

硕士学位论文



弹性光网络环境下虚拟光 网络映射的研究

作者姓名： 龚 龙

学科专业： 通信与信息系统

导师姓名： 朱祖勍 副教授

完成时间： 二〇一五年五月

University of Science and Technology of China
A dissertation for master's degree



Virtual Optical Network Embedding over Elastic Optical Networks (EONs)

Author : Long Gong
Speciality : Communications and Information System
Supervisor : Prof. Zuqing Zhu
Finished Time : May, 2015

弹性光网络环境下虚拟光网络映射的研究

六系

龚龙

中国科学技术大学

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文,是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外,论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名: _____ 签字日期: _____

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一,学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权,即:学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅,可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

公开 保密 _____ 年

作者签名: _____ 导师签名: _____

签字日期: _____ 签字日期: _____

摘 要

光网络虚拟借助基础设施即服务的概念提供了一种允许多个逻辑光网络共享物理基础设施的解决方案。本文中，我们集中考虑跨数据中心弹性光网络环境下的透明虚拟网络映射问题。首先，我们为这个问题提出了一个新的基于经典的 0-1 多商品流问题的整数线性规划模型。然后，为了将底层光纤链路的频谱资源使用情况和连续性的考虑整合到节点映射，我们设计一种分层辅助图的方法。该方法根据虚拟光网络的带宽需求以及底层光网络中每一根光纤链路的频谱使用情况，将底层光网络分解成一个个分层辅助图。利用分层辅助图，我们提出一种新的透明虚拟弹性光网络映射算法，该算法通过分层辅助图实现了节点映射和链路映射的协作。通过大量的仿真，我们证明了该算法可以达到对比算法低得多的阻塞率。最后，为了将理论工作推向实际应用，我们通过拓展网络虚拟化平台 OpenVirteX 设计并实现了一种新的弹性光网络虚拟化平台。通过概念验证试验，我们验证了我们的虚拟化平台确实能够实现弹性光网络虚拟化。然后，通过一系列性能测试试验，我们发现在普通 PC 下运行该虚拟化平台，即使在底层弹性光网络有 50 个节点，虚拟光网络有 10 个节点的情况，我们的虚拟化平台能够保证在 0.6 秒内完成对用户虚拟网络的创建，22 秒内完成用户虚拟网络的启动，14 秒内完成用户虚拟网络的停止，1.4 秒内完成用户虚拟网络的删除。

关键词： 弹性光网络，网络虚拟化，虚拟光网络映射，软件定义网络，OpenFlow，分层辅助图

ABSTRACT

Leveraging the concept of infrastructure as a service (IaaS), optical network virtualization provides a solution to sharing the physical infrastructures among different logical optical networks. In this paper, we focus on transparent virtual optical network embedding (VONE) over the inter-data-center elastic optical infrastructures. First of all, we propose a novel integer linear programming (ILP) model, which is based on the classic all-or-nothing multi-commodity flow problems. Then, to integrate the consideration of the substrate fiber links' spectral continuity and consecutiveness into the node mapping, we design a layered-auxiliary-graph (LAG) approach, by which we decompose the substrate optical network (SON) into several layered graphs according to the bandwidth requirement of the virtual optical network request (VONR) and the spectral usage of all the SFLs in the SON. Taking the advantages of LAG, we then propose an efficient heuristic algorithm, which achieves coordinated node and link mapping. Through extensive numerical simulations, we demonstrate that our proposed VONE algorithm achieves much lower blocking probability than benchmark algorithms. Finally, to put our VONE algorithm into practice, we also propose and implement an elastic optical network virtualization platform by extending the software-defined networking based network virtualization platform, OpenVirteX (OVX). Through a proof-of-concept experiment, we demonstrate that our platform can achieve virtualization of elastic optical networks. We also test the performance of our elastic optical network virtualization platform by a series of experiments, which indicates that by operating our network virtualization platform on PCs, even when the SON has a scale of 50 nodes and the virtual optical networks have a scale of 10 nodes, the platform could still finish creating of user's virtual network in 0.6 seconds, starting the virtual networks in 22 seconds, stopping the virtual network in 14 seconds and deleting the virtual network in 1.4 seconds.

Keywords: Elastic optical network, Network virtualization, Virtual optical network embedding, software-defined networking (SDN), OpenFlow, Layered-auxiliary-graph (LAG)

摘 要	I
ABSTRACT	III
目 录	V
表格索引	VII
插图索引	X
算法索引	XI
主要符号对照表	XIII
第一章 绪论	1
1.1 研究背景	1
1.1.1 弹性光网络的产生背景与主要优势	1
1.1.2 网络虚拟化的产生背景与虚拟网络映射	3
1.2 研究现状	6
1.2.1 虚拟网络映射	6
1.2.2 网络虚拟化平台	8
1.3 论文主要内容与论文结构	9
1.3.1 论文主要内容	9
1.3.2 论文结构	10
第二章 理论模型	11
2.1 网络模型	11
2.1.1 底层光网络	11
2.1.2 透明虚拟光网络请求	12
2.2 问题描述与整数线性规划模型	13
2.2.1 问题描述	13
2.2.2 整数线性规划模型	14
第三章 算法设计	17
3.1 分成辅助图构建	17
3.2 节点映射	17
3.3 链路映射	19

第四章 仿真评估	21
4.1 仿真参数	21
4.2 对比算法	22
4.3 性能评估参数	24
4.4 仿真结果以及结果分析	26
4.4.1 DT 拓扑下仿真结果	26
4.4.2 随机拓扑下的仿真结果	28
第五章 试验平台	31
5.1 系统架构	31
5.2 协议拓展	33
5.3 主要功能模块	35
5.3.1 虚拟软件定义弹性光网络管理系统	35
5.3.2 虚拟弹性光网络映射器	37
5.3.3 拓展的 OpenVirteX	38
5.4 试验结果	40
5.4.1 试验环境	40
5.4.2 系统功能测试	41
5.4.3 系统性能测试	50
第六章 总结与未来工作展望	53
6.1 总结	53
6.2 未来工作展望	53
参考文献	55
致 谢	63
在读期间发表的学术论文与取得的研究成果	65

表格索引

4.1	主要仿真参数的值或者取值范围	23
4.2	不同算法单个解的质量对比（底层光网络资源利用率：1%-10%）	26
4.3	不同算法单个解的质量对比（底层光网络资源利用率：11%-20%）	26
5.1	虚拟软件定义弹性光网络管理系统抓包结果详解	44
5.2	弹性光网络虚拟化平台四种主要操作所需时间（单位：毫秒） . .	51

1.1	WDM v.s. EON	2
1.2	透明弹性光网络中 RMLSA 示意	4
1.3	网络虚拟化示意	5
2.1	透明虚拟光网络映射示意	11
2.2	极大连续频谱块实例	12
3.1	分层辅助图构建示意	18
3.2	基于分层资源容量的节点映射算法实例	19
4.1	Deutsche Telecom (DT) 拓扑 (链路上数字表示其长度, 单位: km)	21
4.2	随机拓扑 (链路长度: 100 km)	22
4.3	DT 拓扑下的阻塞率	27
4.4	DT 拓扑下的平均最长光路径长度	28
4.5	随机拓扑下的阻塞率	28
4.6	随机拓扑下的平均最长光路径长度	29
5.1	系统框架	31
5.2	虚拟化系统工作流程实例	32
5.3	支持弹性光网络的 OF 的工作原理: (a) 主机 A 到主机 B 的光路 (b) 对应交换机上的流表信息	34
5.4	虚拟软件定义弹性光网络管理系统新建页面	36
5.5	虚拟软件定义弹性光网络管理系统用户中心页面	37
5.6	(a) 通过 E-OVX, 底层物理网络 (橙色) 与虚拟网络 (蓝色) 从逻辑上被分离开, 同时, 不同的虚拟网络在逻辑上也是隔离的, 每个用户只能“看到”自己的虚拟网络。虚拟网络 vnet 1 和 vnet 2 的拓扑信息以及通过 API 配置的它们与物理网络之间的映射都被存储在 Global Map (绿色) 中 ^[104] (b) 在 Global Map 中, 虚拟网络与物理网络之间的分割通过 OpenFlow 消息的虚拟化以及解虚拟化实现 ^[104]	38
5.7	E-OVX 地址改写机制实例 ^[90]	39
5.8	实验环境 (实线: 物理机器, 虚线: 虚拟机)	42
5.9	系统功能验证试验中使用的底层网络拓扑 (其中红色数字表示端口号)	43
5.10	系统功能验证试验中使用的虚拟网络拓扑	43
5.11	虚拟软件定义弹性光网络管理系统上 Wireshark 抓包截图	45

5.12 主机 h_1 上 Wireshark 抓包截图	45
5.13 主机 h_3 上 Wireshark 抓包截图	46
5.14 图 5.10 所示虚拟软件定义网络的映射关系	47
5.15 E-OVX 上 Wireshark 抓包截图	47
5.16 POX 发送给 E-OVX 的 FlowMod 消息	48
5.17 E-OVX 发送给底层网络的 FlowMod 消息	48
5.18 实验过程中流表信息	49
5.19 弹性光网络虚拟化平台四种主要操作所需时间的累积概率分布图	52

算法索引

3.1	分层辅助图的构建	17
3.2	基于分层资源容量的节点映射	18
3.3	基于分层辅助图的链路映射	20
3.4	<i>LRC-LSP</i> 算法	20
4.1	基于资源容量的节点映射	23
4.2	<i>RC-(SP-FF)</i> 算法	24
4.3	<i>RC-LSP</i> 算法	25

