doi:10.11959/j.issn.1000-436x.2016083

SDN 试验床网络虚拟化切片机制综述

刘江[1,2]，黄韬[1,2]，张晨[1]，张歌[1]
（1. 北京邮电大学网络与交换技术国家重点实验室，北京 100876；2. 北京未来网络科技高精尖创新中心，北京 100124）

摘 要：未来网络体系架构和关键技术的研究需要灵活开放的测试验证环境，基于传统分布式的网络架构难以达到动态虚拟化、有效管控和新协议灵活部署的需求。随着软件定义网络（SDN）技术的出现和发展，上述问题找到了有效的解决途径，因此，基于 SDN 构建网络试验床成为了近年来该领域的主流研究方向之一。其中，基于 SDN 的网络虚拟化切片技术更是试验床中的核心支撑技术，可以根据不同试验的需求切分物理网络资源，从而提供并行、独立的网络环境。将重点研究基于 SDN 的试验床中使用的网络虚拟化切片机制，从“流量识别和切片网络标识”、“虚拟节点抽象”和“虚拟链路抽象”这 3 个关键技术出发，对当前基于 SDN 试验床中的典型网络虚拟化切片机制进行介绍与分析，并总结了该领域未来可行的研究方向。

关键词：软件定义网络；OpenFlow；网络虚拟化；试验床
中图分类号：TP393.0 **文献标识码：**A

Research on network virtualization slicing mechanism in SDN-based testbeds

LIU Jiang[1,2], HUANG Tao[1, 2], ZHANG Chen[1], ZHANG Ge[1]
(1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. Beijing Advanced Innovation Center for Future Internet Technology, Beijing 100124, China)

Abstract: The researches on future network architecture and key logies need test environment both open and flexible. Traditional distribute architecture was short on effective control and new protocol deployment. The emerging of the software defined networking (SDN) technology provides a promising way to s this problem and become a major research direction in recent years. In the SDN based network testbed, the virtual network slicing technology was a key issue since it could separate physical resource and provided individual virtual network environment. Therefore, the slicing methods of some typical network virtualization platforms with the perspective of “slice identifier”, “virtual nodes abstraction” and “virtual links abstraction” were introduced , and the future research directions were concluded in this field.

Key words: software defined networking, OpenFlow, network virtualization, testbed

1 引言

随着应用层业务种类与需求愈发的多样化，尤其是对服务质量需求的不断提升，基于 TCP/IP 的传统网络架构面临着越来越严峻的挑战。为此，研究人员提出了许多新型的算法、协议和网络架构。为了获得可靠的试验数据，并且不对现有的生产网络造成影响，往往需要通过建设专门的网络试验床为这些创新性的技术提供真实、独立的试验环境。网络试验床可以分为专用和通用 2 类[1]，专用试验床用于特定的网络试验，平台资源不能为不同类型的试验所复用，构建成本较高，而且平台间难于进

收稿日期：2015-11-01；**修回日期：**2016-03-15
基金项目：国家自然科学基金资助项目（No.61302089）；国家高技术研究发展计划基金资助项目（“863”计划）（No.2015AA016101, No.2015AA015702）；北京市科技新星计划基金资助项目（No.Z151100000315078）
Foundation Items: The National Natural Science Foundation of China (No.61302089), The National High Technology Research and Development Program of China(863 Program)(No.2015AA016101, No.2015AA015702), Beijing New-Star Plan of Science and Technology (No.Z151100000315078)

行互通。而通用试验床则能够承载不同类型的网络试验，并且能够为试验提供编程接口，具备一定的经济性和灵活性。

通用试验床中，不同的试验要彼此隔离，因此往往需要使用网络虚拟化的技术来并行支撑这些试验。PlanetLab[2]通过隧道叠加在传统的IP网络上，并为不同试验网络划分不同的网段，以提供彼此独立的试验环境。不过PlanetLab切片的开通需要复杂的隧道配置，另外跨越公网进行Overlay组网，其传输服务质量也难以得到有效的保障；GENI[3]则通过在骨干网上划分VLAN的方式实现不同试验网络的隔离，不过这使试验拓扑受限于骨干网的拓扑而不够灵活。

近年来，软件定义网络（SDN, software defined network）引发了广泛的关注，基于SDN的通用网络试验床也得到了广泛的建设，如GENI OpenFlow[4]、OFELIA[5]、RISE[6]等。SDN将网络的转发逻辑与设备分离，允许用户在远端的控制器上进行灵活的编程，以集中控制设备的转发，能够为试验新的算法、协议和网络架构提供了便利的手段。为了满足SDN试验对网络多样、多变的需求，SDN试验床需要具备更灵活、自动化的网络虚拟化能力，要求能够动态调度底层物理网络资源，按需为试验者分配试验拓扑以及链路带宽等逻辑资源，并将这些资源与服务自动地编排、打包提供给试验用户，这往往需要通过专门的SDN网络虚拟化平台来实现。

在网络虚拟化试验床中，通常将一个试验获得的逻辑资源称为一个“试验切片”，SDN网络虚拟化平台的设计关键就是网络的切片机制，即如何在物理资源和逻辑资源间进行映射，并在不同的逻辑资源间实现隔离。本文将从“流量识别和切片网络标识”、“虚拟节点抽象”和“虚拟链路抽象”3个关键技术对SDN网络虚拟化平台的切片机制进行深入的分析。

2 SDN试验床概述

2.1 SDN技术简介

SDN技术起源于Ethane[7]，它通过在一个逻辑集中的控制器上进行编程，远程地对企业网络进行管理，获得了更好的安全性与更强的可管理性，这种数控分离的架构以及网络的可编程性即为SDN的本质特征。集中式的控制器更容易获得网络的全局视图——除了网络拓扑以外，还包括链路实时带宽，网络中的流量特征等关键信息。结合这些信息，控制器能够方便地对流量进行调度以实现实时高效的网络优化。

OpenFlow[8]的提出是SDN的一个重要的里程碑，集中式的控制器与OpenFlow交换机间交互信令以收集网络视图信息，并通过分发流表项来指导底层交换机的转发。OpenFlow流表项主要包含“匹配域”和“动作集”2部分，符合“匹配域”特征的流量，交换机将按照相应的“动作集”对其进行处理。在OpenFlow v1.0中，“匹配域”涵盖了物理层到传输层的主要字段，控制器可以基于其中的一个字段或者组合多个字段来描述流量的特征。“动作集”则提供了改写、转发、入队等多种处理方式，结合灵活的匹配方式，控制器可以对网络中各种流量进行灵活的处理。OpenFlow在美国各个大学中的成功部署，使SDN相比于传统网络的显著优势得以体现，极大地推动了SDN技术的发展。OpenFlow技术仍在不断发展完善中，目前已经成为SDN的主流技术之一。

2.2 基于SDN技术构建试验床

SDN由于其灵活的特性，可以为网络研究人员实践新的网络算法与协议提供便利的试验手段，SDN试验床就是专门用来承载这些网络试验的。SDN试验床要对计算、存储、网络等物理基础设施进行高度的抽象，按照试验需求对其进行动态的调度，试验切片需要与底层的物理资源相解耦，切片间要确保高度的隔离性，使从每个试验者各自的角度来看，都仿佛在独自占用底层的物理资源。

