

# 基于软件定义网络的边缘控制部署机制\*

路 亚

(重庆电子工程职业学院 人工智能与大数据学院, 重庆 401331)

**摘要:** 针对软件定义网络 (software-defined networking, SDN) 中单一控制器容易发生过载导致较长时延的问题, 提出一种基于 SDN 的边缘控制模型, 该模型采取分层部署方式将边缘计算集成到 SDN 中, 每个边缘控制器控制其覆盖范围内部署的所有子边缘控制器和交换机, 负责区域内网络设备的通信量。为了方便管理边缘控制器之间的交互, 该模型引入一个控制器代理模块, 将设备请求转发给父控制器或将路由信息发送给子控制器来协调控制器之间的工作。实验结果表明, 与基于 SDN 的传统网络相比, 该方法依托部署在网络设备边缘的计算和存储服务, 减轻了 SDN 主控制器上的负载, 降低了转发平面和控制平面之间的延时, 显著地改善了总处理延时和带宽使用情况。

**关键词:** 边缘计算; 软件定义网络; 延时; 控制器

**中图分类号:** TP393      **文献标志码:** A      **文章编号:** 1001-3695(2020)12-052-3773-04

doi: 10.19734/j.issn.1001-3695.2019.07.0307

## Edge-based deployment mechanism based on software defined network

Lu Ya

(Artificial Intelligence & Big Data College, Chongqing College of Electronic Engineering, Chongqing 401331, China)

**Abstract:** Aiming at the problem that single controller in SDN is prone to overload and lead to long delay, this paper proposed an edge control model based on SDN, which integrated edge computing into SDN-based network by layered deployment. Each edge controller charged all of the sub-edge controllers and switches deployed and traffic from network devices within its coverage area. In order to manage the interaction between edge controllers conveniently, the proposed model introduced a controller agent module that forwarded device requests to the parent controller or sent routing information to the child to coordinate the work between the controllers. The experimental results show that compared with the traditional SDN-based network, the proposed method alleviates the load on the main SDN controller, minimizes the delay between the forwarding plane and the control plane and improves significantly total processing latency and bandwidth usage, which relies on the computing and storage services deployed on the edge of the network device.

**Key words:** edge computing; software defined network; delay; controller

## 0 引言

随着计算机网络的不断发展, 软件定义网络<sup>[1]</sup>在现代网络设计中发挥了重要作用, 越来越多地应用于校园网络、企业网络、光网络等场景。SDN 将网络的控制平面与数据转发平面进行分离, 控制平面是由独立主机构成的控制器来运行, 可以控制整个网络的交换机, 是整个 SDN 体系的核心平面, 其时延、带宽和吞吐量决定了 SDN 的性能。SDN 具有灵活的软件编程能力, 能够满足网络的资源规模扩展、自动化管理和灵活组网的要求<sup>[2]</sup>。但是, SDN 的设计给集中式控制器带来了很大的压力, 这可能会影响整个网络的性能, 并随着数据量的增加而导致较长的延时。

交换机等网络设备是数据通信中起重要作用的网络组件, 因此 SDN 在设计时考虑了交换机与控制器之间的通信问题, 并为其提供灵活的路由管理, 但是整个 SDN 的性能表现主要取决于控制器处理大容量通信的能力。SDN 将网络设备从控制应用程序中解放出来的理念导致了集中式单控制器的高需求问题。远程控制器接收数据的增多, 要求作出响应的的时间变短, 延时则相应增加。控制器具有有限的处理能力和较慢的

