q-partition 算法采用智能化的方法，不断地朝着最大收益方向调整算法的搜索方向，避免了盲目遍历全部状态空间，相较于 ABC-partition 算法表现出更好的寻优能力，因而具有最小的平均分割时间。

2) 对比 3 种算法的平均映射开销。观察当服务链构建请求长度增加时，各算法平均映射开销的变化趋势。为便于表述，对 3 种算法的平均映射开销进行“归一化”处理，假设一定请求长度下，某种算法的映射开销为 $Cost(A)$，最大映射开销为 $Cost(M)$，则可依据式(15)评估算法的平均映射开销。

$$Cost(\overline{A})=\frac{Cost(A)}{Cost(M)} \tag{15}$$

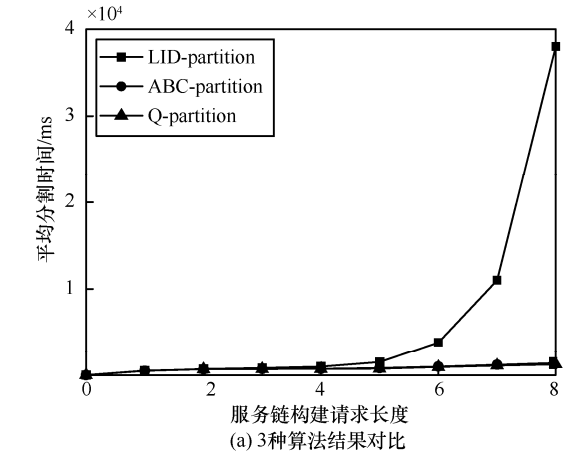


图 5　服务链构建请求平均分割时间

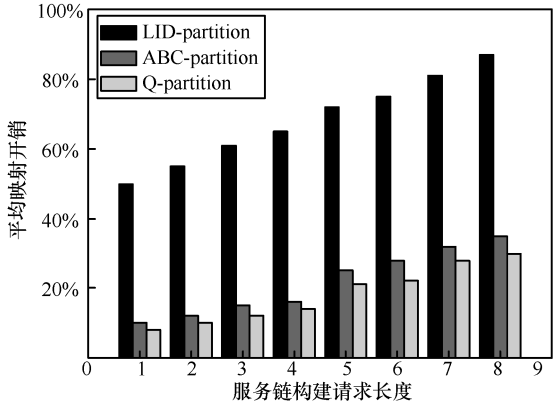


图 6　服务链构建请求平均映射开销

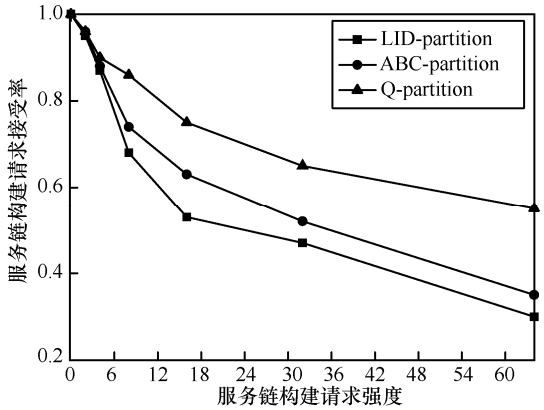


图 7　服务链构建请求接受率

如图 6 所示，3 种算法的平均映射开销都随着服务链构建请求长度的增加而增加。LID-partition 算法在服务链构建请求长度较小时（长度在1～4之间），就呈现出较高的平均映射开销值（平均映射开销在 58%左右），而随着构建请求长度的增加，平均映射开销高达 87%。是因为 LID-partition 算法中的虚拟链路分割是二次规划问题，因此在问题规模较大时，显著增加了映射的开销。Q-partition 算法和 ABC-partition 算法的平均映射开销相差不大，在实验请求长度的范围内，两者映射开销均小于 25%，并且 Q-partition 算法略优于 ABC-partition 算法，这是因为 Q-partition 算法具有更加智能的寻优能力，保证了开销的最小化。