SDN试验床的基本架构如图1所示。其中，控制框架联系了底层的物理资源和上层的试验用户，负责对试验切片进行全生命周期的管理。控制框架接收试验者提交的切片资源需求，结合物理资源的分布情况与实时状态，将其以最优的方式映射到物理资源中以支持试验的开展。切片开通后，控制框架负责实时监测切片的资源状态，当切片需求改变时动态调整映射关系，试验结束后控制框架将回收切片资源以备后续使用。一个合理的控制框架应该包含以下几个功能模块：试验管理平台负责记录用户数据，接收切片需求并监测运行切片的状态；云管理平台负责调度底层的计算存储资源，为切片分配虚拟机和存储单元；

网络虚拟化平台负责调度底层的网络资源，满足试验对于网络拓扑以及链路带宽的需求，并保证切片网络间的高度隔离；联邦互联接口实现与其他试验床的资源互操作。

在上述功能中，网络虚拟化平台是本文的主要研究对象，需要满足试验者对于切片网络的多样化需求。图 1 给出几个典型的试验需求，其中，试验 1 希望验证某增强型 STP 协议的可行性，需要试验床为其提供一个环型的网络拓扑；试验 2 希望在网络中运行一个 QoS 分级策略，需要试验床为其提供一个线性的网络拓扑，并且保障切片中每条链路的带宽相同；试验 3 研究的是数据中心内的流量调度问题，需要试验床为其提供树型的拓扑，各层间具备一定的带宽收敛比。这些切片都需要运行在同一套底层物理网络设施中，网络虚拟化平台在满足各试验需求的同时，还要保障不同试验切片间的高度隔离。

网络虚拟化平台的实现难点体现在，当并行运行的试验个数越来越多，或者试验的需求越来越复杂时，对网络虚拟化平台资源分配的效率就要求越高。另外，试验的需求往往不是一成不变的，如当试验 1 在 3 台交换机构成的环型拓扑中验证成功后，很可能希望扩大切片网络以验证协议的可扩展性，这就要求网络虚拟化平台能够动态地调整切片形态，同时不影响其他切片的运行。总之，为了满足多样多变的试验需求，SDN 试验床中的网络虚拟化平台将起着至关重要的作用。

3　SDN 网络虚拟化问题及切片技术

3.1　SDN 网络虚拟化问题概述

传统的网络虚拟化技术，如 VLAN 和 VPN，设备运行着各自的转发逻辑，对这样的网络进行虚拟化需要分别对每一台设备进行操作，再加上不同厂商的设备具有不同的硬件架构和软件逻辑，因此网络虚拟化的配置和操作通常是非常复杂的。另外由于自动化的缺失，当虚拟网络发生变化时，更改原来的配置工作量将十分巨大，因此并不适合构建网络虚拟化平台。

SDN 的集中控制和可编程能力则恰好解决了上述问题，即可以在集中式架构的合适位置引入“转换单元”，按照一定的策略实现虚拟逻辑资源与真实物理资源间的映射。这种映射可以非常灵活，虚拟逻辑资源可能是物理资源的一个子集，也可能是完全解耦于物理资源的。当虚拟网络发生变化时，“转换单元”还可以自动调整映射的策略。通过 SDN 这种集中式可编程特性实

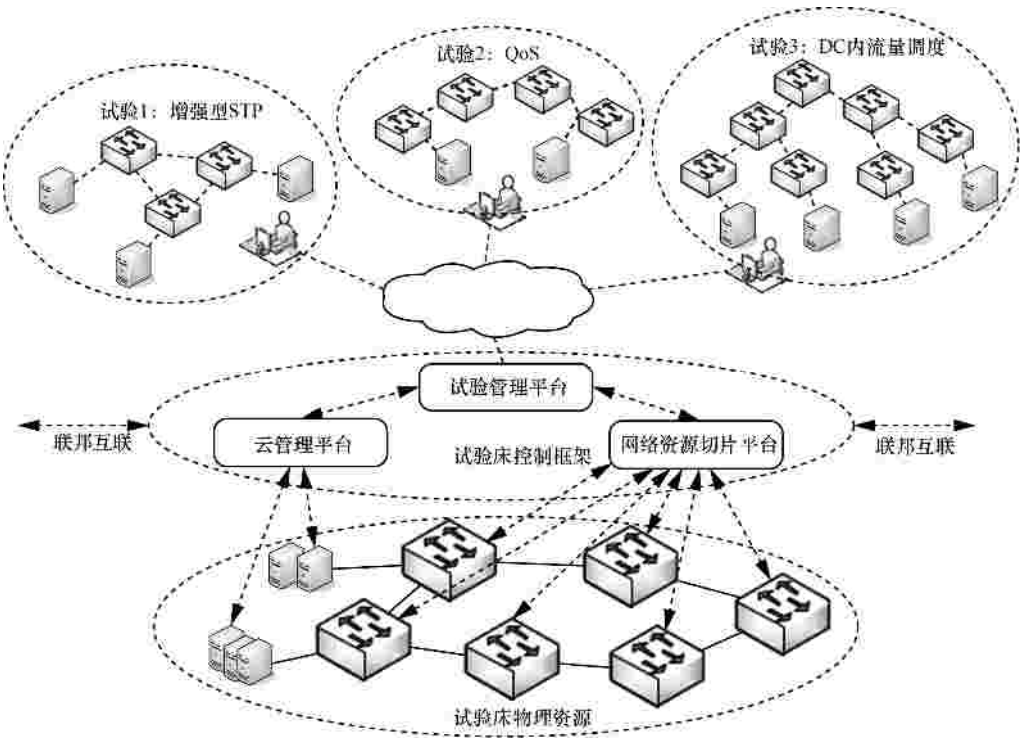

图 1　SDN 试验床基本架构

现自动化，可以简化传统网络虚拟化场景中复杂的配置工作，使网络虚拟化技术能够更具灵活性和弹性。

在基于 SDN 的网络虚拟化平台中，“转换单元”可以工作在不同的位置上，从而带来不同的性能特征，主要可以分为以下 3 类，分别与计算虚拟化的 Bare Metal、HyperVisor、OS Container 这 3 类技术相对应，如图 2 所示。

1) 转换单元集成于交换机中。每个物理交换机运行多个虚拟交换机实例，每个实例连接一个控制器，转换单元根据虚拟网络的映射信息把流量交付给相应的控制器进行处理。在这种思路下，虚拟化直接在转发设备上进行，可类比于计算虚拟化中的 Bare Metal 方式。ONF 在 OF-CONFIG[9]的白皮书中提及了这种情况，即一个物理的 OF-Capable Switch 中可以有多个 OF-Logical Switch。

2) 转换单元作为一个独立的外置设备，工作在物理交换机和各虚拟网络的控制器之间。在这种情况下，转换单元根据虚拟网络的映射信息，对物理交换机和控制器之间交互的信令进行转换，将控制信令交付给正确的控制器和物理交换机处理。在这种模式下可以将“转换单元”看作物理交换机和控制器间的透明代理。这种思路可以类比于计算虚拟化中的 HyperVisor 方式，其代表技术有 FlowVisor[10]、VeRTIGO[11]和 OpenVirtex[12]等。

3) 转换单元位于控制器中，作为控制器中一个特殊的 APP。这种模式下，虚拟网络的用户共同接受一个控制器的调度，不同的虚拟网络可以运行不同的 APP。这种思路类比于计算虚拟化中的 Container 方式，FlowN[13]是基于该思路进行设计的，OpenDayLight[14]通过 VTN 组件实现了网络虚拟化，NVP[15]也是基于这种方式实现数据中心中多租户的隔离。