响应时间, 降低了整个网络性能。另一个延时源来自需要由控制器处理的大量数据, 网络链接能够感知到大量数据跨网络传输产生的延时。这部分延时是由带宽大小决定的。因此, SDN 的集中处理方法需要采取不同的技术来解决这些局限性<sup>[3]</sup>。对于单一控制器平面, 可以采取多线程技术降低控制平面的时延问题<sup>[4]</sup>, 但是随着数据平面交换机数量和终端数量的增加, 单一控制器的性能已经不足以应对网络规模的扩大。因此, 研究人员提出了增加控制器数量的分布式控制平面部署<sup>[5]</sup>, 通过多个控制器的协同工作, 可以大幅提升控制平面的数据处理能力, 同时降低控制平面的时延, 但是分布式控制平面部署也存在控制器间负载不均衡问题<sup>[6]</sup>。该问题可以从两个方面进行解决: 迁移交换机<sup>[7]</sup>和改变控制器部署位置及配置方式<sup>[8]</sup>。文献[7]给出了利用迁移交换机动态调整网络负载的机制, 但是在迁移过程中容易导致目标控制器的僵化, 反复进行迁移也对网络性能产生影响。文献[8]提出一种基于多参数节点排序的控制器位置部署策略, 通过对节点进行排序分层和计算节点间的控制权值来确定控制器位置。但是, 这种依靠提高控制器数量的方法极大提高了网络的维护成本。文献[9]采用多层次  $k$  路划分方法实现 SDN 区域划分, 降低了 SDN 跨域流数

**收稿日期:** 2019-07-25; **修回日期:** 2019-09-06      **基金项目:** 重庆市重点产业共性关键技术创新专项重点研发项目 (cstc2017zdey-zdyfx0017); 重庆市教委科学技术研究项目 (自然科学类) (KJ132207)

**作者简介:** 路亚 (1981-), 男, 河北永年人, 副教授, 硕士, 主要研究方向为云计算、网络与信息安全技术 (Luyacq160@126.com)。

量,提高了流表构建效率。但全局调整的实施对网络稳定性产生了一定影响,可能会导致各项应用不能正常开展。

针对上述问题,本文提出了一种基于边缘计算<sup>[10]</sup>的控制器负载均衡机制,通过提高边缘控制器和交换机之间的紧密性,减少总通信量拥塞和边缘控制器与交换机间的延时,降低通信链路的带宽占用量,从而保证非常短的响应时间。本文的创新点主要是利用边缘控制将路由计算和信息迁移到网络设备附近,通过分布在网络上的多个边缘控制节点降低中央控制器的计算压力,平衡整体通信量负载,而且由于边缘控制器运行代价较小,不会对网络稳定性产生影响。

### 1 软件定义网络和边缘计算

#### 1.1 软件定义网络

SDN 是由 Emulex 提出的一种创新型网络架构,其目的主要是对网络的有限资源进行最优利用,并且通过将控制平面与数据转发平面相分离来实现灵活的网络管理。SDN 控制架构的发展大致可分为四个阶段:4D 架构、Ethane 架构、OpenFlow 架构以及可扩展性架构。图 1 给出了基于 OpenFlow 的典型 SDN 体系结构。

该结构主要包括三层:底层是基础设施层,该层的主要功能是根据控制层提供的指令对网络数据进行转发;中间层是控制层,由于底层设备复杂的控制逻辑被抽象出来集中在控制层上,所以它具有可编程控制应用程序,主要有拓扑发现、设备管理、路由计算、防火墙等功能模块,为网络设备提供一组有关通信量管理的指令;顶层是应用层或应用平面,它包含一组业务应用程序,可以为网络提供额外的服务,如负载均衡、安全性和监控应用程序。

通常,SDN 架构中存在三种不同类型的组件,即远程中央控制器、网络设备(如交换机)和通信协议(如 OpenFlow 协议<sup>[10]</sup>)。其中,远程中央控制器是网络的核心组件,主要任务是根据预先编程的控制应用来进行网络通信量管理;网络设备部署在基础设施层中,这些设备是 SDN 体系结构的关键组成部分,它们的主要任务是根据 SDN 控制器提供的一些路由指令进行交互用于转发网络通信量;通信协议是将网络设备连接到远程中央控制器,当网络设备接收到新的通信量时,它们使用 OpenFlow 通信通道向远程控制器发送所需操作的请求。

