



专题: SDN/NFV

## 面向 SDN 网络虚拟化平台的控制器放置算法

董晓东<sup>1,2</sup>, 郭志强<sup>1,2</sup>, 陈胜<sup>3</sup>, 周晓波<sup>1,2</sup>, 齐恒<sup>3</sup>, 李克秋<sup>1,2</sup>

(1. 天津大学计算机学院, 天津 300350;

2. 天津市先进网络技术与应用重点实验室, 天津 300350;

3. 大连理工大学计算机学院, 辽宁 大连 116024)

**摘要:** 随着网络技术发展, 以网络虚拟化为手段解决 TCP/IP 网络体系结构僵化问题已成为未来网络领域发展的主流方向之一。SDN (software defined networking, 软件定义网络) 作为一种新兴的网络体系结构, 为网络虚拟化提供了有效的解决方案。首先总结了当前具有代表性的 SDN 网络虚拟化平台, 并对比了 SDN 与传统网络环境中部署虚拟网的区别, 然后针对 SDN 网络虚拟化平台中的虚拟网络映射问题, 提出一种时延敏感的虚拟化控制器放置算法, 最后通过实验验证了该算法在提高网络资源的利用效率的同时, 保证了控制器与底层交换机的通信时延在可接受范围之内。

**关键词:** 软件定义网络; 网络虚拟化; 虚拟网络映射; 时延敏感

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2017091

## SDN hypervisor oriented vSDN controller placement algorithm

DONG Xiaodong<sup>1,2</sup>, GUO Zhiqiang<sup>1,2</sup>, CHEN Sheng<sup>3</sup>, ZHOU Xiaobo<sup>1,2</sup>, QI Heng<sup>3</sup>, LI Keqiu<sup>1,2</sup>

1. School of Computer Science and Technology, Tianjin University, Tianjin 300350, China

2. Tianjin Key Laboratory of Advanced Networking (TANK), Tianjin 300350, China

3. School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

**Abstract:** With the development of the future internet technology, the future network architecture which utilizes network virtualization as a tool to solve the rigidity of traditional network becomes the main stream. As a new network architecture, software defined networking provides an effective network virtualization solution. Firstly, related technology of network virtualization in SDN was concluded, distinguished differences between deploying virtual network in SDN and traditional network was compared, and aiming at the virtual network mapping problem in SDN virtualization platform, a latency sensitive SDN virtualization controller placement algorithm. At last, experiment results show that the algorithm not only can assure the delay between SDN controllers and switches in an acceptable range, but also improve the utilization efficiency of network resources.

**Key words:** software defined networking, network virtualization, virtual network embedding, latency sensitive

收稿日期: 2017-01-13; 修回日期: 2017-03-28

基金项目: 国家重点研发计划基金资助项目 (No.2016YFB1000205); 国家自然科学基金重点基金资助项目 (No.61432002); 国家自然科学基金资助项目 (No.61370199, No.61672379); 大连市高级人才创新工程项目 (No.2015R049)

**Foundation Items:** National Key Research and Development Program of China (No.2016YFB1000205), The State Key Program of National Nature Science of China(No.61432002), The National Natural Science Foundation of China(No.61370199, No.61672379), The Dalian High-Level Talent Innovation Program(No.2015R049)

## 1 引言

当前互联网的发展离不开TCP/IP网络体系结构的支撑。在该体系结构中，TCP/IP起到了承上启下的核心作用。对上层协议而言，TCP/IP屏蔽了数据链路层和物理层协议差异，使上层协议专注于应用的实现；对下层协议而言，又屏蔽了应用层协议，使下层协议只需要专注于基于IP地址的数据转发。但是，随着云计算、大数据及物联网时代的到来，人们对网络应用多样化及网络服务的需求不断提升，网络中的数据流量呈爆炸式增长，直接导致了网络服务质量的下降。运营商为了保证网络服务质量，必须被动地不断进行网络扩容，导致运营维护成本成倍增长。种种迹象表明，以TCP/IP为核心的网络体系结构逐渐成为互联网产业发展的瓶颈。

为了解决TCP/IP网络体系结构中的问题，业界广泛开展了大量关于未来网络体系结构的研究工作。其中一个关键的问题是如何根据服务的具体需求在共享的物理资源上定制与其他服务隔离的专用网络，由此诞生网络虚拟化的概念。网络虚拟化旨在一个共享的物理网络资源之上创建多个VN（virtual network，虚拟网络），同时每个VN相互隔离并可以独立地部署及管理。网络虚拟化的目的是提高网络资源的使用效率，简化网络资源的管理以及实现网络定制化，其本质上是一种资源共享技术。以网络虚拟化为手段解决TCP/IP网络体系结构僵化问题，已成为未来网络体系结构发展的主流方向，欧盟FP7计划中的4WARD项目<sup>[1]</sup>、FEDERICA项目<sup>[2]</sup>、FELICA项目<sup>[3]</sup>、AGAVE项目<sup>[4]</sup>以及美国FIA计划中的ChoiceNet项目<sup>[5]</sup>都对此开展了大量的研究。在AT&T公司的Domain 2.0白皮书<sup>[6]</sup>以及5G<sup>[7]</sup>中，利用网络虚拟化技术实现面向服务的网络定制成为了各大运营商改进、升级产业的重要手段。随着SDN（software defined networking，软件定