由于使用网络试验床的研究人员往往希望由自己的控制器来定义虚拟网络切片的行为，因此 SDN 试验床主要采用了第 2 种模式，即基于透明代理的模式实现网络虚拟化平台。本文将要介绍的网络虚拟化平台及其虚拟切片技术都是基于这一模式实现的。

3.2 SDN 网络虚拟化切片关键技术

网络虚拟化平台的主要工作包括 2 个部分：切片和映射。其中，切片主要负责将虚拟网络标识和隔离，映射主要负责将物理资源按合理地分配给虚拟网络切片。理想的情况下，SDN 试验床应该能够满足试验者对切片拓扑以及网络性能（包括带宽、试验、抖动等）的任意需求，网络虚拟化平台根据用户的需求进行最优的资源映射并自动开通试验切片网络，需求变化时能够弹性地延展切片网络或者回收过期切片的网络资源。切片开通后，用户将获得一些从分布式的物理设备中抽象出来的虚拟节点，虚拟节点间通过虚拟链路相连，不同的切片间要保证高度的隔离。

本文将重点研究 SDN 试验平台的切片技术，如图 3 所示。与真实的物理网络类似，一个虚拟的网络切片也是由虚拟节点和虚拟链路组成，因此，节点、链路抽象是切片技术的重要组成部分。另外，切片技术还应该能够识别流量，即创建虚拟网络和外界的接口规则。最后，切片应该能够支持完备的

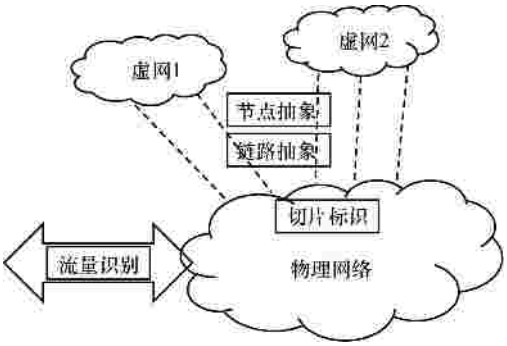

图 3 SDN 网络虚拟化切片关键技术

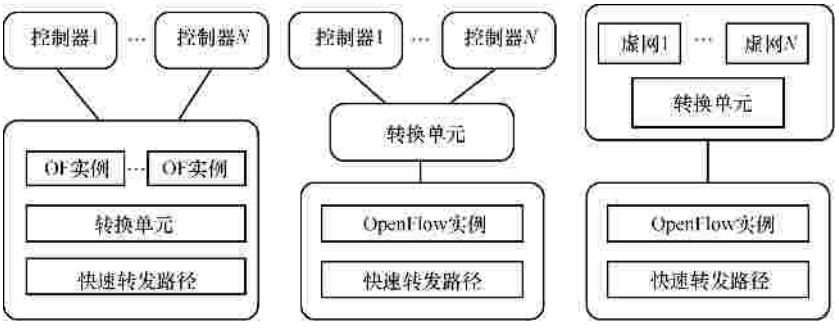

图 2 SDN 网络虚拟化中的转换单元

切片标识，即在虚网运行过程中，尽管 SDN 控制器可能任意调整流量，仍然能够通过标识保障虚网间的隔离性。因此，本节将从流量识别与切片标识、虚拟节点抽象、虚拟链路抽象 3 个关键技术出发，对 SDN 试验床的网络虚拟化切片技术进行详细分析。

3.2.1　流量识别与切片标识

SDN 网络虚拟化平台要对不同的试验进行隔离，最基本的前提就是需要识别不同试验用户的流量，并对其进行标识。流量的识别往往发生在接入设备上，识别的策略可以是面向主机的（如基于物理端口号或者基于 MAC 地址的），也可以是面向业务的（如基于 IP 地址或者应用端口号的）。OpenFlow 匹配域的 N-Tuple 机制可以基于上述字段的任意组合制定适宜的流量识别策略，具备很强的灵活性。接入设备识别出流量后，往往还需要对流量进行特定的切片标识，以便后续的传输设备进行特定的处理，为了并行地支持多个试验，不同切片网络流量的标识不能互相冲突。另外，在切片内部，控制器应该可以实现对流量进行尽可能灵活的修改，以实现网络功能创新，如可以替换目的 IP 实现网络自动代理功能，但此时应该保证设计的切片标识不受影响，避免切片流量泄漏。

3.2.2　虚拟节点抽象

虚拟节点抽象需要向用户描述虚拟化的节点模型，具体可以分为“一虚多”和“多虚一”2 种模型。“一虚多”模型对网络技术来说比较常见：在传统网络中一台局域网交换机逻辑上被 VLAN 分为多个虚拟交换机；一些厂家的路由器也支持实例化成为多个虚拟路由器，如思科的 Logical Router。类似地，SDN 网络虚拟化平台应该能够生成并管理虚拟节点，并在虚拟节点和物理节点间做透明的映射。同时，SDN 网络虚拟化平台要保证不同虚拟节点间 CPU、转发表等资源的相互隔离。相反，在“多虚一”模型中，网络虚拟化平台需要对多台物理设备中的资源进行组合，向试验者呈现一个或多个虚拟设备。

3.2.3　虚拟链路抽象

虚拟链路抽象需要向用户描述虚拟化的链路模型，用于表示虚拟节点间的连接。虚拟链路可以是物理交换机内部的一个通路，可以与物理链路有着一一映射的关系，也可以由多条物理链路和多台物理设备共同模拟形成一条虚拟链路。VPN 网络中，GRE 技术通过隧道对数据分组进行二次封装以模拟虚拟链路，MPLS 技术通过分发 LSP 标签来模拟虚拟链路。SDN 试验床中，网络虚拟化平台面对着试验用户多样的试验拓扑需求，需要具备生成与物理拓扑完全解耦的逻辑拓扑能力，虚拟链路技术对此至关重要。由于不同的试验对于带宽的需求可能是不同的，虚拟链路技术的实现关键是应该尽可能满足试验用户对于这些性能指标的需求。

4　SDN 试验床网络虚拟化切片机制分析

SDN 试验床的网络虚拟化切片机制主要包含流量识别与切片标识、虚拟节点抽象、虚拟链路抽象 3 种关键技术，然而在实现过程中，却可以设计多种不同的技术手段，本文将对现有的 SDN 试验床的网络虚拟化切片机制进行分析，比较分析各种实现手段的优势及不足。

4.1　FlowVisor

FlowVisor[10]（以下简称 FV）由斯坦福大学于 2010 年开发，是第一款基于透明代理模式的 SDN 网络虚拟化平台。FV 通过在网络中切分出并行、独立的切片，为网络试验的创新提供了便利。FV 具备充分的灵活性，通过 OpenFlow 协议的匹配字段，将网络中不同用户、不同网段或不同业务，均看作可以虚拟化调度的资源，管理员通过多个字段的组合灵活地集中定义切片网络策略，省去传统虚拟化方式对路由器和交换机的复杂的手动配置。