主要符号对照表

API	应用程序编程接口
ATM	异步传输模式 (Asynchronous Transfer Mode)
BPSK	二相相移键控 (Binary Phase Shift Keying)
BV-T	可变波长收发器 (Bandwidth-Variable Transponder)
BV-WSS	可变带宽波长选择交换机 (Bandwidth-Variable Wavelength Selective Switch)
CPU	中央处理器 (Central Processing Unit)
CSB	连续频谱块 (Contiguous Slot-Block)
DT	Deutsche Telecom
E-OVX	拓展的 OpenVirteX (Extended OpenVirteX)
EON	弹性光网络 (Elastic Optical Network)
FS	频谱带宽槽 (Frequency Slot)
InP	基础设施提供商 (Infrastructure Provider)
LAG	分层辅助图 (Layered Auxiliary Graph)
LLDP	链路层发现协议 (Link Layer Discovery Protocol)
LRC-LSP	基于分层资源容量的节点映射和分层辅助图的链路映射的透明虚拟弹性光网映射算法
MCSB	极大连续频谱带宽块 (Maximal Contiguous Slot-Block)
NVP	网络虚拟化平台 (Network Virtualization Platform)
OEO	光电光转换 (Optical-Electrical-Optical)
OF-AG	OpenFlow 代理 (OpenFlow Agent)
O-OFDM	正交频分复用 (Optical Orthogonal Frequency Division Multiplexing)
OVX	OpenVirteX
PC	个人电脑 (Personal Computer)
PLI	物理层损伤 (Physical Layer Impairment)
QPSK	四相相移键控 (Quaternary Phase Shift Keying)
QoS	服务质量 (Quality of Service)
RC-LSP	基于资源容量和分层辅助图的链路映射的透明虚拟弹性光网映射算法
RC-(SP-FF)	基于资源容量和最短路径 First-Fit 的链路映射的透明虚拟弹性光网络映射算法
RMLSA	路由、调制与频谱分配问题 (Routing, Modulation Level, and Spectrum Assignment)

VN	虚拟节点 (Virtual Node)
VNE	网络虚拟化 (Virtual Network Embedding)
VOL	虚拟光链路 (Virtual Optical Link)
VPN	虚拟专用网 (Virtual Private Network)
RSA	路由和频谱分配 (Routing and Spectrum Assignment)
RWA	路由与波长分配 (Routing and Wavelength Assignment)
SD-EON	软件定义弹性光网络 (Software-Defined Elastic Optical Networking)
SDN	软件定义网络 (Software-Defined Networking)
SFL	底层光纤链路 (Substrate Fiber Links)
SN	底层节点 (Substrate Node)
SP	服务提供商 (Service Provider)
VEONE	虚拟弹性光网络映射器 (Virtual Elastic Optical Network Embedder)
WDM	波分复用 (Wavelength Division Multiplexing)
vSD-EONMS	虚拟软件定义弹性光网络管理系统 (Virtual Software-Defined Elastic Optical Network Management System)

第一章 绪论

近年来, 云计算^[1]取得了飞速发展, Amazon, Google, Microsoft, IBM, 阿里, 百度, 新浪等国内外 IT 企业纷纷将自己的服务逐步转移到云端^[2-8], 云服务逐渐成为人们日常生活的必需品。云服务的崛起极大地推动了数据中心网络 (Data-Center Networks, DCNs) 的发展^[9]。为了提供更好的用户体验 (Quality of Experience, QoE), 同时也为了保证云服务的高有效性, 云服务提供商开始在各地部署数据中心, 使用地理上分散的数据中心 (Geographically Distributed Data Centers, Geo-DCs) 来承载自己的云服务。连接这些数据中心的网络, 我们称为跨数据中心网络 (Inter-Data-Center Networks, Inter-DCNs), 跨数据中心网络中的负载具有高峰值吞吐量、高突发性等特点^[10,11], 这使得如何建立和管理跨数据中心网络变得颇具挑战。幸运的是, 光网络, 特别是弹性光网络^[12]具有高带宽容量, 低时延, 低功耗等优点, 成为了构建跨数据中心网络的理想物理基础设施^[13]。

与此同时, 网络虚拟化^[14]作为云计算的核心支持技术之一, 也随着云计算的发展受到学术界与工业界的高度重视。其中最具挑战的难题——虚拟网络映射问题——更是成为了近几年网络领域最为热门的研究课题之一。然而, 现存大部分对于虚拟网络映射的研究主要集中在 2/3 层网络, 而有关虚拟光网络映射, 特别是虚拟弹性光网络映射的研究相对缺乏。因此, 本文以跨数据中心弹性光网络为背景, 从理论模型、算法设计以及平台搭建等方面对虚拟弹性光网络映射问题展开了深入研究。

本章首先详细阐述了虚拟弹性光网络映射问题的研究背景和研究现状, 然后, 简短总结了本文的主要工作以及相关贡献, 最后, 列出了本文剩余部分的组织结构。

1.1 研究背景

1.1.1 弹性光网络的产生背景与主要优势

近年来, 互联网中呈指数增长的带宽需求极大地推动了高带宽, 可拓展性强的通信和网络技术的发展。其中, 光纤通信网络以其巨大的带宽容量受到学术界与工业界的高度重视, 被认为是未来网络体系中核心与骨干网中不可缺少的组成成分。然而, 现有光通信技术只能实现对光纤带宽资源相对粗糙的分配与调度, 因此, 它在应对高度动态的互联网流量方面的表现并不尽人意。为了克服现有光网络的这些缺陷, 弹性光网络 (Elastic Optical Networks, EONs)^[12,15,16]作为一种新兴的光网络技术应运而生。相比于传统的波分复用 (Wavelength Division Multiplexing, WDM) 光网络^[17], 弹性光网络具有更加精细的波长粒度。WDM 通常具有 50 GHz 或者 100 GHz 的波长粒度, 而弹性光网络的频谱资

源粒度能够到达 12.5 GHz 甚至更低，另外，弹性光网络通过可变波长收发器 (Bandwidth-Variable Transponders, BV-Ts)^[18] 采用连续频谱带宽槽的方式来实现对频谱资源的分配，因此，弹性光网络可以实现对频谱资源更加灵活的分配与调度，为每一个连接请求分配恰好合适的带宽资源，达到子波长粒度，从而，提高对网络频谱资源的利用率。此外，弹性光网络还可以根据传输距离等因素灵活地选择调制级别^[19]。图1.1形象地展示了弹性光网络的上述优势，从图中我们可以看到传统的 WDM 光网络在进行频谱资源分配时是基于一个个离散的波长的，不管连接请求需要多少带宽资源，他都只能为其分配整数个波长，由于其带宽粒度相对粗糙，因此，很可能会造成大量的带宽资源浪费。弹性光网络则拥有小得多的带宽粒度，同时，它还具有灵活的调制级别选择，如图1.1(b)所示，在服务一个需求 50 Gb/s 的请求时（这里假设每一个带宽槽在采用 BPSK 调制的情况下，其容量是 12.5 Gb/s），根据传输距离的不同，它可以采用灵活地带宽分配方案与调度，从而，实现带宽资源的高效利用。

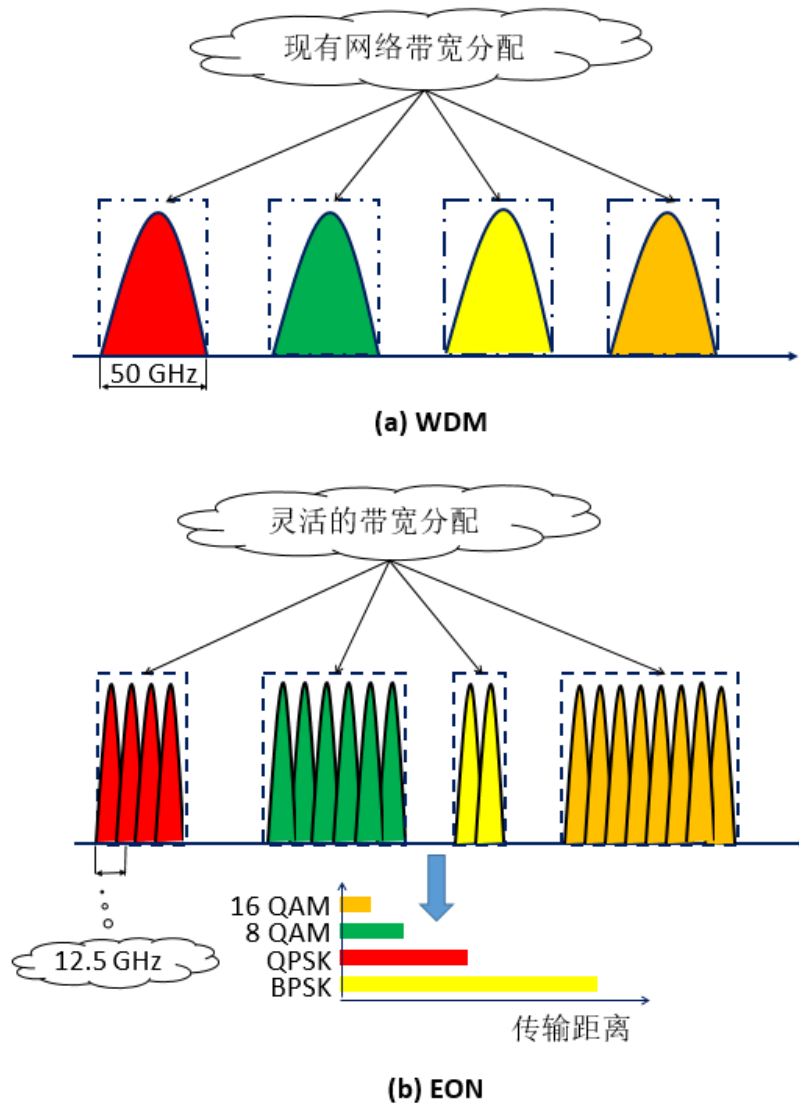


图 1.1: WDM v.s. EON

当然，弹性光网络在带来如此多好处的同时，也带来了新的挑战。它“弹性”的本质也使得网络资源的分配与调度变得更加复杂。具体而言，它使得传统 WDM 光网络中的路由与波长分配问题（Routing and Wavelength Assignment, RWA）^[20] 变成了路由、调制与频谱分配问题（Routing, Modulation Level, and Spectrum Assignment, RMLSA）^[21,22]。由于目前弹性光网络中的频谱转换技术（Spectrum Conversation）^[23] 尚不成熟，而光电光转换（Optical-Electrical-Optical, OEO）^[24] 能耗非常大，因此，目前针对弹性光网络的研究主要集中在透明弹性光网络^[25]。在透明弹性光网络中，RMLSA 问题需要满足如下几个独特的限制条件：