义网络）、网络虚拟化技术以及NFV（network function virtualization，网络功能虚拟化）的结合日益紧密<sup>[8]</sup>，网络虚拟化技术已经成为支持未来网络部署、管理以及服务分离的关键技术。

SDN为网络虚拟化的实现提供了有效支撑。该技术实现了控制层面与数据层面的解耦分离，将对网络的控制及管理抽象到逻辑或者虚拟的实体，提供灵活、开放的控制接口，使网络交换设备只负责转发，有助于网络虚拟化的实现。美国的GENI（Global Environment for Network Innovations）项目<sup>[9]</sup>、Internet2组织<sup>[10]</sup>以及欧盟的FIRE（Future Internet Research and Experiment）项目<sup>[11]</sup>都致力于搭建基于SDN的虚拟化网络实验床。当前较为成熟的SDN网络虚拟化平台包括FlowVisor<sup>[12]</sup>、OpenVirteX<sup>[13]</sup>以及FlowN<sup>[14]</sup>等。

在SDN网络虚拟化平台中，所产生的切片网络实例被称为vSDN（virtual SDN，虚拟SDN），它继承了VN和SDN各自的优点，将灵活的网络管控与面向服务的自定义网络相结合。与传统网络虚拟化平台相比，SDN网络虚拟化平台技术呈现一些新的特征。首先，SDN控制层利用数据平面接口可以对网络底层的物理设备资源进行抽象，每个vSDN切片都需要一个控制器实例通过南向接口对其进行管理，这种控制层面和数据层面分离的思想简化了虚拟网络的构建和对网络资源的管理。其次，由于控制器需要不断与网络设备进行通信，为了保证虚拟网络正常运转，vSDN的控制器与底层交换机之间的通信通道需要避免流量拥塞，同时尽可能降低通信时延。在实际的应用场景中，由于控制器往往采用带内接入的方式与交换机建立控制链路，网络环境复杂。大多数SDN网络虚拟化平台需要在配置vSDN时由管理员手动指定控制器位置。这种手动指定的方式难以在动态、大规模的网络环境中应用。

本文针对SDN网络虚拟化平台中的控制器位置选择问题，提出了一种时延敏感的vSDN控制



器放置算法。其基本思想是：从备选位置中选出 vSDN 控制器的放置位置，在保证 vSDN 控制器与底层交换机的通信时延在可接受的范围内的同时，提高网络资源的利用效率。

## 2 研究背景与相关工作

首先介绍传统网络中常用的虚拟化技术以及其原理，然后概括了 SDN 网络虚拟化的实现原理并以 3 种业界最关注的 SDN 网络虚拟化平台为例进行了详细分析，最后描述了 SDN 环境下虚拟网络映射的研究现状。

### 2.1 传统网络中的虚拟化技术

#### (1) VLAN

VLAN (virtual local area network, 虚拟局域网) [15]可以在逻辑上将一组物理设备组织起来形成一个虚拟的网段，一个 VLAN 就是一个广播域。IEEE 于 1999 年颁布了用于标准化 VLAN 方案的 802.1Q 协议标准草案。VLAN 除了将网络划分为多个广播域，有效地控制广播风暴的发生以外，通过 VLAN 可以灵活地将用户按照其业务进行分类和聚合，从而达到简化网络管理并降低网络运维成本的目的。此外，还可以通过 VLAN 将含有敏感数据的用户组与其他网络隔离，进而增强局域网的安全性。目前，定义 VLAN 的方法很多，主要包括：基于端口定义的 VLAN、基于 MAC 地址定义的 VLAN、基于路由定义的 VLAN 以及基于策略定义的 VLAN。

#### (2) VPN

VPN (virtual private network, 虚拟专用网) [16]的功能是在公用网络上建立一个临时、安全的隧道链接，从而实现专用网络。VPN 常用于远程访问，其基本工作原理如图 1 所示，首先 B 发起对 A 的请求，B 的 VPN 网关接收到 B 发出的消息并进行检查；发现是对 A 的请求，对数据分组进行 VPN 技术封装之后发送给 A 所处的局域网的公网 IP 地址；A 的 VPN 网关接收到 B 发送来的数据