#### 1.2 边缘计算

边缘计算是指在网络边缘执行计算的一种新型计算模型,其操作对象包括来自于云服务的下行数据和万物互连服务的上行数据。由于具有分布式和低延时计算、超越终端设备的资源限制、降低网络通信量压力等特点,边缘计算的研究得以迅速发展。与云计算模型不同,边缘计算是一个虚拟化框架,在终端设备和主要云数据中心之间提供计算、存储和其他服务,如图 2 所示。边缘计算中终端设备与云计算中心的请求与响应是双向的,终端设备不再是单方向云计算中心发出请求,还能够完成云计算中心下发的计算任务。

### 2 SDN 的边缘控制模型

#### 2.1 边缘控制模型

为了增强边缘计算能力,将云系统作为 SDN 主控制器,然后添加包含一组虚拟资源的控制边缘,这些虚拟资源可被视为

主控制器的代理。边缘控制主要是控制在网络设备附近允许的应用程序和路由信息,其基本思想是通过额外的计算和存储能力来增强网络边缘。这种设计有助于减轻主控制器的带宽负担和 OpenFlow 通信中的延时。为了达到更小的延时效果,可以将整个控制进程拆分为较小的进程,在不同区域内执行。同时,这种部署可以保证每个网络设备都可以以低延时的方式访问控制器控制范围内的任何地方,而且不会增加整个网络的通信量。

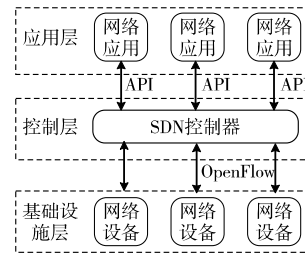


图1 基于OpenFlow的SDN体系结构  
Fig.1 SDN architecture based on OpenFlow

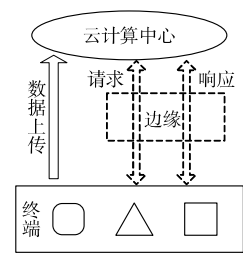


图2 边缘计算的双向计算流模型  
Fig.2 Bidirectional computational flow model for edge computing

边缘控制采取分层方式部署,每个控制器只需要掌握与其区域相关的路由信息即可。图 3 给出了 SDN 分级边缘控制区域示意图。边缘控制器的位置是灵活的:每个边缘控制器可以通过高速链路直接连接到其区域覆盖内的一个或多个交换机上,高效地处理局部数据流。此外,每个边缘控制器负责管理其指定区域内的通信量,并控制在其覆盖范围内部署的任何其他边缘控制器(子控制器),而所有边缘控制器都由主 OpenFlow 控制器控制,并且它能够访问其中的任何一个。

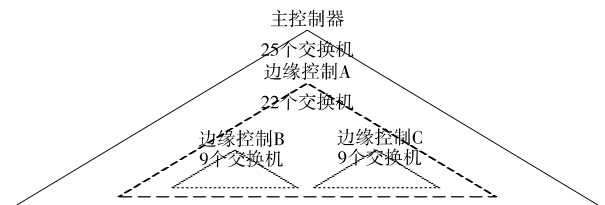


图3 分级边缘控制示意图  
Fig.3 Hierarchical edge control schematic diagram

转发平面由一系列相互连接的网络设备表示,具有丰富的交互场景:交换机到交换机、交换机到控制器以及其他网络设备之间的交互。本文方法的层次结构允许控制器和交换机之间进行更多的交互和响应。每个边缘控制器都部署在 SDN 的边缘区域,为转发平面的部分设备提供近距离的服务,并为该边缘提供单独的计算能力。这种分离设计可以平衡数据通信量,避免拥塞,减少控制平面和转发平面之间的传输延时。边缘控制可以在网络边缘执行通信量处理、数据存储和分发服务等丰富的实时交互服务,从而为主控制器的卸载提供一个理想平台。部分卸载过程允许每个控制器执行较少的计算量,但是父控制器需要较高的存储成本,以便存储来自于子控制器的信息。边缘的数量由网络的大小和数据量决定。每个边缘控制器都有一个独立的计算和存储服务,可以根据其位置的便利性部署在边缘区域。