1. 频谱连续性约束：同一条光路（lightpath）的不同光链路上分配的频谱资源必须位于同一频谱段；
2. 频谱邻接性约束：分配给同一条光路的频谱资源在频域必须是连续的；
3. 频谱不重叠性约束：不同光路在有公共链路时，分配在这些光路的频谱在频域不能重叠；

图1.2展示了透明弹性光网络中 RMLSA 的一个例子，图1.2(a)展示了三个请求的带宽需求量以及对应光路，图1.2(b)展示了它们的频谱资源分配情况。在本例中，我们假设在采用 BPSK 调制级别时，每一个频谱带宽槽的带宽容量是 12.5 Gb/s，这里三个请求都选用了 BPSK 的调制级别。我们可以看到，由于频谱连续性约束，分配给三个请求对应光路的所有光链路上都分别采用了同一频谱段的频谱资源。又由于频谱邻接性约束，分配给每一个请求的频谱资源在频域上都是连续的，例如，请求 3 的光路就使用了频谱带宽槽 3,4,5。最后，我们可以看到由于请求 1 和请求 3 同时用到了光链路 (v_1, v_2) ，因此，分配给它们的频谱资源在频域上不能重叠*。

1.1.2 网络虚拟化的产生背景与虚拟网络映射

作为克服互联网“僵化”^[26-28]的有力手段之一，同时，也作为云计算的重要支持技术之一^[13]，网络虚拟化在近年来获得了学术界和工业界的高度关注与重视^[14,29-32]。

所谓网络虚拟化^[33]，是指多个逻辑上隔离的虚拟网络共存在同一个底层（或物理）网络上共享其计算和网络等资源。其中，每一个虚拟网络由若干虚拟节点（如，虚拟路由器）以及连接这些节点的虚拟链路组成。图1.3是一个网络虚拟化的示意图，图中描述了两个虚拟网络，即虚拟网络 1 和虚拟网络 2 共同存在于同一个底层物理网络之上。网络虚拟化与目前应用相当广泛的虚拟专用网（Virtual Private Network, VPN）^[34]有些类似，虚拟专用网指的是在公用网络

*实际上，它们的频谱资源之间还需要一定的保护带宽（guard-band）^[12]。

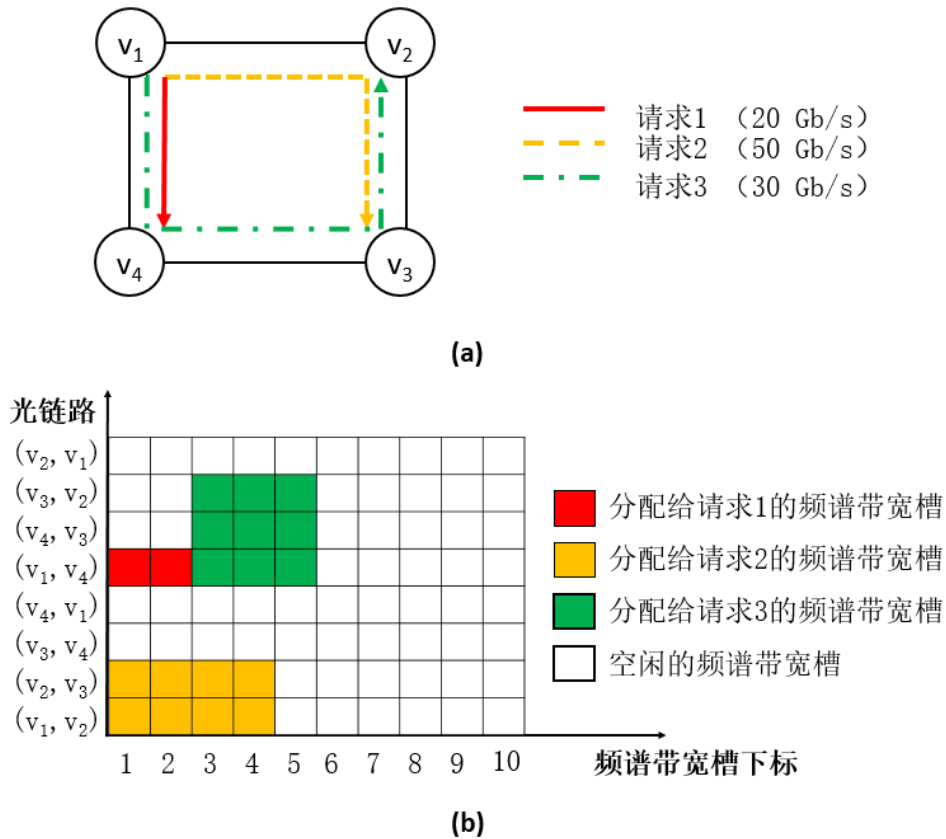


图 1.2: 透明弹性光网络中 RMLSA 示意

上建立的专用网络。之所以称其为虚拟网，主要是因为整个虚拟专用网的任意两个节点之间的连接并没有传统专网所需的端到端的物理链路，而是架构在公用网络服务商所提供的网络平台（如，Internet 等）之上的逻辑网络。但是，虚拟专用网存在如下缺陷。

- (1) 所有的虚拟网络采用相同的协议栈，阻碍了多种网络解决方案的共存。
- (2) 不同的虚拟网络之间没有实现真正的隔离。
- (3) 基础设施提供商的角色和服务提供商的角色没有完全分开。

而网络虚拟化则通过虚拟网络的独立可编程特性克服了上述缺点，也就是说每一个虚拟网络不必局限于 IP 协议，或者任何特定的协议，任何架构的网络都可以建设在虚拟网络之上。网络虚拟化的另一个好处就是，它可以实现服务提供商（Service Providers, SPs）和基础设施提供商（Infrastructure Providers, InPs）的彻底分离。其中，服务提供商通过虚拟网的形式租用设备提供商的各种资源为端用户提供相关服务。因此，网络虚拟化使得在不影响当前网络架构的条件下，引入和部署新的网络架构、协议以及应用成为可能，能够有效的支持和激发网络技术的创新。正因为如此，网络虚拟化技术已经被看做通向未来网络的可行途径和未来互联网必须具备的关键特性之一^[33]。

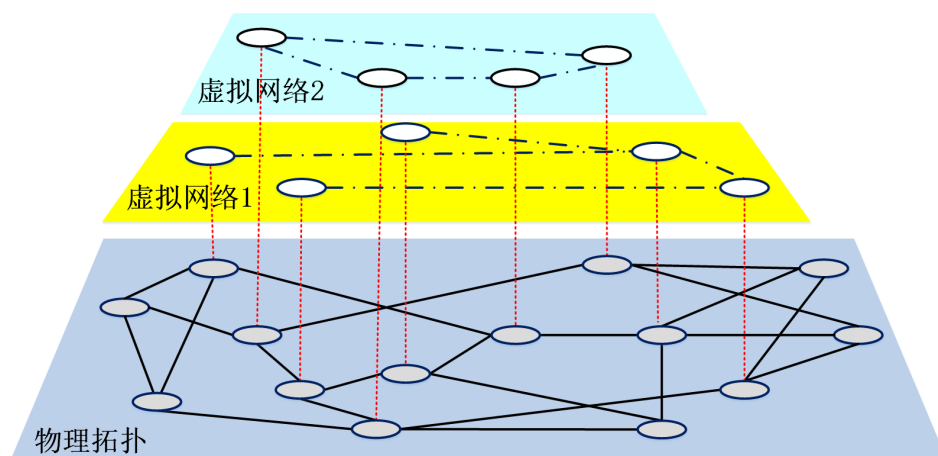


图 1.3: 网络虚拟化示意

与此同时，网络虚拟化也给基础设施提供商们带来了新的挑战，其中最大的难题便是虚拟网络映射问题^[35]，即如何高效地将底层网络的资源分配给每个虚拟网络。虚拟网络映射问题实际上包括了节点映射和链路映射两个子过程。其中，在节点映射中，我们需要为虚拟网络请求的每一个虚拟节点寻找一个合适的满足其资源需求等约束的物理节点，同时，还要保证来自同一个虚拟网络请求的不同虚拟节点不能映射在相同的物理节点上*。链路映射则需要将每一根虚拟链路映射至底层网络中的一条或多条物理路径，并保证这些路径能够满足对应虚拟链路的带宽等需求。在 2006 年的 INFOCOM 上，佐治亚理工学院的 Mostafa Ammar 教授领导的网络与通信课题组首次提出了虚拟网络映射问题^[38]，之后，便引起了国内外学术界的广泛关注^[39-59]。世界各国的学术研究者分别从理论模型、精确算法（即，能求解到最优解的算法）以及启发式算法等方面对该问题以及该问题的变种（如，多域虚拟网络映射，QoS 感知的虚拟网络映射，可生成的虚拟网络映射等等）展开了广泛而深入的研究。虽然，目前 2/3 层网络中的虚拟网络映射问题已经研究得相对透彻，传统光网络环境下的虚拟网络映射问题也有了比较丰富的研究成果。但是，弹性光网络下的虚拟网络映射问题的研究才刚刚起步。此外，弹性光网络的特殊性质使得 2/3 层网络和传统光网络下的虚拟网络映射问题的研究成果无法直接应用于弹性光网络。然而，正如本章第 1.1.1 小节所述，弹性光网络具有高带宽和支持更加灵活的带宽分配与调度，与此同时，它还具有传统光网络低延时、低功耗等优势，被认为是网络虚拟化技术的颇有潜力的物理设施^[13,60-62]。因此，开展弹性光网络下的虚拟网络映射问题的研究迫在眉睫。