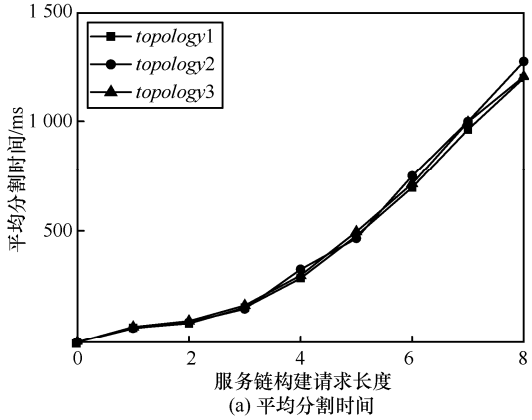
3) 对比 3 种算法的服务链构建请求接受率。观察当服务链构建请求强度增加时，各算法服务链构建请求接受率的变化趋势。为便于观察，将服务链构建请求强度 p_A 分别设置为 $(0,2,4,8,16,32,64)$。如图 7 所示，Q-partition 算法具有最优的服务链构建请求接受率，这得益于本文提出的跨域服务链映射框架。该框架下，全局服务代理采用轮询方式接受跨域服务链映射请求，在保证公平性的同时，避免了基础设施平面资源分配的冲突，因此能够更好地完成服务链构建请求的映射任务。此外，基于 Q-learning 的 Q-partition 算法能够高效地完成构建请求的映射任务，因此可以在有限时间内接受更多的服务链构建请求。

5.2.2　算法适应性评估

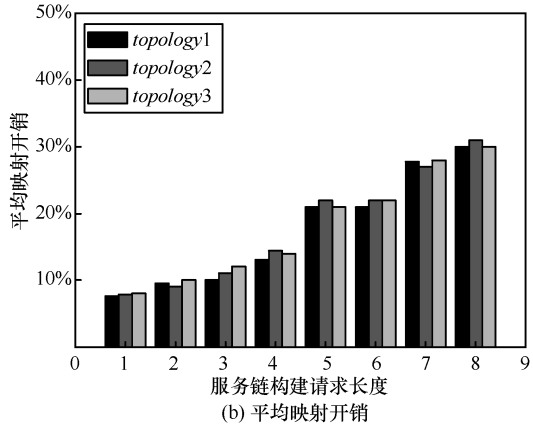
针对本文所提 Q-partition 算法，在不同物理网络拓扑连接条件下，从平均分割时间和平均映射开销 2 个方面分析算法的适应性。

上述实验中，边界节点间采用全连接的方式进行。但应指出的是，全连接的物理网络在实际运用中并不具有代表性。参考文献[24]中的实验参数设置，本节将连接概率设置为 0.2、0.5 和 1.0，分别对应 *topology*1、*topology*2 和 *topology*3，体现边界节点低概率、中概率和全连接的 3 种方式，从而有效评估算法的适应性。

如图 8 所示，Q-partition 算法并没有受物理网络拓扑变化的影响，拥有较好的适应性。



(a) 平均分割时间



(b) 平均映射开销

图 8　不同拓扑下的算法稳定性

6　结束语

本文研究了 SDNFV 环境下跨域服务链映射问题。提出了一种分层分域、区域集中管理与全局协同调度相结合的虚拟服务资源管控架构。在此基础上，针对跨域服务链映射问题，建立了一种采用轮询机制的请求响应框架，在此框架下设计了一种基于 Q-learning 机制的跨域服务链构建请求分割算法。仿真实验表明，本文方法相较传统方法在平均分割时间、平均映射开销和服务链构建请求接受率等方面具有更好的优化效果。后续工作中将针对可靠性条件下的跨域服务链映射问题进行进一步研究。