#### 2.2 架构

为了将边缘控制完全集成到 SDN 体系结构中,引入一个控制器代理模块,用来管理边缘控制器之间的交互。如图 4 所示,控制器代理模块通过将设备请求转发给父控制器或将路由信息发送给子控制器来管理与其他控制器的交互。除此之外,控制器代理还可以利用之前的主控程序处理网络的转发平面。

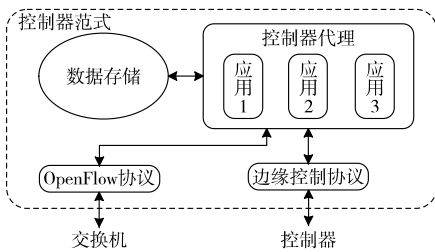


图4 边缘控制器范式示意图  
Fig.4 Schematic diagram of edge controller paradigm

### 2.3 总体控制方案

边缘控制可以在低延时下为一定数量的交换机提供服务的一个重要因素是其位置的分布。图 5 显示了边缘控制器的位置和覆盖范围。边缘存储包含有关其指定区域的数据,并且可以在其区域内独立执行路由计算。边缘控制器的任务主要包括三点:

- a) 处理网络基础设施中的通信量以及与应用程序层协调附加网络服务。
- b) 与父控制器进行交互,处理来自不同区域目标的请求。
- c) 与子控制器交互,响应控制器请求。

该模型请求管理工作流程如图 6 所示。对于到达的数据包流,边缘控制器负责确定要转发该流的路由路径。每个控制器只能在其覆盖范围内找到通向目的地的路由。如果数据包流目的地不在此边缘控制所属的同一区域内,边缘控制器会将此请求转发给其父控制器,然后由父控制器根据路由信息进行指令响应。这一过程可以减少主控制器的处理工作量,同时在边缘控制器内完成控制任务。

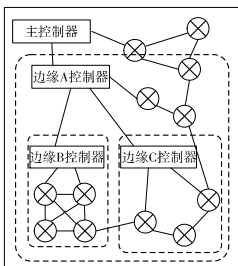


图5 整体边缘控制的位置和覆盖范围  
Fig.5 Position and coverage of overall edge control

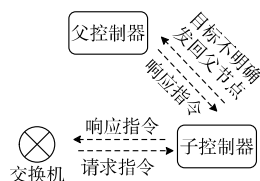


图6 总体控制流程示意图  
Fig.6 Schematic diagram of overall control flow

### 3 实验结果与分析

为了测试本文方法的性能,选用配置为 Intel Core i7 @ 3.2 GHz 和 8 GB RAM 的机器进行测试。为了对传统的 SDN 控制模型和提出的控制模型的性能、控制器负载情况以及对网络设备发送一组请求后采取响应的总处理延时进行评估,本文在 OpenDaylight 仿真器中模拟基于 SDN 的传统网络的性能,使用轻网络仿真器 MiniNet 中的 Python 测试模型,测试基于边缘控制 SDN 的性能,显示应用边缘控制对控制平面性能的影响。

#### 3.1 实验设置

实验采用两种网络:一种是基于 SDN 的传统网络,该网络包含 1 个 SDN 控制器和 25 个互连交换机;另一种是所提出的基于边缘控制的 SDN。该网络拓扑结构如图 3 所示,包括 1 个 SDN 主控制器,3 个边缘控制器和 25 个互连交换机。实验中,将 25 个互连交换机中的一些分配给 3 个边缘控制器(边缘 A、B 和 C):边缘控制器 B 和 C 分别配有 9 个交换机,边缘控制器 A 是 B 和 C 的上级,分配了另外 4 个交换机。主控制器负责处