FV 是 OpenFlow 物理交换机（以下简称 OF 交换机）与控制器间的透明代理，其工作原理如图 4 所示。对于 OF 交换机来说，FV 是一个特殊的控制器，从各个切片的控制器视角来看，FV 则相当于一个物理的 OpenFlow 网络。FV 负责截获 OF 物理交换机产生的信令，并根据流量的特征和各个切片的匹配规则，将信令送给相应的控制器。另外，切片控制器产生的信令也会首先经过 FV，FV 对信令进行修改和约束后分发给相应的物理 OF 交换机。可见，FV 能够根据虚拟网络流空间的定义规则进行消息的劫持、修改、分发等操作，而切片控制器则只能看到 FV 向它提供虚拟的切片网络视图。FV 对于控制信令是没有限制的，GENI OpenFlow 中部署的 FV 版本是基于 OpenFlow v1.0 的。

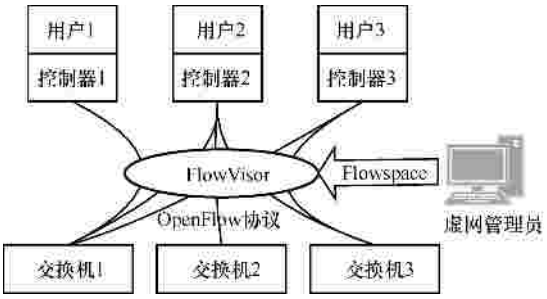

图 4 FlowVisor 工作原理

4.1.1 流量识别与切片网络标识

OpenFlow v1.0 提出了基于 12-Tuple 的匹配方式，包括从 1 层的物理端口到 4 层的应用端口等 12 个字段都可以用来组合匹配数据流，而且提供了相对丰富的处理动作。相应地，基于 OpenFlow 实现的 FV，允许将这些字段（除了 VLAN 优先级字段）都开放给管理员去灵活地制定流量识别策略，其粒度则可粗可细。一个策略被称为一个 Flowspace，一个切片可能对应着多个不连续的 Flowspace。除了更强灵活性以外，切片网络还获得了更大的弹性：只要是匹配了 Flowspace 的流量，就会被自动地识别，不再像传统局域网中新用户的接入需要管理员手动地绑定 VLAN 接入端口。

与传统网络中基于 VLAN、VNI 这些专用的标识字段不同，FV 直接根据流量的特征去识别切片，并不对流量进行附加的标识。

FV 的 Flowspace 机制虽然灵活，但存在着以下 2 个问题：1）切片的地址空间无法复用，如当不同试验用户需要重叠的 IP 地址时很可能会发生匹配规则的冲突；2）控制器很可能会改变 Flowspace 中的某些字段，这样很可能就在切片网络的某个地方发生流量泄露，即切片 A 的流量被识别为切片 B 的流量，导致错误的处理，即 FV 并未实现具备很强隔离性的切片标识。

4.1.2 虚拟节点抽象

在虚拟节点抽象过程中，FV 会对虚拟节点上的资源包括端口、CPU、转发表进行隔离，控制器与交换机握手时，FV 对 SwitchFeatures 消息中物理 OF 交换机上的端口号进行合理的过滤后送给控制器，控制器只能看到属于该切片网络的端口。控制器下发包括泛洪动作在内的流表时，FV 会将该动作映射为多条流表，只从属于该切片的端口上进行转发。CPU 的隔离实现起来相对复杂，FV 给出了节约物理交换机上 CPU 资源的一些思路，希望通过减轻 CPU 的负载来间接地保证虚拟节点间的 CPU 隔离。转发表的隔离对于 FV 是天然实现的，只要各个切片的 Flowspace 不发生冲突，转发表就不会出现冲突的情况。

FV 在一定程度上实现了虚拟资源的隔离，但是 FV 并没有对虚拟节点进行显式的抽象。一方面这使试验用户无法直接管理其虚拟节点，另一方面切片控制器能看到的交换机 ID 和端口 ID 很可能是不连续的，这不符合虚拟化的透明性原则。虽然 FV 并没有显式地抽象虚拟节点实例，但在一定程度上可以说它具备了“一虚多”的基本能力，但对于“多虚一”的场景 FV 并未做支持。

4.1.3 虚拟链路抽象

切片包含的虚拟节点可能是部署在不同的物理交换机上的，需要通过实际的物理链路组合来模拟虚拟链路。FV 明显的缺点是一条虚拟链路只能对应一条物理链路，无法跨越多个物理交换机实现虚拟链路，这意味着逻辑拓扑只能是物理拓扑的一个子集，而无法实现任意切片拓扑的映射。比如说切片逻辑拓扑中相邻的虚拟交换机 VOF1 和 VOF2，它们分别被部署到 OF1 和 OF2 中，则 FV 要求 OF1 与 OF2 在物理上是直连的，这极大地限制了切片部署的灵活性。另外一个问题是，当不同的虚拟节点被部署在相同的物理交换机中，虚拟链路的实现要求设备能够抽象出一个类似于“环回”的逻辑端口在不同的虚拟节点间进行连接，FV 并没有支持这种机制。

4.2 ADVisor

ADVisor（advanced FlowVisor）[16]是由 Create-Net 实验室开发的一款专门面向 OpenFlow 网络的 SDN 网络虚拟化平台，主要解决了 FV 不能支持切片拓扑的任意映射的问题。拓扑任意映射场景如图 5 所示。图 5（b）中的左右 2 种场景都是由图 5（a）映射而来的试验切片拓扑。图 5（b）中，左边这种映射机制下，试验拓扑只是物理拓扑的一个子集，试验拓扑中相邻的虚拟节点在物理上必须相邻；而右边这种映射机制下则没有这种限制关系，虚链路的实现依赖于沿途交换机的透明转发。相比较而言，后者能够更加灵活地调度试验网中的网络资源，如试验需要一个 3 台节点形成的环状拓扑，左边这种机制便无法实现，右边这种机制便可以虚拟出物理网络中不存在的虚链路来满足该需求。FV 没有这么灵活的虚拟链路技术，它只能实现左边这

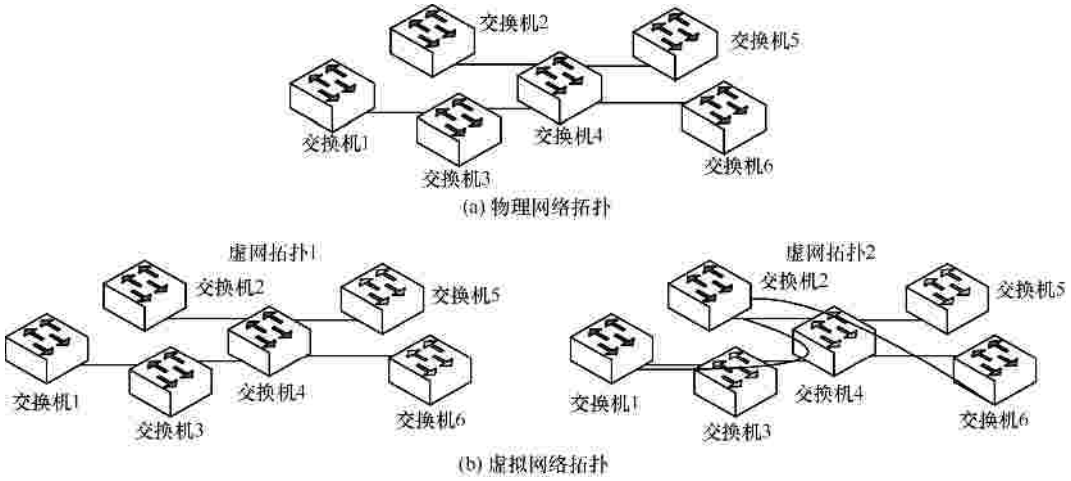

图 5　ADVisor 支持拓扑的任意虚拟化

种映射，而 ADVisor 针对这一点进行了改进，为试验网提供了对实现右边场景中虚拟化需求的支持。不过 ADVisor 只允许通过手动的配置来指定映射关系，仍然不够灵活。