附录 A　定理 1 证明

证明式(8)由两部分组成，分别是节点映射开销 Obj_n 和域间链路映射开销 Obj_l。节点映射开销 Obj_n 是固定的常数，所以是有界的，域间链路映射开销 Obj_l 由流量矩阵和连接边界节点间链路的最小单位开销矩阵的乘积形式表示，是离散的有限值，故式(8)定义的收益函数 R 的值有界。证毕。

附录 B　定理 2 证明

证明学习因子 $\lambda=\dfrac{1}{\left(1+\Phi_t(mum)\right)}\in(0,1]$，满足式(13)的要求。

将 Q 值函数初始值定义为 $Q_0(s_t,a_t)$，对 $\forall a_t\in A,s_t\in S$，均依据式(12)更新，直至获得最优值 $Q^*(s_t,a_t)$。

对 $R(s_t,a_t)$、$Q_0(s_t,a_t)$ 和 $Q^*(s_t,a_t)$，令常量 $\varepsilon_1,\varepsilon_2,\varepsilon_3,\varepsilon_4$ 和 $\gamma(0<\gamma<1)$ 满足

$$\varepsilon_1 R(s_t,a_t)\leqslant\gamma\max Q^*(s_{t+1},a_{t+1})\leqslant\varepsilon_2 R(s_t,a_t) \tag{16}$$

$$\varepsilon_3 Q^*(s_t,a_t)\leqslant Q_0(s_t,a_t)\leqslant\varepsilon_4 Q^*(s_t,a_t) \tag{17}$$

其中，$0<\varepsilon_1\leqslant\varepsilon_2<\infty$，$0\leqslant\varepsilon_3\leqslant\varepsilon_4<\infty$，此处需证明对 $\forall\varepsilon_1,\varepsilon_2,\varepsilon_3,\varepsilon_4$，经过迭代后可以收敛于最优值 $Q^*(s_t,a_t)$。

如果 $0\leqslant\varepsilon_3\leqslant\varepsilon_4<1$，对 $\forall t=1,2,\cdots$，函数 $Q_t(s_t,a_t)$ 满足

$$\left[1+\frac{\varepsilon_3-1}{\left(1+\varepsilon_2^{-1}\right)^t}\right]Q^*(s_t,a_t)\leqslant Q_t(s_t,a_t)\leqslant\left[1+\frac{\varepsilon_4-1}{\left(1+\varepsilon_1^{-1}\right)^t}\right]Q^*(s_t,a_t) \tag{18}$$

下面，通过数学归纳法证明式(18)。

当 $t=1$ 时，有

$$\begin{aligned}Q_1(s_t,a_t)&=R(s_t,a_t)+\gamma\max Q_0(s_{t+1},a_{t+1})\geqslant\\&R(s_t,a_t)+\varepsilon_3\gamma\max Q^*(s_{t+1},a_{t+1})\geqslant\\&\left(1+\varepsilon_2\frac{\varepsilon_3-1}{1+\varepsilon_2}\right)R(s_t,a_t)+\gamma\left(\varepsilon_3-\frac{\varepsilon_3-1}{1+\varepsilon_2}\right)\max Q^*(s_{t+1},a_{t+1})=\\&\left(1+\varepsilon_2\frac{\varepsilon_3-1}{1+\varepsilon_2}\right)\left[R(s_t,a_t)+\gamma\max Q^*(s_{t+1},a_{t+1})\right]=\\&\left(1+\frac{\varepsilon_3-1}{1+\varepsilon_2^{-1}}\right)Q^*(s_t,a_t)\end{aligned} \tag{19}$$

同理可得

$$Q_1(s_t,a_t)\leqslant\left(1+\frac{\varepsilon_4-1}{\left(1+\varepsilon_1^{-1}\right)^t}\right)Q^*(s_t,a_t) \tag{20}$$