分组后对其进行解析，发现是 B 发送过来的 VPN 数据分组后将其转发给 A。根据 OSI 参考模型，VPN 技术可以根据其实现方法在模型中对应的位置进行分类：应用层的 SSL (secure socket layer, 安全套接字层)；会话层的 socket5；网络层的 IPsec (internet protocol security, internet 协议安全性)；数据链路层的 PPTP (point to point tunneling protocol, 点对点隧道协议)和 L2F (level 2 forwarding protocol, 二层转发协议)。

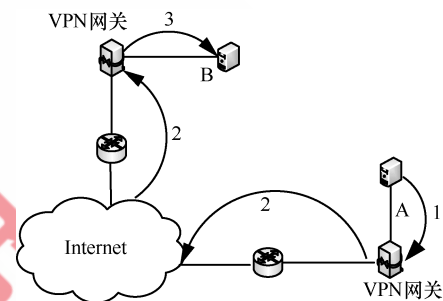


图 1 VPN 基本工作原理

### 2.2 SDN 网络虚拟化平台

传统 SDN 网络虚拟化平台原理如图 2 (a) 所示，网络应用程序 A 和 B 通过北向接口与控制器进行通信，进而通过南向接口与物理设备交互，实现相应的网络策略。逻辑上集中的 SDN 控制器拥有全局的网络视图，向上层应用提供丰富的 API 以实现“可编程”。因此，通过在控制层面和转发层面之间加入 SDN 网络虚拟化平台，可实现 SDN 网络虚拟化。

SDN 网络虚拟化平台的工作原理分别如图 2 (b) 和图 2 (c) 所示。在图 2 (b) 中，SDN 网络虚拟化控制平台直接与底层物理网络设备交互，以透明的方式转义以及转发控制数据；图 2(c)中 SDN 网络虚拟化控制平台通过屏蔽链路发现协议的方法造成每个 vSDN 控制器直接控制相应的虚拟网络拓扑的错觉[17]。不管在图 2 (b) 还是在图 2 (c) 中，虚拟网络都依托一个物理网络生成，但是任意不同的虚拟网络之间相互隔离、互不影响。控制层面的隔离、数据层面隔离以及地址空间隔离

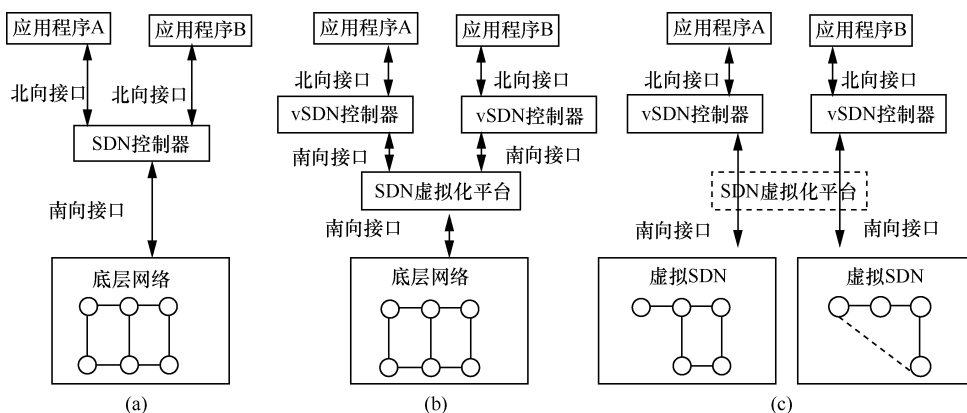


图2 SDN网络虚拟化平台原理

是网络虚拟化中最重要的技术。

本文以3种最受关注的虚拟化平台：FlowVisor、OpenVirteX以及FlowN为例，对其基本原理进行了介绍。

### (1) FlowVisor

FlowVisor是最早的开源SDN网络虚拟化平台，采用Java语言编写，南向接口采用OpenFlow协议。其最初的设计目的是实现实验网络和运营网络的流量隔离，并于2009年开始应用于美国斯坦福大学校园网，是目前应用案例最多的SDN网络虚拟化平台。FlowVisor的体系结构如图3所示，FlowVisor位于底层物理设备和上层控制器之间，以代理的方式透明地通过OpenFlow协议管理网络设备。在FlowVisor中，每一个vSDN都被称为一个“切片”，每个网络切片都由一个上层控制器负责管理。FlowVisor将一部分分组头字段空间抽取出来，称为流空间或地址空间。不同vSDN的地址空间相互隔离、互不重叠，vSDN控制器只负责控制属于其地址空间内的网络。当vSDN控制器向底层交换机下发命令时，控制器将命令发送给FlowVisor；然后，FlowVisor将命令交给与vSDN控制器对应的网络转义模块进行处理；分片器对命令的有效性进行验证，在修改后交给FlowVisor转发模块，再通过其下发给底层物理网络设备。在底层网络设备向控制器发起请求时也需要执行类似流程。由于FlowVisor