ADVisor 同样位于控制器和交换机间作为透明代理，通过试验网管理者手动配置的 virtual topology configuration 来指导各个切片中虚拟节点和虚拟链路的生成与维护。相比于 FV，ADVisor 增加了 3 个子模块 Port Mapper、Topology Monitor 和 Link Broker，它们根据 Virtual topology configuration 的映射信息，共同实现了虚拟化的映射工作。

4.2.1　流量识别与切片网络标识

ADVisor 没有沿用 FV 的 Flowspace 机制识别切片，而是要求将主机的接入端口与某一切片绑定，所有该主机的流量在网络入口处会被打上 slice_tag_si 的标签，在传输过程中该标签就替代了 FV 的 Flowspace 规则，唯一地标识该流量所属的切片，直到流量被送出该切片时剥掉该标签。考虑到 OF 的协议版本限制，ADVisor 采用了 12 位 VLAN ID 中的一部分比特作为 slice_tag_si 标识切片网络。这种做法虽然不如 FV 灵活，但有效地解决了 FV 的虚网间流量泄露的问题，其代价就是在单级 VLAN 的技术条件下，用户将不允许通过 VLAN 字段来标识业务，另外切片网络的数量上限也必然受到了 VLAN 字段位数的限制。

4.2.2　虚拟节点抽象

ADVisor 通过 Virtual topology configuration 的配置文件来记录物理交换机与虚拟节点的端口号之间的映射关系，切片的控制器不会看到不连续的端口号。这实现起来并不复杂，但是相比于 FV 已有了明显的提升。ADVisor 也同样没有支持虚拟节点的“多虚一”模型。

4.2.3　虚拟链路抽象

虚拟链路技术的实现，依赖于对经过虚拟链路的流量进行标识，以便网络虚拟化平台在沿途的交换机中直接进行转发，而不再上报给切片的控制器。同样由于 OpenFlow v1.0 协议的限制，ADVisor 中使用 VLAN 字段的一些比特作为 slice_tag_vl 去标识虚拟链路上的流量。VLAN 字段总共有 12 bit，试验网管理者需要合理地分配给 slice_tag_vl 和用于标记切片网络的 slice_tag_si，slice_tag_si 和 slice_tag_vl 组合在一起形成 slice_tag，唯一地标识了某个切片网络中的某一条虚链路。

只有虚链路两端连接的虚拟节点才能够与切片的 Controller 交互。物理交换机将流量上报时，ADVisor 中的 Topology Mionitor 根据 slice_tag 判断该物理交换机是虚拟节点还是虚链路上的沿途交换机，前者经 Port Mapper 映射后交给切片的控制器处理，后者则由 Link Broker 直接完成透明的转发，这样在对切片控制器透明的情况下，模拟了虚拟节点间的邻接关系，实现了虚拟链路的功能。然而，ADVisor 中这些映射关系都需要手动进行配置，导致网络状态发生改变时，虚链路的调整不够灵活，难以实现动态优化和路径的 Fail-Over 机制。

4.3　VeRTIGO

VeRTIGO[11]是由 Create-Net 实验室开发的一款 SDN 网络虚拟化平台，在欧盟 FP7 项目下的 OFELIA 试验床中得到了部署。相比于 ADVisor，VeRTIGO 通过算法[18]动态地进行映射，支持虚链

路的动态优化和 Fail-Over 机制。VeRTIGO 还支持了节点虚拟化的“多虚一”模式，允许将多台物理设备抽象为试验网络中一台逻辑的设备，如图 6 所示。

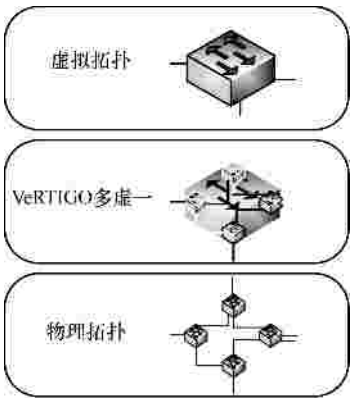

图 6 VeRTIGO 支持虚拟节点的“多虚一”模型

VeRTIGO 在 ADVisor 的基础上增加了几个关键性的模块。Web UI 接受试验用户的切片请求，VT Planner 通过高效的映射算法[17]自动地生成切片配置文件，指导其他模块进行资源抽象与隔离。Node Virtualizer 则实现了“Abstarct Node”以支持节点的“多虚一”模型。

4.3.1 流量识别与切片网络标识

VeRTIGO 虽然是基于 ADVisor 进行开发的，但是其流量识别和切片网络标识技术却使用了 FV 的 Flowspace 机制，提升了网络虚拟化切片定义的灵活性，并未实现 ADVisor 的 slice_tag_si。

4.3.2 虚拟节点抽象

VeRTIGO 实现了虚拟节点的“多虚一”模型，试验者提出对于切片网络的链路带宽等指标的需求后，VT Planner 进行自动映射，并通过 Node Virtualizer 模块自动地封装出一个虚拟的 Abstract Node。这个唯一的虚拟节点将代替底层的物理设备与控制器进行交互，包括握手、SwitchFeatures 等消息都由其代理，控制器向 Abstract Node 下发的 FlowMod 和 PacketOut 消息，也将经过映射并分发给对应的物理 OF 交换机。一个 Abstract Node 往往对应多个物理交换机，跨越多条物理链路，这些物理交换机和链路分别可看作 Abstract Node 的接口线卡和内部背板走线。基于上述思路，VeRTIGO 实现了 SDN 网络虚拟化平台中虚拟节点“多虚一”模式。“多虚一”模型的部署，还将带来流表放置的优化问题[18,19]，是一个可以研究的技术点。

4.3.3 虚拟链路抽象

相比于 ADVisor 通过手工配置进行映射，VeRTIGO 采用了 VT Planner 进行自动化的映射。ADVisor 采用 VLAN 字段中的部分比特作为 slice_tag_vl 去标记虚链路上的流量，这限制了用户对于 VLAN 字段的使用。VeRTIGO 在 VT Planner 映射试验需求后，会在 Storage 中存储流经虚链路的 Header Sequence，当这些流量从沿途交换机上报时，VeRTIGO 的 Classifer 模块识别 Header Sequence，并直接由 Internal Controller 代理用户的控制器向该交换机下发流表以进行透明的转发。

通过算法动态地进行映射的另一个好处就是能够增强网络虚拟化平台的健壮性，通过检测链路的实时负载，VeRTIGO 可以自动地进行虚拟链路映射的动态优化，也能够支持 Fail-Over 机制。