即当 $t=1$ 时，满足式(18)。

假设 $t=\varpi-1$，$\varpi=2,3,...$ 时，满足式(18)。那么，当 $t=\varpi$ 时，有

$$\begin{aligned}Q_\varpi(s_t,a_t)&=R(s_t,a_t)+\gamma\max Q_0(s_{t+1},a_{t+1})\geqslant\\&R(s_t,a_t)+\gamma\left[1+\frac{\varepsilon_2^{\varpi-1}(\varepsilon_3-1)}{(1+\varepsilon_2)^{\varpi-1}}\right]\max Q^*(s_{t+1},a_{t+1})\geqslant\\&\left[1+\varepsilon_2^{\varpi}\frac{\varepsilon_3-1}{(1+\varepsilon_2)^{\varpi}}\right]\left[R(s_t,a_t)+\gamma\max Q^*(s_{t+1},a_{t+1})\right]=\\&\left[1+\frac{\varepsilon_3-1}{\left(1+\varepsilon_2^{-1}\right)^{\varpi}}\right]Q^*(s_t,a_t)\end{aligned} \tag{21}$$

同理可得

$$Q_\varpi(s_t,a_t)\leqslant\left[1+\frac{\varepsilon_4-1}{\left(1+\varepsilon_1^{-1}\right)^{\varpi}}\right]Q^*(s_t,a_t) \tag{22}$$

因此，对 $\forall t=1,2,\cdots$，满足式(18)。

随后，证明当 $0\leqslant\varepsilon_3\leqslant1\leqslant\varepsilon_4<\infty$ 时，满足

$$\left[1+\frac{\varepsilon_3-1}{\left(1+\varepsilon_2^{-1}\right)^t}\right]Q^*\left(s_t,a_t\right)\leqslant Q_t\left(s_t,a_t\right)\leqslant\left[1+\frac{\varepsilon_4-1}{\left(1+\varepsilon_2^{-1}\right)^t}\right]Q^*\left(s_t,a_t\right) \tag{23}$$

由式(19)和式(21)可证式(23)的左半部分，这里不再赘述。对于式(23)的右半部分，令 $t=1$，有

$$\begin{aligned}Q_1\left(s_t,a_t\right)&=R\left(s_t,a_t\right)+\gamma\max Q_0\left(s_{t+1},a_{t+1}\right)\leqslant\\&R\left(s_t,a_t\right)+\varepsilon_4\gamma\max Q^*\left(s_{t+1},a_{t+1}\right)+\\&\frac{\varepsilon_4-1}{1+\varepsilon_2}\left[\varepsilon_2R\left(s_t,a_t\right)-\gamma\max Q^*\left(s_{t+1},a_{t+1}\right)\right]=\\&\left[1+\frac{\varepsilon_4-1}{\left(1+\varepsilon_2^{-1}\right)}\right]Q^*\left(s_t,a_t\right)\end{aligned} \tag{24}$$

同上，利用数学归纳法可得式(23)的右半部分。因此，当 $0\leqslant\varepsilon_3\leqslant\varepsilon_4<\infty$，对 $\forall t=1,2,\cdots$，Q 值函数满足式(18)。

综上，对任意常量 $\varepsilon_1,\varepsilon_2,\varepsilon_3,\varepsilon_4$，当 $t\to\infty$ 时，可得式(14)。证毕。

参考文献：

[1] 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.

[2] 黄韬, 刘江, 霍如, 等. 未来网络体系架构研究综述[J]. 通信学报, 2014, 35(8): 184-197.
HUANG T, LIU J, HUO R, et al. Survey of research on future network architectures[J]. Journal on Communications, 2014, 35(8): 184-197.

[3] MIJUMBI R, SERRAT J, GORRICHO J L, et al. Network function virtualization: state-of-the-art and research challenges[J]. IEEE Communications Surveys & Tutorials, 2017, 18(1): 236-262.

[4] SUN C, BI J, ZHENG Z, et al. NFP: enabling network function parallelism in NFV[C]//2017 ACM Conference of the Special Interest Group on Data Communication. 2017:43-56.