*实际上，也有一些学者在研究虚拟节点可分割的虚拟节点映射^[36]，即一个虚拟节点可以同时映射至多个物理节点的节点映射，以及虚拟节点可合并的虚拟节点映射^[37]，即来自同一虚拟网络的多个虚拟节点准许映射至相同的物理节点。

1.2 研究现状

1.2.1 虚拟网络映射

1.2.1.1 2/3 层网络下的虚拟网络映射算法

作为网络虚拟化中一个极具挑战的问题——虚拟网络映射——在近年来俨然成为了网络研究者的宠儿。在国内外学术会议和期刊上，各种虚拟网络映射相关的研究成果被来自世界各地的网络研究者们争先恐后地提出。在本小节，我们将简单介绍其中比较有代表性的研究成果。关于近年来虚拟网络映射问题研究成果的系统介绍，请参见综述类文献^[35,63]。

2008 年，美国普林斯顿大学的 Yu 等人通过将每个节点的计算资源乘上该节点所有直连链路上的带宽资源之和定义为该节点的局部资源容量，提出了一个基于贪婪节点映射和 K 最短路径链路映射的虚拟网络映射算法^[39]。随后，研究者们发现通过在节点映射中适当考虑整个网络的拓扑信息能够提高虚拟网络映射算法的效果^[49,56,59,64]。其中，文献^[49,56,59]均采用了基于 PageRank^[65]的方式来考虑整个网络的资源信息，而文献^[64]则借助拓扑学中集中度 (centrality)^[66]的概念来实现全局资源信息的考虑。

以上这些研究成果都只考虑了没有位置限制的虚拟网络映射，实际中，可能有些应用需要对虚拟节点有位置限制。因此，加拿大滑铁卢大学的 Chowdhury 等人在文献^[51]中提出了一种基于增广图的方式将有位置限制的虚拟网络映射问题建模成一个带有特殊限制的多商品流问题^[67]，得到了一个节点映射和链路映射协作的混合整数线性规划映射模型。这也是虚拟网络映射问题第一个数学模型。考虑到混合整数规划的高时间复杂度问题，他们提出了采用将整数线性规划松弛成线性规划结合确定或者随机取整的方式来实现高效的启发式算法。在文献^[52]中，雅典国立科技大学的 Papagianni 等人和西班牙加泰罗尼亚理工大学的 Cervellò-Pastor 等人拓展了文献^[51]中的模型和算法，实现了对节点和链路上有多种资源以及其他 QoS 要求情况下的虚拟网络映射。但是，混合整数线性规划松弛成线性规划再结合取整的方式很容易导致非常差的解，甚至是非可行解^[68]。在文献^[53]中，美国佐治亚州立大学的 Hu 等人提出了一种基于路径的混合整数线性规划模型，同时，他们采用列生成^[69]的方式为虚拟网络映射算法提供了一个相对“高效”的求解最优解的方法。

考虑到底层网络的故障可能会同时导致多个虚拟网络受到严重影响，因此，许多学者开始了对可生存虚拟网络映射的研究。加拿大滑铁卢大学的 Rahman 等人在文献^[70]中首先为考虑底层网络单链路失效情形的可生成虚拟网络映射问题建立了混合整数线性规划模型，然后采用松弛的方法为该问题提供了一种有效的启发式算法。加拿大滑铁卢大学的 Jarray 等人在文献^[71]中针对底层网络节点失效以及多链路失效情形下的可生成虚拟网络映射提出了一种借助列生成技术^[72]的节点映射算法和基于预配置圈^[73]的链路映射算法。国内电子科学技术

大学的 Hongfang Yu 等人早在 2010 年就针对单域失效（即，地理位置相关的多个底层节点和链路同时失效）情形下的可生成虚拟网络映射问题展开了研究^[74]，他们首先为该问题建立了一个混合整数线性规划模型，然后，又提出了基于不受限分别优化以及受限逐步优化策略的两种启发式可生成虚拟网络映射算法。随着跨数据中心网络应用的逐步广泛，加上数据中心节点相对比较脆弱，因此，考虑跨数据中心网络中设备节点（即，数据中心）失效的可生成虚拟网络映射在近年来也逐步被重视起来。在文献^[75]中，国内电子科技大学大学的 Yu 等人针对单设备节点失效情形的可生成虚拟网络映射问题提出了基于共享的保护策略，并为之提出了混合整数线性规划模型。而在文献^[76]，国内北京大学的 Bingli Guo 等人提出了失效依赖的可生成虚拟网络映射方法。

1.2.1.2 光网络下的虚拟光网络映射算法

相比于 2/3 层网络，光网络下的虚拟网络映射算法的研究浪潮兴起得相对晚一些，而且已有的工作主要集中于固定栅格的波分复用网络下的虚拟光网络映射。

在文献^[77]中，通过假设底层网络每个节点上都装配有足够的波长转换器^[78]，美国加州大学戴维斯分校的 Zhang 等人为 WDM 网络下的虚拟网络映射建立了一个混合整数线性规划模型，并且提出了两个基于贪婪思想的启发式算法。随后，英国艾塞克斯大学的 Peng 等人在文献^[79]考虑物理层损伤，设计了 WDM 网络中物理层损伤感知的虚拟光网络映射算法。在文献^[80]，西班牙加泰罗尼亚理工大学的 Page 等人为 WDM 网络下透明和不透明虚拟光网络映射问题分别建立了整数线性规划模型。但是，在文献^[79,80]中，节点映射都被忽略了。

近年来，弹性光网络下特别是跨数据中心弹性光网络下的虚拟光网络映射问题也逐步引起研究者的重视。在文献^[81]中，西班牙加泰罗尼亚理工大学的 Page 等人考了弹性光网络下虚拟光网络映射中的链路映射。在文献^[37]中，通过准许来自同一个虚拟网络请求的多个虚拟节点映射到同一个物理节点，美国乔治华盛顿大学的 Zhao 等人为跨数据中心弹性光网络中的不透明虚拟光网络映射提出了一个整数线性规划模型和两个启发式算法。英国艾塞克斯大学的 Hammad 等人在文献^[82]中为弹性光网络下的不透明虚拟网络映射建立了整数线性规划模型，并为之提出了一个基于递归的启发式算法。美国佐治亚州立大学的 Shakya 等人和美国纽约州立大学水牛城分校的 Zilong Ye 等人合作研究了弹性光网络下不透明虚拟光网络映射和重配置问题^[83]，他们提出了对齐和连续性感知的虚拟光网络映射算法以及相对连续丢失感知和不对齐感知的虚拟光网络重配置算法。

由于光网络中光纤带宽巨大，底层网络故障带来的损失也更大。因此，可生成虚拟光网络映射问题也逐渐成为一个热门的研究话题。在文献^[84]中，国内中山大学的 Bowen Chen 等人和北京邮电大学以及美国德克萨斯大学达拉斯分校

合作，共同研究了弹性光网络下的可生成虚拟光网络映射问题，他们以最小化网络资源消耗为目标，为该问题建立了一个整数线性规划模型，并提出了最大带宽需求虚拟链路映射到最短距离底层光路的启发式映射算法。美国德克萨斯大学达拉斯分校的 Weisheng Xie 等人研究了弹性光网络下损伤感知的可生成虚拟光网络映射问题^[85]，他们以最小化收发器、中继器以及底层共享资源的消耗为目标，提出了单链路失效下可生成的虚拟光网络映射的整数线性规划模型和启发式算法。

1.2.1.3 小结

综上所述，2/3 层网络环境下的虚拟网络映射问题的研究已经相对成熟，传统光网络环境下的虚拟光网络映射问题的研究工作也比较完善，但是，弹性光网络环境下的虚拟光网络映射，特别是透明虚拟光网络映射问题的研究还比较缺乏。现存的关于弹性光网络环境下的虚拟光网络映射问题的研究工作大部分都没有考虑节点映射，即便是那些考虑了节点映射的研究工作也主要集中在对链路映射的改进，很少有研究考虑如何在弹性光网络环境下的虚拟光网络映射中从弹性光网络的特性出发来实现节点映射和链路映射的协作。因此，本文以实现节点映射与链路映射的协作为目标对弹性光网络环境下的透明虚拟光网络映射问题展开了深入研究。

1.2.2 网络虚拟化平台

在本章第1.2.1小节，我们介绍了大量的虚拟网络映射算法，然而，如果没有相关的技术去实现网络虚拟化，这些算法也不会有用武之地。近年来，兴起的软件定义网络（Software-Defined Networking, SDN）^[86]，旨在增强网络的可编程性，使网络做到应用感知，将网络的数据面和控制面解耦合，采用集中控制的方式来实现对路由、交换和网络的管理与控制，为网络虚拟化提供了一种实现方案。特别是 OpenFlow^[87] 的提出，加快了网络虚拟化从理论走向了实际的步伐。近年来，许多基于软件定义网络架构的网络虚拟化平台被纷纷提出，接下来我们将介绍其中比较有代表性的几个。

2009 年，美国斯坦福大学的开放网络课题组实现了第一个基于软件定义网络架构的网络虚拟化平台——Flowvisor^[88]。Flowvisor 采用分片技术将一个底层 OpenFlow 网络“切割”成多个资源池（也称分片）。每一个分片由底层网络的部分交换机和链路构成，可以有自己的网络控制器。但是，由于在 Flowvisor 体系下多个分片共享相同的流表空间和地址空间，因此，每一个分片无法拥有自己独立的地址空间，另外，由于 Flowvisor 基于分片技术，因此，它的每一个分片只能是底层网络的一个子集，而不能是任意拓扑。