理来自边缘控制器 A 和剩余 3 个未分配交换机的通信量。实验测试主要对 10 000 个随机产生的新数据流进行处理。

在测试过程中,假设所有到达的通信量都是新的,并且在任何流表中都没有匹配的条目,则意味着控制器必须在没有缓冲延时的情况下对所有到达的流量作 100% 处理。其他假设如下: a) 每个控制器的处理时间是 245  $\mu$ s; b) 交换机服务延时为 10.25  $\mu$ s; c) 考虑到传播延时通常受到邻近效应的影响,随机选择包长度、带宽大小、物理链路长度和传播速度。

上述参量值是基于先前的研究<sup>[11]</sup>进行设定的,这些研究测试了 OpenFlow 交换机在有/无控制器交互的情况下传输数据包所需的处理时间。

#### 3.2 实验结果与分析

表 1 给出了基于 SDN 的传统网络与所提出的边缘控制模型在处理相同工作时的负载比较,以及对于 25 个网络设备的请求、每个控制器处理时的负载大小。从表中可以看出,在正常情况下传统的 SDN 控制器必须 100% 地处理整个负载。而提出的边缘控制模型的实验结果表明,主控制器和三种不同的边缘控制器之间的负载得到了有效的分散:主控制器的负载降低至 23%,边缘控制器 A、B 和 C 负载分别是 65%、37% 和 29%。从表 1 中还可以看出,边缘控制器 B、C 路由请求的 63% 和 70% 的目标位置都位于各自区域之外,需要边缘控制器 A 参与处理。除此之外,边缘控制器 A 存在 4% 的路由请求的目标位置在 A 区之外,需要转发到主控制器。

表 1 控制器负载实验结果  
Tab. 1 Experimental results of controller load

控制器	覆盖量	总负载量/1%	转发请求/1%
传统 SDN 控制器	25 个交换机	100	0
AAMcon 分布式控制器	25 个交换机,4 个控制器	75	0
本文模型主控制器	3 个交换机和边缘 A	23	0
边缘 A	4 个交换机和边缘 B、C	65	4
边缘 B	9 个交换机	37	63
边缘 C	9 个交换机	29	70

图 7 给出了不同网络控制器处理相同请求数量时花费的总处理时间。从图中可以看出,传统 SDN 控制器处理 1 000 个请求的总耗时为 252 ms,10 000 个请求的总耗时增加至 2 415 ms;文献 [12] 中 AAMcon 分布式控制器处理 1 000 个请求的平均总耗时是 174 ms,10 000 个请求总耗时为 1 571 ms;而本文方法的边缘控制器处理 1 000 个请求的平均总耗时是 95 ms,10 000 个请求总耗时增加到 923 ms,相应的主控制器的总处理时间已减少到 54 ms(1 000 个请求)、551 ms(10 000 个请求)。

图 8 阐述了不同网络控制器处理相同请求数量时整个带宽的使用情况。图中曲线清楚地表明了单个链接中传输请求总量的平均值。测试结果表明,基于边缘控制的网络带宽性能明显优于其他方法,相对于传统网络使用率平均降低了 45%。