[5] LI Y, CHEN M. Software-defined network function virtualization: a survey[J]. IEEE Access, 2015, 3(1): 2542-2553.

[6] BHAMARE D, JAIN R, SAMAKA M, et al. A survey on service function chaining[J]. Journal of Network and Computer Applications, 2016, 75(3):138-155.

[7] XIONG G, HU Y, LAN J, et al. A mechanism for configurable network service chaining and its implementation[J]. KSII Transactions on Internet and Information Systems, 2016, 10(8):3701-3727.

[8] ZHANG Q, XIAO Y, LIU F, et al. Joint optimization of chain placement and request scheduling for network function virtualization[C]//2017 IEEE International Conference on Distributed Computing Systems. 2017:731-741.

[9] 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

[10] 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.

[11] 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

[12] LIN P, BI J, WANG Y. WEBridge: west-east bridge for distributed heterogeneous SDN NOSespeering[J]. Security and Communication Networks, 2015, 8(10):1926-1942.

[13] LIN P, BI J, WOLFF S, et al. A west-east bridge based SDN inter-domain testbed[J]. Communications Magazine IEEE, 2015, 53(2):190-197.

[14] WANG Y, BI J, LIN P, et al. SDI: a multi-domain SDN mechanism for fine-grained inter-domain routing[J]. Annals of Telecommunications, 2016, 71(11-12):625-637.

[15] HOUIDI I, LOUATI W, AMEUR W B, et al. Virtual network provisioning across multiple substrate networks[J]. Computer Networks, 2011, 55(4):1011-1023.

[16] GROSSMAN P. Introduction to algorithms[M]. MCGRAW: MIT press, 2011.

[17] ZHANG Q, LIN M, YANG L T, et al. Energy-efficient scheduling for real-time systems based on deep Q-learning model[J]. IEEE Transactions on Sustainable Computing, 2017, 12(9):41-49.

[18] WATKINS C, DAYAN P. Q-learning[J]. Machine Learning, 1992, 8(3-4):279-292.

[19] WEI Q, LEWISF L, SUN Q, et al. Discrete-time deterministic Q-learning: a novel convergence analysis[J]. IEEE transactions on cybernetics, 2016: 1-14

[20] DIETRICH D, RIZK A, APADIMITRIOU P. Multi-domain virtual network embedding with limited information disclosure[C]//The 16th IFTP Networking Conference. 2013: 1-9.

[21] 贾伟, 夏靖波. 跨域虚拟网络映射问题研究[J]. 电子与信息学报, 2016, 38(3):728-734.
JIA W, XIA J B. Research on virtual network embedding across multiple domains[J]. Journal of Electronics & Information Technology, 2016, 38(3):728-734.

[22] WANG X, LOGUINOY D. Understanding and modeling the Internet topology: economics and evolution perspective[J]. IEEE/ACM Transactions on Networking, 2010, 18(1):257-270.

[23] 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.

[24] CHOWDHURY M, SAMUEL F, BOUTABA R. PolyViNE: policy-based virtual network embedding acrossmultiple domains[C]//The 2nd ACM SIGCOMMVISA. 2010: 49-56.

[作者简介]

张红旗（1962–），男，河北唐山人，博士，中国人民解放军战略支援部队信息工程大学教授、博士生导师，主要研究方向为网络与信息安全。

黄睿（1993–），女，新疆乌鲁木齐人，中国人民解放军战略支援部队信息工程大学硕士生，主要研究方向为软件定义网络、网络功能虚拟化、网络安全。

杨英杰（1971–），男，河南郑州人，博士，中国人民解放军战略支援部队信息工程大学教授、硕士生导师，主要研究方向为网络与信息安全。

常德显（1977–），男，河南南阳人，博士，中国人民解放军战略支援部队信息工程大学副教授、讲师，主要研究方向为网络与信息安全。

张连成（1982–），男，河南商丘人，博士，中国人民解放军战略支援部队信息工程大学副教授、讲师，主要研究方向为软件定义网络。