4.4 OpenVirtex

OpenVirteX[12]（以下简称 OVX）是 On.Lab 团队基于 FV 研发的一款 SDN 网络虚拟化平台，其架构如图 7 所示。OVX 具备以下特点：1）为试验提供彼此隔离的虚拟 SDN 网络（vSDN），允许试验者自定义拓扑与网络编址方案；2）显式地为 vSDN 抽象出虚拟节点，允许试验者直观地对虚拟节点进行管理；3）借助后端数据库对网络状态的实时记录，具备了对试验切片执行网络快照的能力。

4.4.1 流量识别与切片网络标识

OVX 在接入交换机上根据端口号和 MAC 地址识别主机所属的切片，流量经过接入交换机后，其源 MAC 地址将被 OVX 重写。重写后的源 MAC 地址的前 24 位为 OVX 从 IEEE 申请得到的保留 OUI，后 24 位将携带着切片网络的标识信息在数据平面传输。传输途中的虚拟节点将根据重写的数据分组源 MAC 地址来识别切片的流量，并送给相应的切片控制器。目的主机所在的出口交换机上，源 MAC 地址将被回写为源主机的 MAC 地址。

相比于 FV 的 Flowspace 机制，OVX 通过重写 MAC 地址解决了切片地址空间无法复用的问题。相比于 ADVisor 使用 VLAN 的部分比特来标识切片的机制，OVX 开放了试验用户在内部切片使用 VLAN 的能力。另外，在接入/出口交换机上 OVX 还会重写/回写源 IP 地址和目的 IP 地址，因此 OVX

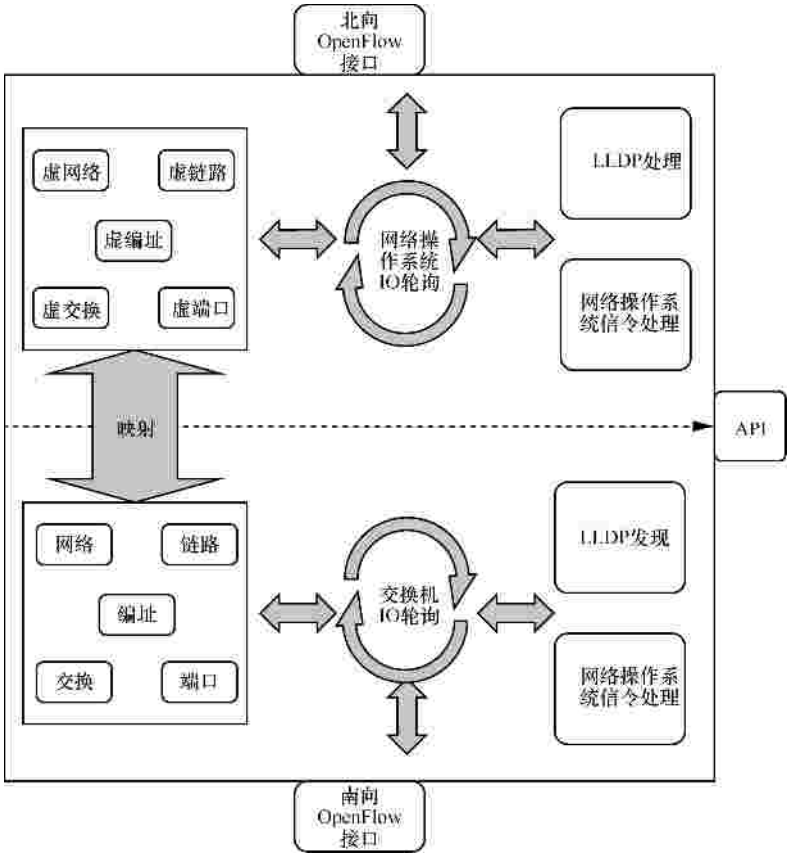

图 7　OpenVirtex 对网络资源进行显式抽象

支持 IP 地址空间的复用，即 2 个虚拟网络切片可以使用同样的 IP 地址。

OVX 的不足在于，它没有支持同一切片中不同的虚拟节点部署在同一物理交换机的场景，在这种的情况下，单个虚拟网络切片的节点数量不能大于物理网络的节点数量。

4.4.2　虚拟节点抽象

在网络虚拟化中，从试验用户的角度来看，虚拟网络中的设备应是其独立拥有的，这需要网络虚拟化平台能够对虚拟交换机进行完整的抽象，实现模拟数据分组的缓存和虚拟流表匹配等功能。OVX 对虚拟节点进行了显式的抽象，试验者能够直接对其进行管理，形成了较为完整的虚拟数据平面。在此基础上，OVX 将切片控制器投放用于拓扑探测的 LLDP 分组限制在了虚拟数据平面内部，避免了将这些数据分组被投放到真实的设备中，节约了数据平面的带宽资源。

OVX 对“一虚多”和“多虚一”2 种模型都提供了良好的支持。“一虚多”模型中，一台物理交换机可以被抽象为多个不同的虚拟节点，包括虚拟设备的标识和虚拟端口的端口号，对于试验者来说都会是连续而完整的。在“一虚多”模型中，OVX 支持将多个物理交换机模拟一个逻辑上的 OVXBigSwitch，通过手动配置或者最短路径计算来模拟 OVXBigSwitch 的内部背板走线。

4.4.3　虚拟链路抽象

与切片网络标识技术类似，OVX 通过对目的 MAC 地址进行重写来标识虚拟链路以及该虚拟链路所承载的流量。具体重写规则如下：经过重写后的目的 MAC 地址的前 24 位为 OVX 从 IEEE 申请得到的保留 OUI，后 24 位将携带着 vLink ID 和 Flow ID 的信息。这里的 Link ID 即为该切片网络中虚拟链路的标识，而 Flow ID 不是对应着某一个流表，而是记录着该次通信的源主机和目的主机的 MAC 地址，以便在出口交换机上进行主机的 MAC 地址回写。接入交换机重写后，源 MAC 地址用来区分不同的切片流量，在传输过程中不会改变；目的 MAC 地址中的 vLink ID 用来标识流量当前所在的虚拟链路，每经过一个中继虚拟节点后改变一次。

在实现虚拟链路时，OVX 可以备份一条次优的路径，当主用虚拟链路经过的物理节点或者物理链路失效时，OVX 自动地将该虚拟链路映射到备份路径上，以实现 Fail-Over 机制。当失效点恢复时，OVX 将重新切换回原来的最优路径。

4.5 CoVisor

CoVisor[20]是 Princeton 大学以 OVX 为原型开发的一款 SDN 网络虚拟化平台，设计架构如图 8 所示。其中，不同试验的控制器负责实现不同的网络服务，试验用户通过编写 CoVisor API 制定切片的拓扑。当不同的控制器下发的流表被分发到同一台物理 OF 交换机中时，很可能出现无法协同工作的情况，CoVisor 的设计目标是在提供虚拟拓扑的基础上，对不同控制器下发的流表进行重新的编译，以协调它们对网络的控制逻辑，同时监测各控制器的行为以防止其操作越界。

4.5.1 流量识别与切片网络标识

CoVisor 适用于试验平台中的网络虚拟化，每个切片可以针对不同的业务进行控制。基于这个出发点，CoVisor 使用了 FV 的 Flowspace 机制，根据流量的特征对切片网络进行识别和标识。

4.5.2 虚拟节点抽象

由于 CoVisor 具备对流表进行聚合的能力，因此它支持将同一网络切片中的多个虚拟节点部署到同一台物理交换机中，完整地实现了虚拟节点抽象的“一虚多”模型。下面通过一个实例对此进行分析。