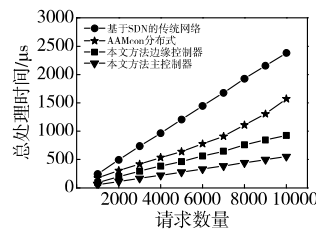


图7 控制器的总处理时间  
Fig.7 Total processing time of the controller

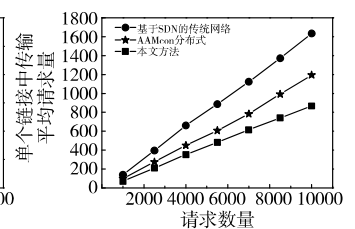


图8 单个链接中传输的平均请求量  
Fig.8 Average number of requests transmitted in a single link

通过实验测试证实了将边缘控制集成到基于 SDN 中的有效性,可以减少主控制器上的负载,极大改善了路由请求的处

理时间。同时,结果还表明边缘控制可以以较低的延时、最小的带宽和极少的跳转保证网络设备达到实时响应。

#### 4 结束语

针对软件定义网络中单一控制器容易发生过载导致较长时延的问题,本文提出了一种基于 SDN 的边缘控制模型。该模型是基于 SDN 控制器与网络设备之间的物理邻近性可以提高网络性能的思想,将边缘计算与 SDN 相结合的边缘控制模型,通过将计算和存储服务置于网络设备附近,减轻了 SDN 主控制器的负载,降低了交换机和控制器之间的总延时。此外,边缘控制有助于平衡数据流量,避免 SDN 中的拥塞,减少控制平面和转发平面之间的传输延时。仿真结果表明,本文模型可以在保持较低延时的同时显著地管理较高的网络负载。