2013 年，美国 Elysium Digital 的 Drutskoy、美国科罗拉多大学的 Keller 以及美国斯坦福大学的 Rexford 共同提出了 FlowN^[89] 网络虚拟化平台。FlowN 采

用数据库技术来维护虚拟网络到物理网络的映射。因此，每一个虚拟网络可以有自己的地址空间、拓扑以及控制逻辑。但是，由于 FlowN 采用共享控制器的架构，需要将每一个虚拟网络的控制逻辑映射到共享的控制器，因此，用户对虚拟网络的控制丧失了一定的灵活性。

2014 年，美国斯坦福大学的开放网络课题组借助他们开发 Flowvisor 的经验提出了一种新的网络虚拟化平台——OpenVirteX^[90]。OpenVirteX 通过虚拟网络和物理网络之间的轻耦合实现了地址虚拟化、拓扑虚拟化以及控制功能虚拟化使得每一个虚拟网络可以任意指定自己的地址空间和拓扑。但是，由于 OpenVirteX 基于 OF 1.0^[91] 开发，因此，很难实现定量的带宽隔离。

2015 年，德国慕尼黑工业大学的 Blenk 等人提出了一个新的基于软件定义网络架构的网络虚拟化平台——HyperFlex^[92]。HyperFlex 通过将 SDN 网络的监督系统分解成不同的功能模块，具有高效、灵活、可拓展性强等优点。同时，除了数据面的虚拟化，HyperFlex 首次全面考虑了控制面的虚拟化，通过功能模块结合接纳策略，HyperFlex 有效的避免了由于某些虚拟软件定义网络大量发送 OF 消息而导致监督系统过载现象的发生。

然而，以上网络虚拟化平台都不支持弹性光网络。因此，本文选择了通过拓展 OpenVirteX 来实现跨数据中心弹性光网络环境下的网络虚拟化平台。

1.3 论文主要内容与论文结构

1.3.1 论文主要内容

本文对跨数据中心弹性光网络环境下的透明虚拟光网络映射问题展开了深入研究。首先，借助经典的 0-1 多商品流问题（All-or-Nothing Multi-commodity Flow Problem），我们为透明虚拟弹性光网络映射问题建立一个基于多商品流的整数线性规划模型。然后，我们又为该问题提出了一个基于分层辅助图的高效启发式算法。该算法首先根据透明虚拟光网络请求的带宽需求和底层网络的频谱使用情况，为底层网络建立分层辅助图。利用分层辅助图，我们设计了一种新的基于分层资源容量的节点映射和分层路由与频谱分配的链路映射的透明虚拟光网络映射算法，实现了节点映射和链路映射的协作，同时，也实现了路由与频谱分配的整合。通过大量的数值仿真，我们证明了该算法相对于对比算法能够使用更少的资源去服务更多的虚拟光网络请求。最后，为了将理论工作推向实践，我们还搭建了一个基于软件定义网络的网络虚拟化平台。试验结果表明我们的虚拟化平台确实实现了对弹性光网虚拟化的支持，而且，在普通 PC 下运行该虚拟化平台，即使在底层弹性光网络有 50 个节点，虚拟光网络有 10 个节点的情况，我们的虚拟化平台能够保证在 0.6 秒内完成对用户虚拟网络的创建，22 秒内完成用户虚拟网络的启动，14 秒内完成用户虚拟网络的停止，1.4 秒内完成用户虚拟网络的删除。

1.3.2 论文结构

本文剩余本分的结构如下。在第二章，我们对跨数据中心弹性光网络环境下的透明虚拟光网络映射问题进行了详尽描述，并且为该问题建立了整数线性规划模型。第三描述了本文提出的基于分层资源容量和分层路由与频谱分配的透明虚拟光网络映射算法。然后，第四章是对本文提出算法的仿真评估。在第五章，我们详细描述了我们的试验平台以及试验评估。最后，第六章是对本文的总结以及对未来工作的展望。

第二章 理论模型

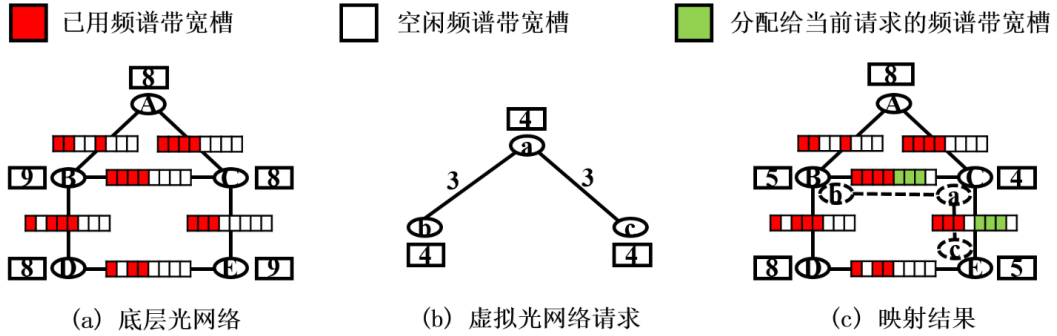


图 2.1: 透明虚拟光网络映射示意

2.1 网络模型

2.1.1 底层光网络

在本文中，我们将底层跨数据中心弹性光网络（本文后面章节在不引起混淆的情况下，将直接简称“底层光网络”）建模成一个无向图，记作 $G^s(V^s, E^s)$ ，其中 V^s 是底层节点（Substrate Nodes, SNs）的集合，而 E^s 则是底层光纤链路（Substrate Fiber Links, SFLs）的集合。每一个底层节点 $v^s \in V^s$ 实际上包含了一个光交换机以及直连在其上的一个数据中心，其上有一定的计算资源容量，记作 $c_{v^s}^s$ 。同时，每一根底层光纤链路 $e^s \in E^s$ 的带宽容量是 B^s 个频谱带宽槽。图2.1(a)展示了一个底层光网络，其中每个底层节点周围方框内的数字代表了其计算资源容量（即，该节点的数据中心中的计算资源容量），每根底层光纤链路上的矩形条则刻画了该光纤链路上的频谱资源使用情况，其中红色的表示已经使用的频谱带宽槽，白色表示空闲的频谱带宽槽。我们采用 P^s 来表示底层网络中所有无环路径的集合，采用 $P_{u^s, v^s}^s \subseteq P^s$, $u^s, v^s \in V^s$, $u^s \neq v^s$ 表示底层光网络中从节点 u^s 到节点 v^s 的无环路径的集合。为了方便描述每根底层光纤链路的频谱使用情况，我们引入了一个长度为 B^s 的 $0-1$ 向量 $b_{e^s}^s$ ，其中 $b_{e^s}^s[j] = 1$ 表示光纤链路 e^s 上的第 j 个频谱带宽槽已经被使用，否则 $b_{e^s}^s[j] = 0$ 。根据 $b_{e^s}^s$ ，我们进一步获得表示频谱使用情况的两个重要参数，即 $w_{e^s}^s$ 和 $z_{e^s}^s$ 。它们分别表示光纤链路 e^s 上所有极大连续频谱带宽块（Maximal Contiguous Slot-Blocks, MCSBs）的起始带宽槽下标和终止带宽槽下标。图2.2展示了极大连续频谱块的一个实例。以下是连续频谱块以及极大连续频谱块的定义。

定义 2.1.1 (连续频谱块). 给定一根光纤链路 e ，其上的频谱带宽槽集合和频谱使用情况向量分别是 F_e 和 b_e ，其上的连续频谱块（Contiguous Slot-Blocks, CSBs）是指一个或者多个在频域连续并且空闲的频谱带宽槽构成的集合，即，如果

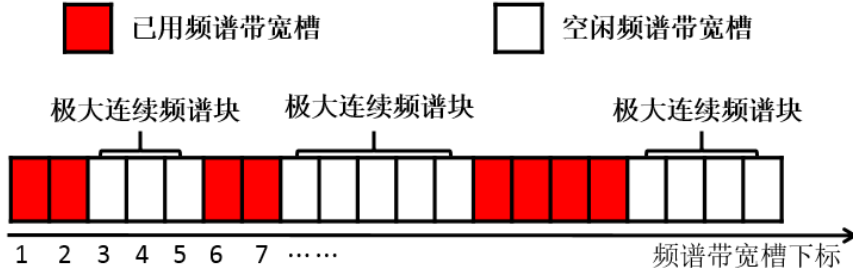


图 2.2: 极大连续频谱块实例

$C_e \subseteq F_e$ 是光纤链路 e 上的一个连续频谱块，那么，

$$F_e[j] \in C_e \Rightarrow b_e[j] = 0, \quad 1 \leq j \leq |F_e|, \quad (2.1)$$

$$\left. \begin{array}{l} F_e[i] \in C_e, \\ F_e[j] \in C_e, \end{array} \right\} \Rightarrow F_e[k] \in C_e, i \leq k \leq j, \quad 1 \leq i \leq j \leq |F_e|, \quad (2.2)$$

其中， $F_e[j]$ 表示第 j 个频谱带宽槽， $|F_e|$ 表示集合 F_e 的势，也就是光纤链路 e 上频谱带宽槽的总数。

定义 2.1.2 (极大连续频谱块). 给定一根光纤链路 e ，其上的频谱带宽槽集合和频谱使用情况向量分别是 F_e 和 b_e ，其上的极大连续频谱块是指频谱带宽槽个数不能再增加的连续频谱块，即，如果 $C_e \subseteq F_e$ 是光纤链路 e 上的一个极大连续频谱块，那么，

$$\left. \begin{array}{l} F_e[j] \in C_e, \\ F_e[j+1] \notin C_e, \end{array} \right\} \Rightarrow b_e[j+1] = 1, \quad 1 \leq j \leq |F_e| - 1, \quad (2.3)$$

$$\left. \begin{array}{l} F_e[j] \in C_e, \\ F_e[j-1] \notin C_e, \end{array} \right\} \Rightarrow b_e[j-1] = 1, \quad 2 \leq j \leq |F_e|, \quad (2.4)$$