需要利用一部分分组头字段空间来区分不同的网络切片。因此FlowVisor无法利用全部的流空间地址。此外，FlowVisor也不支持vSDN拓扑自定义。

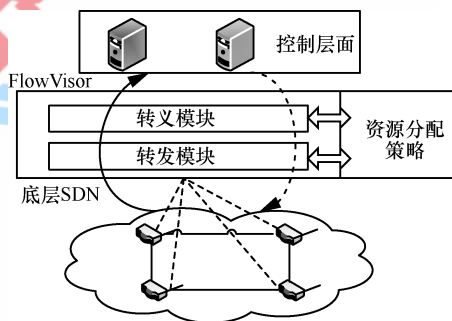


图3 FlowVisor体系结构

### (2) OpenVirteX

OpenVirteX是由ON.Lab（目前已经与ONF合并）开发的开源网络虚拟化平台，也采用Java语言编写，其系统结构如图4所示。为了解决FlowVisor不支持vSDN网络拓扑定制以及地址空间不能完全独立的问题，OpenVirteX采用了两种方法。在网络拓扑虚拟化方面，OpenVirteX采用了屏蔽或伪造NOS（network operating system，网络操作系统）发送的LLDP（link layer discovery protocol，链路层发现协议）消息应答的方法。举例来说，当需要构建虚拟存在的交换机时，OpenVirteX会伪造一个LLDP应答报文并发送给控制器，从而给控制器造成一个链接真实存在的错觉。在地址空间方面，OpenVirteX为不同切片

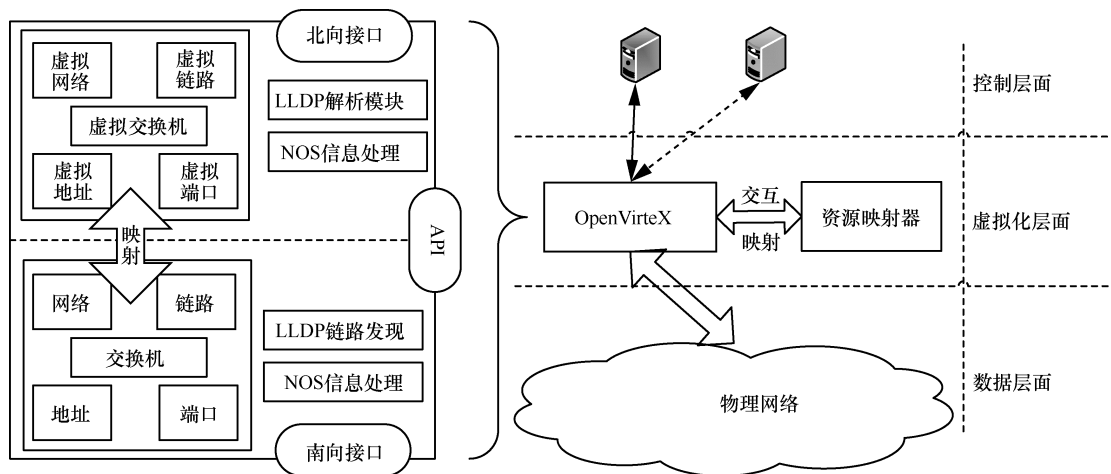


图 4 OpenVirteX 系统结构

分配一个独一无二的标识符，并为每一个终端分配一个由物理 MAC 地址和 IP 地址组合构成的标识符，在数据分组进入边界交换机时利用 IP 地址重写和 MAC 地址重写的方法重写数据分组头。但是，数据分组头重写增加了转发时延和边界交换机的负载。

### (3) FlowN

FlowN 是一款采用容器技术的 SDN 网络虚拟化平台，其目的是解决以下两个问题：减少为每一个 vSDN 分配一个控制器带来巨大的资源开销及网络切片映射数据保留在内存中带来的可扩展性问题。FlowN 本质上是一个改进的 NOX 控制器，不同的 NOX 应用程序运行在相互隔离的容器中，管理属于自身的虚拟网络拓扑，执行本地事件响应，采用相互独立的地址空间，其系统结构

如图 5 所示。当接收到数据层面发送来的请求时，FlowN 首先运行事件管理器来识别该请求所属的用户组，然后将其转发给对应的用户组的事件管理器。当用户事件管理器调用 NOX 方法时，FlowN 会先将其拦截，经过虚拟网络和物理网络的映射和转义之后再下发给底层物理交换机。与其他 SDN 网络虚拟化平台不同的是，FlowN 不需要为每一个虚拟网络切片分配一个独立的控制器实例。此外，由于采用了基于数据库技术的方法保存虚拟节点和物理节点之间的映射关系而不是保存在内存中，FlowN 可以应用于大规模网络。