如图 8（a）所示，在一台物理 OF 交换机 S 中为一个切片网络抽象出 3 台虚拟交换机 A、B 和 C，其中，虚线反映了虚拟链路到物理链路的映射关系，实线链路是 CoVisor 虚拟出来的，不对应任何底层的资源，连接了 AB 与 BC。图 8（b）给出 A、B、C 这 3 台虚拟交换机中的转发流表，为了在 S 上实现正确的转发，CoVisor 要对这些流表项进行聚合。以端口 1 进入 S 的流量为例，如果其目的地为 2.0.0.0/16，聚合后将由端口 2 送出；如果其目的地为 1.0.0.0/24，聚合后将由 S 的端口 3 送出；如果其目的地为 1.0.0.0/8，聚合后将修改目的 IP 为 2.0.0.0 后由 S 的端口 4 送出。为了在 S 中实现这一转发逻辑，CoVisor 将模拟出由端口 1 进入，送往不同目的 IP 地址的分组，由 A、B、C 中的虚拟流表进行处理得到完整的传输路径，并据此对虚拟流表进行聚合，得到最终需要下发到 S 中的 3 条流表，如图 8（c）所示。

CoVisor 对于虚拟节点中流表的隔离也做了改进性的工作，通过对不同切片的控制器开放不同的权限，限制它们对 Match 域的匹配，如负责 MAC Learning 的控制器下发的流表只允许其匹配 MAC 地址，以防止切片控制器的行为越界。

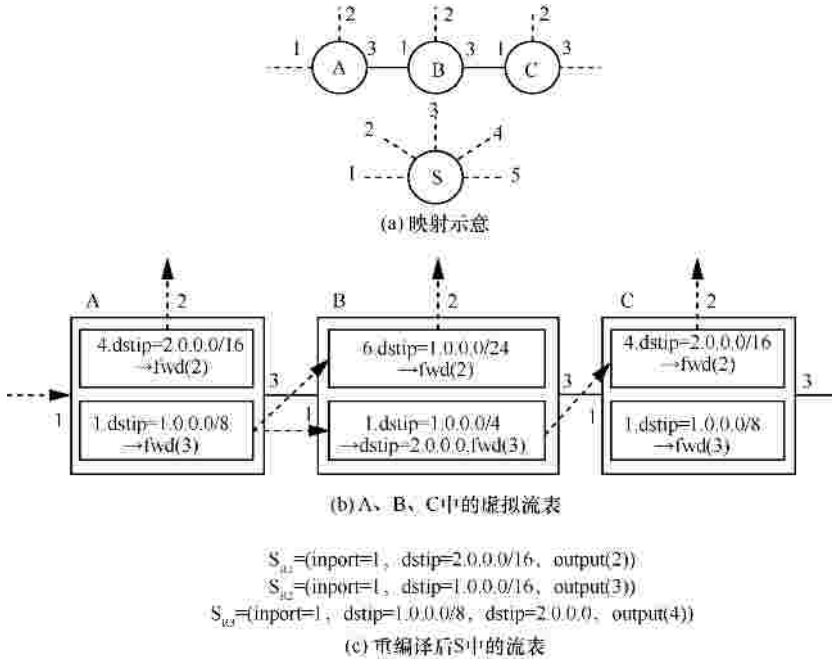

图 8 CoVisor 的流表重编译

4.5.3　虚拟链路抽象

CoVisor 的研究重点不在链路抽象上，因此并未明确地说明其是否支持跨越多台物理交换机的虚拟链路。

4.6　对比分析

前面从流量识别和切片网络标识技术、虚拟节点抽象和虚拟链路抽象 3 个要素技术出发，分析了基于透明代理模式的网络虚拟化平台如何实现虚拟网络的切片。表 1 给出上述 SDN 试验床的虚拟化平台横向对比与总结。

4.6.1　流量识别与切片网络标识

以 FV 为代表的虚拟化平台，包括 VeRTIGO 和 CoVisor，充分利用了 OpenFlow 的特性，能够对流量进行细致的分类，如可将网络中的 HTTP 流量交给切片 A 处理，将内容流量交给切片 B 处理，将潜在的不安全流量交给切片 C 等。上述平台的流量识别策略和切片标识都是直接基于流量特征的，其优点是灵活性较强，并具备对上层业务的感知能力，缺点是可能会面临着流量泄露与控制器操作越界的问题，另外试验用户的地址空间也无法复用。

ADVisor 和 OVX 则根据主机的信息来识别流量，并打上简单的标签来标识不同的切片。ADVisor 中，接入节点通过主机的接入位置即物理端口号来识别流量，并通过 VLAN 字段的部分比特来标识切片；在 OVX 中，接入节点通过物理端口号/主机 MAC 地址来识别流量，并改写源 MAC 地址来标识切片。同一主机产生的所有业务流量都由同一个控制器处理，虚拟化平台只负责将资源切给试验用户，无需考虑控制器间的协调问题。其优点是试验用户地址可以重叠，适用于虚拟多租户的场景，缺点是切片策略粒度较粗而不够灵活，不具备业务的精细感知能力。

4.6.2　虚拟节点抽象

虚拟节点抽象技术分为“一虚多”和“多虚一”2 种模型。“一虚多”模式下需要考虑流表在不同切片间的隔离，这往往要结合“切片网络标识技术”实现。大多数虚拟化平台都支持在一台物理交换机中虚拟出多台交换机，分别交付给不同的切片，然而一些情况下一个切片中的多台虚拟交换机有时不得不部署在同一台物理交换机上（如物理交换机数目小于试验所需的虚拟交换机数目），CoVisor 通过流表的重编译机制支持了这一场景。“多虚一”的模型最初由 VeRTIGO 实现，虚拟交换机 Abstract Node 内部的走线通过相应的虚拟链路技术实现，OVX 的“BigSwitch”也采用了类似的思路。

虚拟节点显式抽象形成的虚拟网络子层可以给网络虚拟化平台带来很多好处。一方面，一些特殊的数据分组可以在虚拟数据平面完成转发（如 LLDP），而不用投放到真实的设备中；另一方面，只有支持显式的虚拟节点才能实现文献[21]中提到的网络虚拟化模型中的有效封装与管控分离。

4.6.3　虚拟链路抽象

虚拟链路上的流量可能承载在一条物理链路上，也可能承载在多条物理链路及其沿途的物理交换机上。FV 只支持前者，而 ADVisor、VeRTIGO、

表 1　SDN 网络虚拟化平台切片机制对比分析

关键技术		FlowVisor	ADVisor	VerTIGO	OpenVirtex	CoVisor
切片网络标识技术	流量识别策略	12 元组的任意组合	物理端口号	12 元组的任意组合	物理端口号/MAC 地址	12 元组的任意组合
	切片标识字段	12 元组的任意组合	VLAN 的部分比特	12 元组的任意组合	改写后的源 MAC 地址	12 元组的任意组合
	试验地址重叠	不支持	支持	不支持	支持	不支持
	适用场景	面向业务流	面向多租户	面向业务流	面向多租户	面向业务流
虚拟节点技术	DPID/Port ID 映射	不支持	支持	支持	支持	支持
	显式抽象	不支持	不支持	不支持	支持	支持
	一虚多	部分支持	部分支持	部分支持	部分支持	完整支持
	多虚一	不支持	不支持	支持	支持	未明确说明
虚拟链路技术	任意拓扑	不支持	支持	支持	支持	支持
	自动化映射算法	不支持	无，通过手动配置	基于链路带宽	基于最短路径	未明确说明
	链路调优	不支持	不支持	支持	不支持	未明确说明