2.1.2 透明虚拟光网络请求

与底层光网络类似，一个透明虚拟光网络（在不引起混淆的情况下，本文后面章节将直接简称为“虚拟光网络”或者“虚拟网络”）请求也被建模成一个无向图 $G^r(V^r, E^r)$ ，其中 V^r 和 E^r 分别是虚拟节点（Virtual Node, VN）和虚拟光链路（Virtual Optical Link, VOL）构成的集合。每一个虚拟节点 $v^r \in V^r$ 需求一定的计算资源，记作 $c_{v^r}^r$ ，与此同时，每一根虚拟光链路需求一定的带宽资源，即 n^r （单位：频谱带宽槽个数）。为了简化问题，我们采用了类似于文献^[93]中的 WDM 光网络环境下的透明虚拟光网络映射模型，即我们要求透明虚拟光网络映射保证分配给每一个虚拟光网络的频谱带宽槽在频域位于同一频谱段。具体而言，就是分配给同一个虚拟光网络请求不同光链路的连续频谱块必须拥有相同的频谱带宽槽下标。图2.1(b)展示了一个透明虚拟光网络请求，其中每个虚拟节点周围方框内的数字表示该节点的计算资源需求量，每根虚拟链路上的数字表示该链路的带宽资源需求量。

2.2 问题描述与整数线性规划模型

2.2.1 问题描述

透明虚拟光网络映射包括如下两个子过程：1) 为每一个虚拟节点分配一个有足够计算资源的唯一的底层光网络节点（即，节点映射）；2) 为每一根虚拟光链路找到一条底层光路径，并在光路径上为其分配满足频谱连续性，频谱邻接性以及频谱不重叠性约束的频谱块（即，链路映射）。具体而言，节点映射和链路映射描述如下：

1. 节点映射：每一个虚拟节点 $v^r \in V^r$ 将被映射到一个唯一的底层物理节点 $v^s \in V^s$ ，采用数学语言，节点映射可以表示成如下映射函数 M_N ，

$$M_N(v^r) = v^s, \quad v^r \in V^r, v^s \in V^s, \quad (2.5)$$

满足，

- 映射约束，即，来自同一个虚拟光网络请求的不同的虚拟节点必须映射至不同的底层节点，采用数学语言描述如下，

$$M_N(v^{r,1}) = M_N(v^{r,2}) \Leftrightarrow v^{r,1} = v^{r,2}, \quad v^{r,1}, v^{r,2} \in V^r, \quad (2.6)$$

- 计算资源容量约束，即，每一个虚拟节点映射至的底层节点必须具有足够的可用计算资源，采用数学语言描述如下。

$$c_{v^r}^r \leq c_{v^s}^s, \quad v^s = M_N(v^r), \quad (2.7)$$

如图2.1(c)所示，图2.1(b)中虚拟请求的节点映射是 $\{a \rightarrow C, b \rightarrow B, c \rightarrow E\}$ 。

2. 链路映射：实际上是一个特殊的路由和频谱分配（Routing and Spectrum Assignment, RSA）问题^[16]。该子过程需要决定如何为每一条虚拟光链路路由（即，找到一条光路径）以及如何为其分配足够数量的连续频谱块，我们采用一个长度为 B^s 的 0-1 向量 b^r 来描述其频谱资源分配情况，其中 $b^r[j] = 1$ 表示第 j 个频谱带宽槽分配给了该请求，否则 $b^r[j] = 0$ 。链路映射可以表示成如下映射函数 M_L ，

$$M_L(e^r) = p^s, \quad e^r \in E^r, p^s \in P^s, \quad (2.8)$$

满足，

- 映射约束，即虚拟光链路 e^r 映射到的底层光路径 p^s 的源-目的节点必须是 e^r 的端节点映射到的底层节点，采用数学语言描述如下，

$$p^s \in P_{M_N(s_{e^r}^r), M_N(d_{e^r}^r)}^s \quad (2.9)$$

其中， $s_{e^r}^r/d_{e^r}^r$ 分别表示链路 e^r 的两个端节点。

- 带容量约束，即，分配给每一根虚拟光链路的频谱带宽槽个数等于其需求量，采用数学语言描述如下，

$$\sum_{j=1}^{B^s} b^r[j] = n^r, \quad (2.10)$$

- 频谱连续性和邻接性约束，即，分配给该虚拟光网络请求的每一根虚拟光链路的频谱带宽槽必须是处于相同频谱段的连续频谱块，如果我们采用 $E_r^s = \{e^s : e^s \in M_L(e^r), \forall e^r \in E^r\}$ 来表示该虚拟光网络请求使用的底层光纤链路的集合，采用 $F_{e^s}^r$ 表示该请求使用的底层光纤链路 $e^s \in E_r^s$ 上分配的频谱带宽槽集合，该约束采用数学语言描述如下，

$$F_{e^s}^r[j] \in F_{e^s}^r \Rightarrow b^r[j] = 1, \quad (2.11)$$

其中， $F_{e^s}^r$ 表示底层光纤链路 e^s 上的频谱带宽槽集合。

$$\{F_{e^s}^r[j] : F_{e^s}^r[j] \in F_{e^s}^r\} \text{是 } e^s \text{ 上的连续频谱块, } \forall e^s \in E_r^s, \quad (2.12)$$

其中， $F_{e^s}^r$ 表示底层光纤链路 e^s 上的频谱带宽槽集合。

- 频谱不重叠性约束，即，分配给该虚拟光网络请求的任意一根虚拟光链路的连续频谱块不能与网络中之前请求的发生重叠，采用数学语言描述如下，

$$\sum_{j=1}^{B^s} b^r[j] \cap \left(\bigcap_{e^r \in E^r} \bigcap_{e^s \in M_L(e^r)} b_{e^s}^s[j] \right) = 0, \quad (2.13)$$

如图2.1(c)所示，图2.1(b)中的虚拟网络请求的节点映射为 $\{(a, b) \rightarrow \{(C, B)\}, (a, c) \rightarrow \{(C, E)\}\}$ ，其中分配的频谱资源是第5–8个频谱带宽槽。

2.2.2 整数线性规划模型

在本节中，我们为透明虚拟光网络映射问题建立了基于多商品流^[67]的整数线性规划模型。该理论模型描述如下：

参数：

- $G^s(V^s, E^s)$: 底层物理光网络。
- $c_{v^s}^s$: 底层节点 $v^s \in V^s$ 的计算资源量。
- B^s : 每根底层光纤链路 $e^s \in E^s$ 上总的频谱宽槽个数。
- $w_{(u^s, v^s), k}^s$: 光链路 $e^s = (u^s, v^s)$ 上第 k 个极大连续频谱块的起始带宽槽下标。

- $z_{(u^s, v^s), k}^s$: 光链路 $e^s = (u^s, v^s)$ 上第 k 个极大连续频谱块的终止带宽槽下标。
- $G^r(V^r, E^r)$: 虚拟光网络请求。
- $s_{e^r}^r/d_{e^r}^r$: 虚拟光链路 $e^r \in E^r$ 端节点。
- $c_{v^r}^r$: 虚拟节点 $v^r \in V^r$ 的计算资源需求量。
- n^r : 每根虚拟光链路的带宽需求量 (单位: 频谱带宽槽个数)。

决策变量:

- ξ_{v^r, v^s} : 0-1 变量, 如果虚拟节点 v^r 被映射到底层节点 v^s , $\xi_{v^r, v^s} = 1$, 否则, $\xi_{v^r, v^s} = 0$ 。
- $f_{u^s, v^s}^{e^r}$: 0-1 变量, 如果虚拟光链路 $e^r \in E^r$, 映射到的底层光路径经过光链路 $e^s = (u^s, v^s)$, $f_{u^s, v^s}^{e^r} = 1$, 否则 $f_{u^s, v^s}^{e^r} = 0$ 。
- $\delta_{(u^s, v^s)}^{(k)}$: 0-1 变量, 如果底层光链路 $e^s = (u^s, v^s)$ 上的第 k 个极大连续频谱块中的某个连续频谱块被分配给当前虚拟光网络请求 $G^r(V^r, E^r)$, $\delta_{(u^s, v^s)}^{(k)} = 1$, 否则 $\delta_{(u^s, v^s)}^{(k)} = 0$ 。
- w : 整型变量, 表示分配给当前虚拟光网络 $G^r(V^r, E^r)$ 的连续频谱块的起始带宽槽下标。
- z : 整型变量, 表示分配给当前虚拟光网络 $G^r(V^r, E^r)$ 的连续频谱块的终止带宽槽下标。

目标函数:

$$\text{Minimize } w + B^s \cdot \sum_{e^r \in E^r} \sum_{(u^s, v^s) \in E^s} f_{u^s, v^s}^{e^r} \quad (2.14)$$

通过目标函数公式(2.14), 我们的首要目标是最小化分配给该请求的总的光链路的条数, 也就是最小化分配给该请求的频谱带宽槽总数, 其次我们还要最小化分配给该请求的连续频谱块的起始带宽槽下标。

约束条件:

1. 节点映射,

$$\sum_{v^s \in V^s} \xi_{v^r, v^s} = 1, \quad \forall v^r \in V^r, \quad (2.15)$$

$$\sum_{v^r \in V^r} \xi_{v^r, v^s} \leq 1, \quad \forall v^s \in V^s, \quad (2.16)$$

公式(2.15)-(2.16)保证了同一虚拟光网络请求中的每一个虚拟节点都被映射到唯一的底层节点。

$$\sum_{v^s \in V^s} \xi_{v^r, v^s} \cdot c_{v^s}^s \geq c_{v^r}^r, \quad \forall v^r \in V^r, \quad (2.17)$$