除了上述 3 个 SDN 网络虚拟化平台，学术界中还有一些其他的 SDN 网络虚拟化平台，这些平台的对比见表 1。

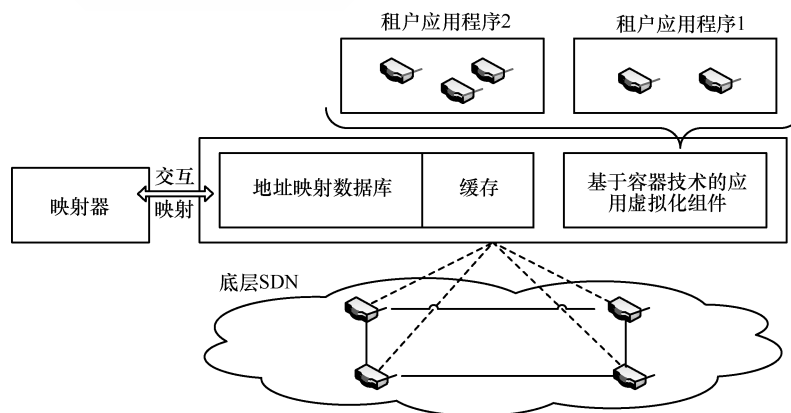


图 5 FlowN 系统结构

表 1 通用 SDN 虚拟化平台设计对比

名称	虚拟化方式	网络拓扑虚拟化	地址空间隔离
FlowVisor	基于虚拟化平台	不支持	不重叠的地址空间
AdVisor <sup>[18]</sup>	基于虚拟化平台	可以隐藏虚拟链路的中间节点	基于 VLAN、MPLS 的地址隔离
VeRTIGO <sup>[19]</sup>	基于虚拟化平台	可以隐藏虚拟链路的中间节点	基于 VLAN、MPLS 的地址隔离
enhanced FlowVisor <sup>[20]</sup>	基于虚拟化平台	不支持	不重叠的地址空间
Slices Isolator <sup>[21]</sup>	基于虚拟化平台	不支持	不重叠的地址空间
OpenVirteX	基于虚拟化平台	支持	支持可重叠的地址空间隔离
FlowN	基于容器技术的应用程序虚拟化	支持	基于 VLAN 的地址空间隔离
Network Hypervisor <sup>[22]</sup>	API 技术的应用程序虚拟化	——	不重叠的地址空间
OF NV Cloud <sup>[23]</sup>	基于虚拟化平台	不支持	基于虚拟 MAC 地址的地址空间隔离
AutoVFlow <sup>[24]</sup>	基于容器技术的应用程序虚拟化	支持	基于虚拟 MAC 地址的地址空间隔离
NVP <sup>[25]</sup>	基于虚拟化平台	——	——
AutoSlice <sup>[26]</sup>	基于虚拟化分布式平台	支持	基于 VLAN、MPLS 的地址隔离

### 2.3 SDN 中虚拟网络映射

在网络虚拟化环境中如何将虚拟网络嵌入底层基础网络设备是一个备受关注的问题，在学术界被称为 VNE (virtual network embedding, 虚拟网络嵌入) 问题。高效的嵌入方法不仅可以提高虚拟网络的性能，而且能够提高物理设备的资源利用效率。VNE 问题已经被证明是 NP 难问题，参考文献[27]总结了传统网络中 VNE 问题的解决方法。尽管所提及的方法某种程度上适用于 vSDN 的嵌入问题，但值得注意的是，与传统网络嵌入相比，vSDN 的嵌入存在一些差异。集中化的控制层面为实现 SDN 网络虚拟化提供了便利的手段，然而嵌入底层物理网络设备的每一个 vSDN 都需要一个控制器对其进行管理。在嵌入 vSDN 时，往往需要管理员手动指定 vSDN 控制器的物理位置。在实际应用中，控制器往往采用带内接入的方式接入网络。为了保证虚拟网络正常运行，在嵌入 vSDN 时必须确保所有的控制器—交换机通道不会出现拥塞和高时延的情况。参考文献[28]提出了可建立虚拟网络拓扑及嵌入控制器的算法。鉴于其离线算法的特性，无法在复杂的动态网络应用场景中部署。

本文针对 SDN 网络虚拟化中控制器位置选择问题提出了一种时延敏感的 vSDN 控制器放置在线算法。该算法优先考虑通信时延，从备选位置选择 vSDN 控制器的放置点，保证控制器与交换