OVX 支持了后者，实现了试验拓扑的任意映射。ADVisor 虚链路的映射是管理员手动配置的，虚拟链路上的流量通过 VLAN 的部分比特来标识；VeRTIGO 使用动态的算法 VT Planner，结合试验用户的带宽需求进行虚链路的映射，虚拟链路上的流量通过 Configuration Files 中记录的 Header Sequence 来标识，OVX 则既支持手配，也支持使用算法进行映射，通过在接入交换机上进行目的MAC地址的重写来标识虚链路上的流量。不过，虚拟链路的标识会限制试验用户对一些字段的使用，如ADVisor 限制了试验用户对于 VLAN 字段的使用，VeRTIGO 由于继承了 FV 的 Flowspace 的概念，为了防止虚拟链路流量泄露，也不允许用户改写 Header Sequence 中的字段。随着 OpenFlow 协议增添新的字段，未来虚拟链路技术的实现可以采用其他字段，从而放宽常用字段的限制。

5 结束语

本文主要描述了 SDN 试验床中基于透明代理模式的网络虚拟化平台，以切片网络标识，虚拟节点和虚拟链路 3 个要件技术为脉络，对该类网络虚拟化平台的不同实现进行了分析与对比，并对当前主流的 SDN 试验床中的网络虚拟化切片技术进行了讨论与分析，最后总结了各项技术的优势与不足。在当前已有研究的基础上，基于 SDN 的网络虚拟化切片技术仍有不少值得研究的内容，具体来说包含以下 3 个方面。

关于流量识别和切片网络标识技术，应结合面向业务流和面向多租户 2 类平台的优点，在入口处通过 Flowspace 提供对业务的精细感知能力，然后打上特定的标签来支持地址的复用。另外，现有的虚拟化平台有一个共同的问题，它们都是基于数据分组中常见的字段或者字段组合去标识切片的，这必然要求强制性地对用户屏蔽掉相关字段，而这些字段有可能是试验用户希望去控制的。未来可能使用一些专用的字段（如 QinQ 技术中的 S-Tag）去标识切片，以使试验用户获得对切片网络的更加完整控制调度权限。

出于管理的需要或者网络优化的考虑，虚拟节点到物理节点的映射可能会发生变化，这涉及到了虚拟节点的迁移问题。另外有一些研究思路利用透明代理的网络虚拟化平台作为南向协议的转换网关，使虚拟化平台不仅仅可以映射协议消息中的字段，还可以在不同的南向协议间进行映射，如标准 OpenFlow 各版本之间进行映射，或者在 OpenFlow TTP[22]与标准 OpenFlow 间进行映射。

同样地，虚拟链路也需要解决动态迁移的问题。另外，由于虚拟链路间的带宽隔离是实现试验网络 QoS 的基础，这方面中多链路的负载均衡、切片流量的统计复用和虚拟链路的流量监测等功能都需要进行进一步的研究。

参考文献：

[1] LIANG J X, LIN Z W, MA Y. Research of future internet network experiment platform: a survey[J]. Chinese Journal of Computers, 2013, 36(5).

[2] CHUN B N, CULLER D E, ROSCOE T. PlanetLab: an overlay testbed for broad-coverage services[J]. Computer Communication Review-CCR, 2003, 33(3): 3-12.

[3] ELLIOTT C. GENI-global environment for network innovations[C]// LCN. c2008.

[4] GENI. GENI OpenFlow[EB/OL]. http://groups.geni.net/geni/wiki/OpenFlow

[5] KÖPSEL A, WOESNER H. OFELIA–pan-european test facility for openflow experimentation[C]//Towards a Service-Based Internet. Springer Berlin Heidelberg, c2011: 311-312.

[6] KANAUMI Y, SAITO S, KAWAI E, et al. RISE: a wide-area hybrid OpenFlow network testbed[J]. IEICE Transactions on Communications, 2013, 96(1): 108-118.

[7] CASADO M, FREEDMAN M J, PETTIT J, et al. Ethane: taking control of the enterprise[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(4): 1-12.

[8] Mckeown N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.

[9] Open Networking Foundation. OpenFlow management and configuration protocol 1.2[EB/OL]. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf, 2014

[10] SHERWOOD R, GIBB G, YAP K K, et al. Flowvisor: a network virtualization layer[R]. OpenFlow Switch Consortium, 2009.

[11] DORIGUZZI C R, GEROLA M, RIGGIO R, et al. Vertigo: network virtualization and beyond[C]// 2012 European Workshop on Software Defined Networking (EWSDN). c2012: 24-29.

[12] AL-SHABIBI A, DE LEENHEER M, GEROLA M, et al. OpenVirteX: make your virtual SDN programmable[C]//The Third Workshop on Hot Topics in Software Defined Networking. ACM, c2014: 25-30.

[13] DRUTSKOY D, KELLER E, REXFORD J. Scalable network virtua-

lization in software-defined networks[J]. Internet Computing, IEEE, 2013, 17(2): 20-27.

[14] MEDVED J, VARGA R, TKACIK A, et al. Opendaylight: towards a model-driven SDN controller architecture[C]//IEEE 15th International Symposium, c2014: 1-6.

[15] KOPONEN T, AMIDON K, BALLAND P, et al. Network virtual za-tion in multi-tenant datacenters[C]//USENIX NSDI. c2014.

[16] SALVADORI E, DORIGUZZI CORIN R, et al. Generalizing vi ual network topologies in OpenFlow-based networks[C]//Global Tele-communications Conference (GLOBECOM 2011). c2011: 1-6.

[17] RIGGIO R, DE PELLEGRINI F, SALVADORI E, et al. Progressive virtual topology embedding in openflow networks[C]. 2013 IFIP/IEEE International Symposium on Integrated Network Management. c2013: 1122-1128.

[18] KANIZO Y, HAY D, KESLASSY I. Palette: Distributing tab s in software-defined networks[C]//2013 Proceedings IEEE INFOCOM. c2013: 545-549.

[19] KANG N, LIU Z, REXFORD J, et al. Optimizing the one big switch abstraction in software-defined networks[C]//Ninth ACM Conference on Emerging Networking Experiments and Technologies. c2013: 13-24.

[20] JIN X, REXFORD J, WALKER D. Incremental update for a composi-tional SDN hypervisor[C]//The Third Workshop on Hot Topics in Software Defined Networking. c2014: 187-192.

[21] CARAPINHA J, JIMÉNEZ J. Network virtualization: a view from the bottom[C]//The 1st ACM Workshop on Virtualized Infrastructure Sys-tems and Architectures. c2009: 73-80.

[22] Open Networking Foundation. OpenFlow table type patterns 1.0[EB/OL].https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/OpenFlow%20Table%20Type%20Patterns%20v1.0.pdf, 2014.

作者简介：

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

张晨（1991-），男，黑龙江哈尔滨人，北京邮电大学硕士生，主要研究方向为软件定义网络、网络虚拟化、云网络。

张歌（1991-），男，新疆乌鲁木齐人，北京邮电大学硕士生，主要研究方向为软件定义网络、网络虚拟化、虚拟私有云网络等。