公式(2.17)保证了每一个被映射的底层节点都有足够的计算资源提供给相应虚拟节点。

2. 链路映射,

$$\sum_{(u^s, v^s) \in E^s} f_{u^s, v^s}^{e^r} - \sum_{(v^s, u^s) \in E^s} f_{v^s, u^s}^{e^r} = \xi_{s_{e^r}, u^s} - \xi_{d_{e^r}, u^s}, \quad \forall e^r \in E^r, \forall u^s \in V^s, \quad (2.18)$$

公式(2.18)是流守恒约束^[67]。它保证了对于除了映射了当前虚拟光链路端节点的底层节点之外的所有的底层节点的总的流入流的总量等于流出流总量。对于映射该虚拟光链路两个端节点的底层节点，一个节点的流入流总量比流出流总量多一个单位，而另一个节点的则少一个单位。

$$\sum_{e^r \in E^r} (f_{u^s, v^s}^{e^r} + f_{v^s, u^s}^{e^r}) \leq 1, \quad \forall (u^s, v^s) \in E^s, \quad (2.19)$$

公式(2.19)保证了对于每条虚拟光链路，它们映射到的底层光路径彼此没有共用链路。

$$z - w + 1 = n^r, \quad (2.20)$$

公式(2.20)保证分配给该虚拟光网络的连续频谱块的大小（即，其中包含的频谱带宽槽的个数）能够满足该虚拟光网络的带宽需求。

$$\sum_k \delta_{(u^s, v^s)}^{(k)} = \sum_{e^r \in E^r} (f_{u^s, v^s}^{e^r} + f_{v^s, u^s}^{e^r}), \quad \forall (u^s, v^s) \in E^s, \quad (2.21)$$

$$w \geq w_{(u^s, v^s), k}^s \cdot \delta_{(u^s, v^s)}^{(k)}, \quad \forall (u^s, v^s) \in E^s, \forall k, \quad (2.22)$$

$$z \leq (z_{(u^s, v^s), k}^s - B^s) \cdot \delta_{(u^s, v^s)}^{(k)} + B^s, \quad \forall (u^s, v^s) \in E^s, \forall k, \quad (2.23)$$

公式(2.21)-(2.23)保证了分配给 $G^r(V^r, E^r)$ 频谱带宽槽在每一根底层光链路上均落在一个极大连续频谱块中。

3. 变量定义域:

$$\xi_{v^r, v^s} \in \{0, 1\}, \quad \forall v^r \in V^r, v^s \in V^s, \quad (2.24)$$

$$f_{u^s, v^s}^{e^r}, \delta_{(u^s, v^s)}^{(k)} \in \{0, 1\}, \quad \forall (u^s, v^s) \in E^s, \forall e^r \in E^r, \forall k, \quad (2.25)$$

$$w, z \in [1, B^s], \quad (2.26)$$

公式(2.24)-(2.26)是对应变量的取值范围。

第三章 算法设计

在本章，我们将详细阐述我们提出的基于分层辅助图的透明虚拟光网络映射算法。首先，我们将介绍分层辅助图（Layered Auxiliary Graph, LAG）的构建方法，然后描述在分层辅助图基础上设计的节点映射和链路映射算法。

3.1 分成辅助图构建

输入: 底层光网络 $G^s(V^s, E^s)$, 虚拟光网络请求的带宽需求量 n^r , 分层辅助图的下标 k

输出: 第 k 层分层辅助图 G_k^{sub}

```

1  $G_k^{sub} = G^s$ ;
2 将  $G_k^{sub}$  中每一根链路的带宽都标记为  $n^r$ ;
3 for  $G^s$  中每一根链路  $e^s \in E^s$  do
4   | if  $\sum_{j=k}^{k+n^r-1} b_{e^s}^s[j] > 0$  then
5   |   | 从  $G_k^{sub}$  中移除链路  $e^s$ ;
6   | end
7 end

```

算法 3.1: 分层辅助图的构建

对于一个虚拟光网络请求 $G^r(V^r, E^r)$ ，我们根据其频谱资源需求量 n^r 以及底层光链路上的频谱使用情况将底层网络分解成一个个分层辅助图。具体而言，对于第 k 层，我们会检查底层光网络的每一根光链路上从第 k 到第 $k + n^r - 1$ 个频谱带宽槽构成的连续频谱块，如果该连续频谱块是可用的，我们就将其放入第 k 层辅助图。算法3.1给出了分层辅助图构建算法的伪代码。图3.1给出了对图2.1描绘的虚拟光网络映射实例构建的分层辅助图的前五层。下面以第五层分层辅助图为例来展示该辅助图构建的细节。由于图2.1(b)所示的虚拟光网络请求需要 3 个频谱带宽槽，因此，第五层应该检测每一个底层光链路上从第 5 到第 7 个频谱带宽槽构成的连续频谱块是否可用，从图2.1(a)我们很容易看出只有链路 (A, B) 和 (B, D) 上的是不可用的，因此它们不能出现在第五层分层辅助图，而其他链路均要放入该分成辅助图，于是，我们便得到了图3.1(e)所示的分层辅助图。

3.2 节点映射

在节点映射中，类似于文献^[39]中资源容量的概念，我们提出了分层资源容量的概念，其定义如下。

定义 3.2.1 (分层资源容量). 对于一个分层辅助图 $G_k^{sub}(V, E)$ ，节点 $v \in V$ 的分层资源容量定义如下：

$$h_v = c_v \cdot \beta_v, \quad \forall v \in V, \quad (3.1)$$

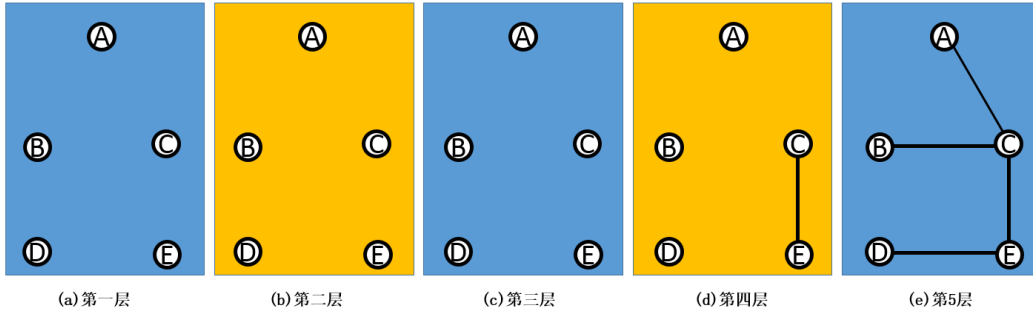


图 3.1: 分层辅助图构建示意

其中, c_v 和 β_v 分别是节点 $v \in V$ 的计算资源量和 v 在分层辅助图 G_k^{sub} 上的节点度。

输入: 第 k 层分层辅助图的第 j 个连通子图 $G_{k,j}^{sub}$, 虚拟光网络请求 G^r
 输出: 节点映射 M_N , 映射状态 F

- 1 对于 $G_{k,j}^{sub}$ 中的每一个节点 v^s 计算其分层资源容量和节点度 β_{v^s} ;
- 2 对于 G^r 中每一个 v^r 计算其节点度 β_{v^r} ;
- 3 按照节点度 β_{v^r} 从大到小对虚拟节点进行排序;
- 4 按照分层资源容量从大到小对 $G_{k,j}^{sub}$ 上的底层节点进行排序;
- 5 **for** 每一个虚拟节点 $v^r \in V^r$, 按节点度从大到小的顺序 **do**
- 6 $F = FAILED$;
- 7 **for** $G_{k,j}^{sub}$ 中每一个未标记的 v^s , 按分层资源容量从大到小的顺序 **do**
- 8 **if** $c_{v^s}^s \geq c_{v^r}^r$ **AND** $\beta_{v^s} \geq \beta_{v^r}$ **then**
- 9 $M_N(v^r) \leftarrow v^s$;
- 10 标记物理节点 v^s ;
- 11 $F = SUCCEEDED$;
- 12 **break**;
- 13 **end**
- 14 **end**
- 15 **if** $F = FAILED$ **then**
- 16 **return**(F);
- 17 **end**
- 18 **end**
- 19 **return**(M_N, F);

算法 3.2: 基于分层资源容量的节点映射

基于分层资源容量我们提出了如下节点映射算法。该算法基于贪婪的思想, 从第一层开始尝试映射直至找到能够成功映射的层或者遍历完所有的层。具体而言, 在某一层上, 首先, 计算每个虚拟节点的节点度, 然后计算该分层辅助图上每个节点的分层资源容量。接下来, 按照节点度从大到小的顺序, 并根据分层资源容量从大到小为每一个虚拟节点选择一个底层节点。算法3.2是该节点映射算法的伪代码。下面, 我们以一个实例来详细说明我们提出的节点映射算法

的工作原理。如图3.2所示，首先，我们构建第一层分层辅助图，然后尝试在其上的每一个连通子图上进行节点映射，发现无法映射成功，于是，我们开始尝试第二层，依然失败。以此类推，直至我们尝试到第五层，通过计算我们得到了各个底层节点的分层资源容量（见图3.2(e)），同时，由于虚拟节点 a 具有最大的节点度，因此，我们最先映射它，按照贪婪的思想，我们找到了分层资源容量最大的节点 C ，由于节点 C 能够满足节点 a 的计算资源需求，同时其节点度也大于 a 的节点度，于是我们将节点 a 映射到节点 C ，类似的，我们依次将节点 b 和 c 分别映射到节点 B 和 E ，从而得到了如图 2.1(c) 所示的节点映射结果。