机的通信时延控制在合理的范围内，从而既保证了 vSDN 的合理嵌入，又提高了网络资源的利用效率。

### 3 时延敏感的 vSDN 控制器放置算法

在实际应用中，尤其是云平台和数据中心网络中，控制器往往采用带内接入的方式接入网络，如图 6 所示。为了保证虚拟网络正常运行，在嵌入 vSDN 时必须确保所有的控制器—交换机通道不会出现拥塞和高时延的情况。本文提出的低时延的 vSDN 控制器放置在线算法，其基本思想是逐层搜索可以确定 vSDN 控制器实例的位置，然后将 vSDN 控制器实例映射到恰好满足时延条件的位置上。其具体流程如算法 1 所示。

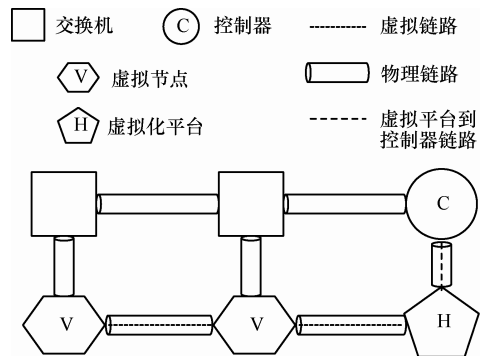


图 6 vSDN 控制器放置示例

#### 算法 1 时延敏感的 vSDN 控制器放置算法

输入：vSDN 请求

初始化：变量  $n$ 、集合  $C$



(1) for 距离 SDN 虚拟化平台距离为  $n$  跳的且与集合  $C$  中节点相连的节点, 计算节点到虚拟交换机的时延

(2) 判断每个节点与 vSDN 请求的最大时延是否满足 vSDN 时延需求

(3) if 不存在满足时延条件的节点

(4) then goto (19)

(5) else

(6) for 每个满足时延条件的节点

(7) 判断节点是否有足够的资源以容纳 vSDN 控制器实例

(8) if 不存在负载符合要求的

(9) then  $n+=step$

(10) goto (2)

(11) else

(12) for 每个满足负载要求的

(13) 将该节点加入  $C$  中

(14)  $n+=step$

(15) goto (2)

(16) end if

(17) end if

(18) if  $C$  为空

(19) then 停止查找, 放弃嵌入

(20) else

(21) 从集合  $C$  中找出时延最大的节点

作为嵌入 vSDN 控制器的目标节点

## 4 实验与分析

### 4.1 实验设置

为了真实地反映本文所提算法的有效性, 本文采用了模拟实验的方法对本文所提出的算法进行了验证。鉴于 OpenVirteX 开源、简单以及功能全面的特点, 本文采用 Mininet+OpenVirteX 的实验环境。实验采用了 CERNet 网络拓扑, 共包括 41 个核心网交换机和 258 条单项链路。为了简化实验, 假设每个控制器实例所占用的资源相同,

每个核心网交换机下可以容纳 4 个控制器实例。本文用 vSDN 请求所能承受的最大控制器—交换机时延表示 vSDN 时延需求。为了不失一般性, 试验中的 vSDN 拓扑的规模以及 vSDN 时延需求均按照正态分布随机生成。

为了评估算法的性能, 本实验将本文所提出的算法与最小时延的 vSDN 控制器放置算法<sup>[26]</sup>进行了对比, 统计了相对于最小时延的 vSDN 控制器放置算法增加的控制器—交换机平均时延和接收的 vSDN 数量。为了降低误差, 每组实验进行了 100 次, 最后取平均值。