#### 参考文献:

- [1] Li Yang, Cai Zhiping, Xu Hong. LLMP: exploiting LLDP for latency measurement in software-defined data center networks [J]. *Journal of Computer Science & Technology*, 2018, 33(2): 277-285.
- [2] 陈超,曹晓梅. SDN 场景中基于双向流量特征的 DDoS 攻击检测方法 [J]. *计算机应用研究*, 2019, 36(7): 2148-2153. (Chen Chao, Cao Xiaomei. Distributed denial of service attack detection based on bidirectional traffic feature in software defined network [J]. *Application Research of Computers*, 2019, 36(7): 2148-2153.)
- [3] Kobo H I, Abu-Mahfouz A M, Hancke G P. A survey on software-defined wireless sensor networks: challenges and design requirements [J]. *IEEE Access*, 2017, 5: 1872-1899.
- [4] Karakus M, Durrresi A. A survey: control plane scalability issues and approaches in software-defined networking (SDN) [J]. *Computer Networks*, 2017, 112(1): 279-293.
- [5] 孙涛,张俊星. SDN 性能优化技术研究综述 [J]. *计算机科学*, 2018, 45(S2): 94-101. (Sun Tao, Zhang Junxing. Review of SDN performance optimization technology [J]. *Computer Science*, 2018, 45(S2): 94-101.)
- [6] 赵国锋,黎军,王新恒,等. 基于 SDN 的 WLAN 负载感知切换方案实现 [J]. *计算机应用研究*, 2019, 36(5): 1454-1458. (Zhao Guofeng, Li Jun, Wang Xinheng, et al. Implementation of WLAN load aware handover method based on SDN [J]. *Application Research of Computers*, 2019, 36(5): 1454-1458.)
- [7] Hu Tao, Yi Peng, Zhang Jianhui, et al. A distributed decision mechanism for controller load balancing based on switch migration in SDN [J]. *China Communications*, 2018, 15(10): 129-142.
- [8] 张岩,黄韬,卢波,等. 基于多参数节点排序的 SDN 控制器部署策略 [J]. *北京邮电大学学报*, 2016, 39(4): 30-34. (Zhang Yan, Huang Tao, Lu Bo, et al. Controller deployment scheme based on multi-factor node ranking in SDN [J]. *Journal of Beijing University of Posts and Telecommunications*, 2016, 39(4): 30-34.)
- [9] 王潇,杨金民. 一种基于 IBPSO 的 SDN 控制器放置优化方案 [J]. *计算机应用研究*, 2019, 36(10): 3069-3071. (Wang Xiao, Yang Jinmin. Optimization scheme for SDN controllers' placement based on IBPSO [J]. *Application Research of Computers*, 2019, 36(10): 3069-3071.)
- [10] Zhang Qingyi, Wang Xingwei, Huang Min, et al. Software defined networking meets information centric networking: a survey [J]. *IEEE Access*, 2018, 6: 39547-39563.
- [11] Buyya R, Son J. Software-defined multi-cloud computing: a vision, architectural elements, and future directions [C] // Proc of International Conference on Computational Science and Its Applications, the 18th IEEE International Conference on Melbourne. Piscataway, NJ: IEEE Press, 2018: 3-18.
- [12] Liu Waixi, Wang Yu, Zhang Jie, et al. AAMcon: an adaptively distributed SDN controller in data center networks [J]. *Frontiers of Computer Science*, 2019, 6(1): 1-16.
- [13] Ning Zhaolong, Wang Xiaojie, Huang Jun. Mobile edge computing-enabled 5G vehicular networks: toward the integration of communication and computing [J]. *IEEE Vehicular Technology Magazine*, 2019, 14(1): 54-61.
- [14] 柴蓉,胡恂,李海鹏,等. 基于 SDN 的 5G 移动通信网络架构 [J]. *重庆邮电大学学报: 自然科学版*, 2015, 27(5): 569-576. (Chai Rong, Hu Xun, Li HaiPeng, et al. SDN-based 5G mobile communication network architecture [J]. *Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition*, 2015, 27(5): 569-576.)
- [8] Ma Lu, Wen Xiangming, Wang Luhan, et al. An SDN/NFV based framework for management and deployment of service based 5G core network [J]. *China Communications*, 2018, 15(10): 86-98.
- [9] Farrel A. Recent developments in service function chaining(SFC) and network slicing in backhaul and metro networks in support of 5G [C] // Proc of the 20th International Conference on Transparent Optical Networks. Piscataway, NJ: IEEE Press, 2018: 1-4.
- [10] Tran T X, Hajisami A, Pandey P, et al. Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges [J]. *IEEE Communications Magazine*, 2017, 55(4): 54-61.
- [11] 姚睿. 基于机器学习的 SDN 网络流量预测与负载均衡 [D]. 成都: 电子科技大学, 2018. (Yao Qian. Machine learning based traffic prediction and load-balancing for software defined network [D]. Chengdu: University of Electronic Science and Technology of China, 2018.)
- [12] Oljira D B, Grinnemo K J, Taheri J, et al. A model for QoS-aware VNF placement and provisioning [C] // Proc of IEEE Conference on Network Function Virtualization and Software Defined Networks. Piscataway, NJ: IEEE Press, 2017: 1-7.
- [13] Miyazawa T, Jibiki M, Kafle V P, et al. Autonomic resource arbitration and service-continuable network function migration along service function chains [C] // Proc of NOMS -IEEE/IFIP Network Operations and Management Symposium. Piscataway, NJ: IEEE Press, 2018: 1-9.
- [14] 王琛,汤红波,游伟,等. 一种 5G 网络低时延资源调度算法 [J]. *西安交通大学学报*, 2018, 52(4): 117-124. (Wang Chen, Tang Hongbo, You Wei, et al. A resource scheduling algorithm with low latency for 5G networks based on effective hybrid genetic algorithm and tabu search [J]. *Journal of Xi'an Jiaotong University*, 2018, 52(4): 117-124.)
- [15] Lee J, Ko H, Suh D, et al. Overload and failure management in service function chaining [C] // Proc of IEEE Conference on Network Software-ization. Piscataway, NJ: IEEE Press, 2017: 1-5.
- [16] Mebarkia K, Zsóka Z. Service traffic engineering: avoiding link overloads in service chains [J]. *Journal of Communications and Networks*, 2019, 21(1): 69-80.
- [17] Lei T H, Hsu Y T, Wang I C, et al. Deploying QoS-assured service function chains with stochastic prediction models on VNF latency [C] // Proc of IEEE Conference on Network Function Virtualization and Software Defined Networks. Piscataway, NJ: IEEE Press, 2017: 1-6.
- [18] Csoma A, Sonkoly B, Csikor L, et al. ESCAPE: extensible service chain prototyping environment using MiniNet, click, NETCONF and POX [J]. *ACM SIGCOMM Computer Communication Review*, 2014, 44(4): 125-126.
- [19] 张若晨. 基于 OpenvSwitch 的代理虚拟交换机在 SDN 网络中的实现与应用 [D]. 广州: 华南理工大学, 2016. (Zhang Ruochen. Implementation and application of OpenvSwitch-based virtual switch in SDN network [D]. Guangzhou: South China University of Technology, 2016.)

(上接第 3752 页)