### 4.2 实验结果

实验结果如图 7 所示。

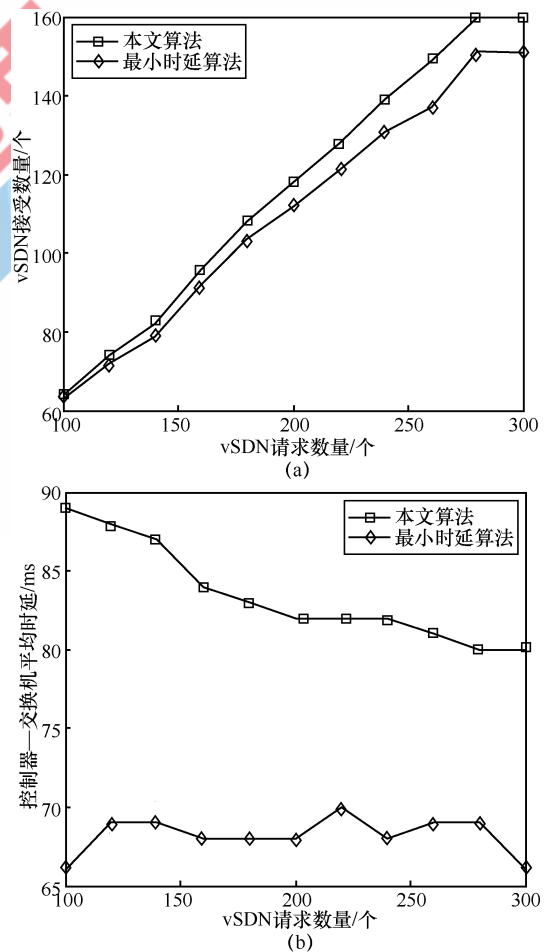


图 7 实验结果

从图 7 (a) 中可以发现, 本文所提出的低时延的 vSDN 控制器放置算法可以有效地提高

vSDN 嵌入的接受率,可以达到提高网络设备的资源利用率的目的。从图 7 (b) 中可以发现,本文所提出的低时延的 vSDN 控制器放置算法虽然增加了控制器到交换机的平均时延,但是时延依然在需求范围之内。总之,本文所提出的 vSDN 控制器放置算法在满足虚拟网络交换机—控制器时延需求的前提下,提高了 vSDN 的接受率及物理网络资源的利用效率。

## 5 结束语

网络虚拟化技术已经成为了解决传统网络体系结构僵化问题的主流技术,其与云计算、数据中心网络、SDN 以及 NFV 的联系日益紧密。针对 vSDN 中控制器位置选择问题,本文提出了一种时延敏感的 vSDN 控制器放置的在线算法。实验验证了该算法既保证了 vSDN 合理地嵌入,又提高了网络资源的利用效率。但是,本文并没有考虑网络中链路的负载情况,也没有考虑多对一、一对多的虚拟网元映射情况,接下来将针对这些问题继续深入地进行研究。

## 参考文献:

- [1] CARAPINHA J, JI M, NEZ J. Network virtualization: a view from the bottom[C]//ACM Workshop on Virtualized Infrastructure Systems and Architectures, August 17, 2009, Barcelona, Spain. New York: ACM Press, 2009: 73-80.
- [2] KIM D Y, MATHY L, CAMPANELLA M, et al. Future internet: challenges in virtualization and federation[C]//2009 Fifth Advanced International Conference on Telecommunications, May 24-28, 2009, Mestre, Italy. New Jersey: IEEE Press, 2009: 1-8.
- [3] PSEL A, WOESNER H. OFELIA: Pan-European test facility for OpenFlow experimentation[C]//4th European Conference on Towards a Service-Based Internet, October 26-28, 2011, Poznan, Poland. New York: ACM Press, 2011: 311-312.
- [4] BOUCADAIR M, GEORGATSOS P, WANG N. The AGAVE approach for network virtualization: differentiated services delivery[J]. *Annals of Telecommunications*, 2009, 64(5): 277-288.
- [5] ROUSKAS G N, BALDINE I, CALVERT K, et al. ChoiceNet: network innovation through choice[C]//2013 17th International Conference on Optical Network Design and Modeling (ONDM), April 16-19, 2013, Brest, France. New Jersey: IEEE Press, 2013: 1-6.
- [6] AT&T. Domain 2.0 white paper[EB/OL]. (2013-11-13) [2017-01-13]. <http://www.docin.com/p-878370536.html>.
- [7] 王胡成, 徐晖, 程志密, 等. 5G 网络技术研究现状和发展趋势[J]. *电信科学*, 2015, 31(9): 156-162.  
WANG H C, XU H, CHENG Z M, et al. Current research and development trend of 5G network technologies[J]. *Telecommunications Science*, 2015, 31(9): 156-162.
- [8] 管红光, 杨宜镇, 任万里, 等. 基于 OpenFlow 的网络虚拟化技术研究应用[J]. *电信科学*, 2014, 30(1): 96-102.  
GUAN H G, YANG Y Z, REN W L, et al. Research and application of network virtualization technology based on OpenFlow[J]. *Telecommunications Science*, 2014, 30(1): 96-102.
- [9] GENI[EB/OL]. [2017-01-13]. <https://sites.google.com/asu.edu/grw-fiu/home>.
- [10] Internet2[EB/OL]. [2017-01-13]. <http://www.internet2.edu/research-solutions/>.
- [11] FIRE[EB/OL]. [2017-01-13]. <https://www.ict-fire.eu/new-ict-13-2016-call-projects-started/>.
- [12] SHERWOOD R, GIBB G, YAP K K, et al. FlowVisor: a network virtualization layer[EB/OL]. (2014-05-14)[2017-01-13]. [https://www.researchgate.net/publication/238109224\\_FlowVisor\\_A\\_Network\\_Virtualization\\_Layer](https://www.researchgate.net/publication/238109224_FlowVisor_A_Network_Virtualization_Layer).
- [13] AL-SHABIBI A, DE LEENHEER M, GEROLA M, et al. OpenVirteX: make your virtual SDNs programmable[C]//The Workshop on Hot Topics in Software Defined Networking, August 22, 2014, Chicago, Illinois, USA. New York: ACM Press, 2014: 25-30.
- [14] DRUTSKOY D, KELLER E, REXFORD J, et al. Scalable network virtualization in software-defined networks[J]. *IEEE Internet Computing*, 2013, 17(2): 20-27.
- [15] COMMITTEE L S. IEEE standard for local and metropolitan area networks—virtual, bridged local area networks[S]. 2006.
- [16] FERGUSON P, HUSTON G. What is a VPN[R]. 1998.
- [17] BLENK A, BASTA A, REISSLEIN M, et al. Survey on network virtualization hypervisors for software defined networking[J]. *IEEE Communications Surveys & Tutorials*, 2015, 18(1): 655-685.
- [18] SALVADORI E, CORIN R D, BROGLIO A, et al. Generalizing virtual network topologies in OpenFlow-based networks[C]//2011 IEEE Global Telecommunications Conference (GLOBECOM 2011), December 5-9, 2011, Houston, Texas, USA. New Jersey: IEEE Press, 2011: 1-6.
- [19] CORIN R D, GEROLA M, RIGGIO R, et al. VeRTIGO: network virtualization and beyond[C]//European Workshop on Software Defined Networking, October 25-26, 2012, Darmstadt, Germany. New York: ACM Press, 2012: 24-29.
- [20] MIN S, KIM S, LEE J, et al. Implementation of an OpenFlow network virtualization for multi-controller environment[C]//International Conference on Advanced Communication Technology, February 19-22, 2012, PyeongChang, Korea. New Jersey: IEEE Press, 2012: 589-592.
- [21] ELAZZAB M, BEDHIAF I L, LEMIEUX Y, et al. Slices isolator for a virtualized OpenFlow node[C]//First International



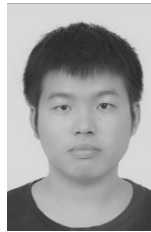


Symposium on Network Cloud Computing and Applications, November 21–23, 2011, Washington, DC, USA. New Jersey: IEEE Press, 2011: 121-126.

- [22] HUANG S, GRIFFIOEN J. Network hypervisors: managing the emerging SDN chaos[C]//International Conference on Computer Communications and Networks, July 30–August 2, 2013, Nassau, Bahamas. New Jersey: IEEE Press, 2013: 1-7.
- [23] MATIAS J, JACOB E, SANCHEZ D, et al. An OpenFlow based network virtualization framework for the cloud[C]//IEEE Third International Conference on Cloud Computing Technology & Science, November 29–December 1, 2011, Athens, Greece. New Jersey: IEEE Press, 2011: 672-678.
- [24] YAMANAKA H, KAWAI E, ISHII S, et al. AutoVFlow: autonomous virtualization for wide-area OpenFlow networks[C]//2014 Third European Workshop on Software Defined Networks, September 1–3, 2014, Budapest, Hungary. New York: ACM Press, 2014: 67-72.
- [25] KOPONEN T, AMIDON K, BALLAND P, et al. Network virtualization in multi-tenant datacenters[C]//11th USENIX Conference on Networked Systems Design and Implementation, April 2–4, 2014, Seattle, WA, USA. New York: ACM Press, 2014: 203-216.
- [26] BOZAKOV Z, PAPADIMITRIOU P. AutoSlice: automated and scalable slicing for software-defined networks[C]//ACM Conference on CONEXT Student Workshop, December 10, 2012, Nice, France. New York: ACM Press, 2012: 3-4.
- [27] FISCHER A, BOTERO J F, TILL BECK M, et al. Virtual network embedding: a survey[J]. IEEE Communications Surveys & Tutorials, 2013(4): 1-19.
- [28] DEMIRCI M, AMMAR M. Design and analysis of techniques for mapping virtual networks to software-defined network substrates [J]. Computer Communications, 2014, 45(2): 1-10.



郭志强 (1994–), 男, 天津大学计算机学院硕士生, 主要研究方向为软件定义网络、云计算、未来网络体系结构等。



陈胜 (1993–), 男, 大连理工大学计算机学院硕士生, 主要研究方向为计算机网络、无线网络、云计算。



周晓波 (1985–), 男, 博士, 天津大学计算机学院副教授, 主要研究方向为信源信道联合编码、协同无线通信、网络信息论、云计算、软件定义网络等。



齐恒 (1981–), 男, 博士, 大连理工大学计算机学院副教授, 主要研究方向为软件定义网络、多媒体计算、数据中心、云计算等。

[作者简介]



董晓东 (1990–), 男, 天津大学计算机学院博士生, 主要研究方向为计算机网络、未来网络体系结构、软件定义网络等。



李克秋 (1971–), 男, 博士, 天津大学计算机学院教授, 主要研究方向为数据中心网络、云计算、无线网络等。