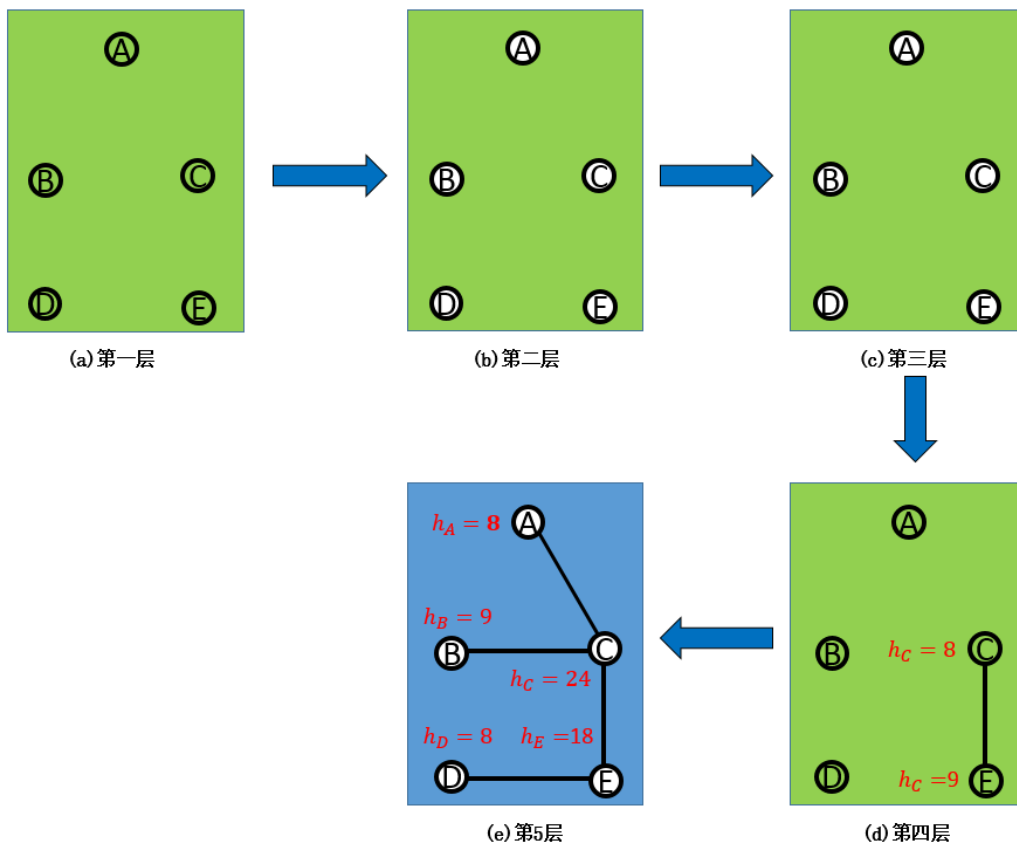


图 3.2: 基于分层资源容量的节点映射算法实例

3.3 链路映射

节点映射完成之后，我们开始进行链路映射。我们的链路映射是基于分层辅助图的最短路径映射。具体而言，对于每一根虚拟光链路，我们采用最短路径算法为其建立一条光路径，然后将该光路径用到的链路从对应分层辅助图上删除，然后，开始下一条虚拟光链路的映射，如果所有虚拟光链路都映射成功，那么映射成功，我们就为该虚拟光网络分配资源。否则，进入到下一层分层辅助图，重复上述过程。如果所有层均映射失败，那么该虚拟光网络请求就被阻塞。

算法3.4展示了我们提出的基于分层资源容量和分层最短路径的透明虚

输入: 第 k 个分层辅助图的第 j 个连通子图 $G_{k,j}^{sub}$, 虚拟光网络请求 G^r
 输出: 链路映射 M_L , 映射状态 F

```

1 for  $G^r$  中每一根虚拟光链路  $e^r = (u^r, v^r)$  do
2   获取节点  $u^r$  和  $v^r$  在  $G_{k,j}^{sub}$  中映射的底层物理节点  $u$  和  $v$ ;
3   在  $G_{k,j}^{sub}$  中采用最短路径算法找  $u$  到  $v$  的最短路径  $p$ ;
4   if 不存在从  $u$  到  $v$  的路径 then
5     | return( $F = FAILED$ );
6   end
7    $M_L(e^r) \leftarrow p$ ;
8   在  $G_{k,j}^{sub}$  中移除  $p$  上的所有光纤链路;
9 end
10 return( $M_L, F = SUCCEEDED$ );
    
```

算法 3.3: 基于分层辅助图的链路映射

拟光网络映射算法 (Layered-Resource Capacity and Layered-Shortest-Path based Transparent Virtual Optical Network Embedding Algorithm) 的伪代码, 我们将其记作 LRC-LSP 算法。

输入: 底层光网络 G^s , 虚拟光网络请求 G^r
 输出: 映射状态 F

```

1  $G_{temp}^s = G^s$ ;
2 for  $k = 1$  to  $B^s - n^r + 1$  do
3   采用算法3.1构建第  $k$  层分层辅助图  $G_k^{sub}$ ;
4   获取  $G_k^{sub}$  中节点个数多于  $G^r$  连通子图, 并记作  $G_{k,j}^{sub}$ ;
5   按照节点个数由大到小对  $G_{k,j}^{sub}$  进行排序;
6   for 每一个  $G_{k,j}^{sub}$ , 按照节点数从多到少 do
7     采用算法3.2进行节点映射;
8     if 节点映射成功 then
9       采用算法3.3 进行链路映射;
10      if 链路映射成功 then
11        | 为  $G^r$  分配资源;
12        | return( $F = SUCCEEDED$ );
13      end
14    end
15  end
16 end
17 标记  $G^r$  被阻塞;
18  $G^s = G_{temp}^s$ ;
19 return( $F = FAILED$ );
    
```

算法 3.4: LRC-LSP 算法

第四章 仿真评估

在本章中，我们设计了大量的仿真从单个解的质量以及阻塞率等多个角度对我们在第三章提出的启发式算法进行了评估。首先，我们在一个 14 点的实际物理拓扑作为底层光网络的条件下，通过比较我们的算法的解和最优解的间隙来衡量我们的算法的单个解的质量，考虑到虚拟网络映射作为一个在线问题，单个解的值量并不能完全衡量某个算法的性能，我们又设计了动态仿真分别在一个 14 点实际拓扑和一个 50 点随机作为底层网络的条件下从阻塞率和平均最长光路径的长度两个方面来进一步衡量我们的算法的性能。

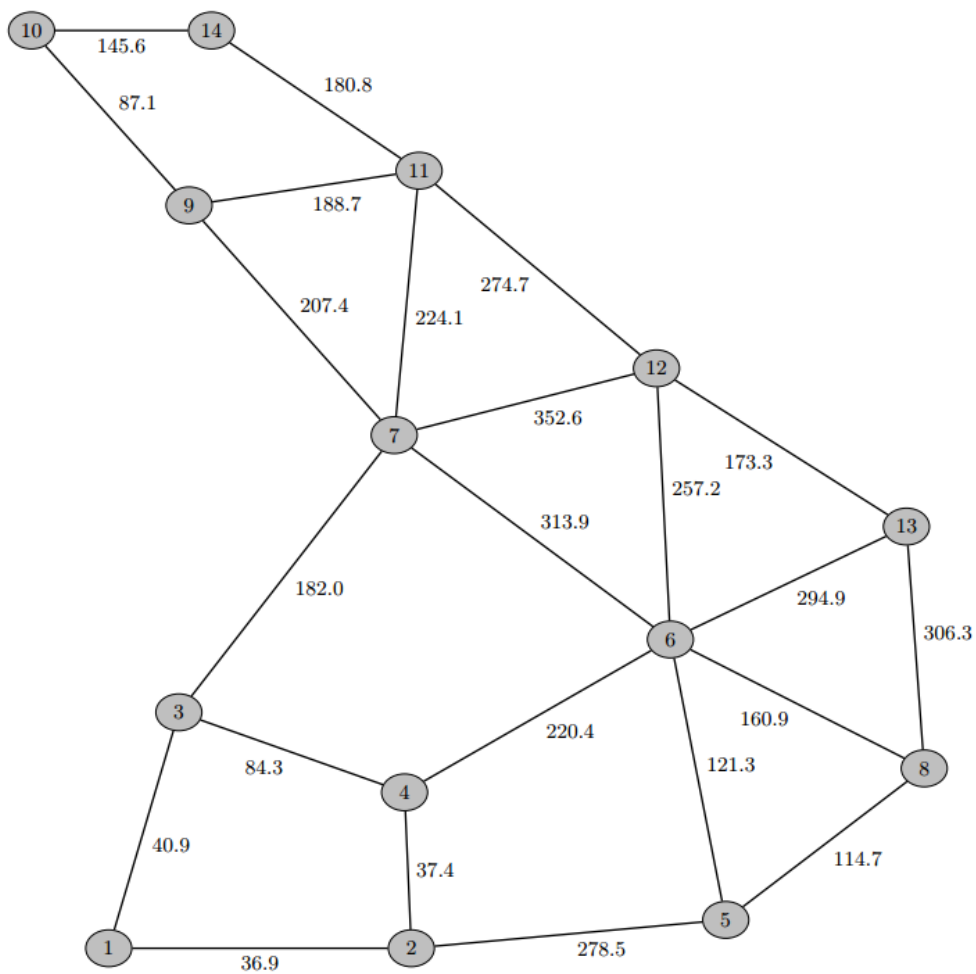


图 4.1: Deutsche Telecom (DT) 拓扑（链路上数字表示其长度，单位：km）

4.1 仿真参数

在我们的仿真中，主要用到 14 点 23 根链路的 Deutsche Telecom (DT) 拓扑^[94] 和一个 50 点 141 条链路的随机拓扑。图4.1和图4.2分别给出了这两个拓扑的拓扑图。图4.1中链路旁的数字表示它的长度（单位：km）。其中随机拓扑是

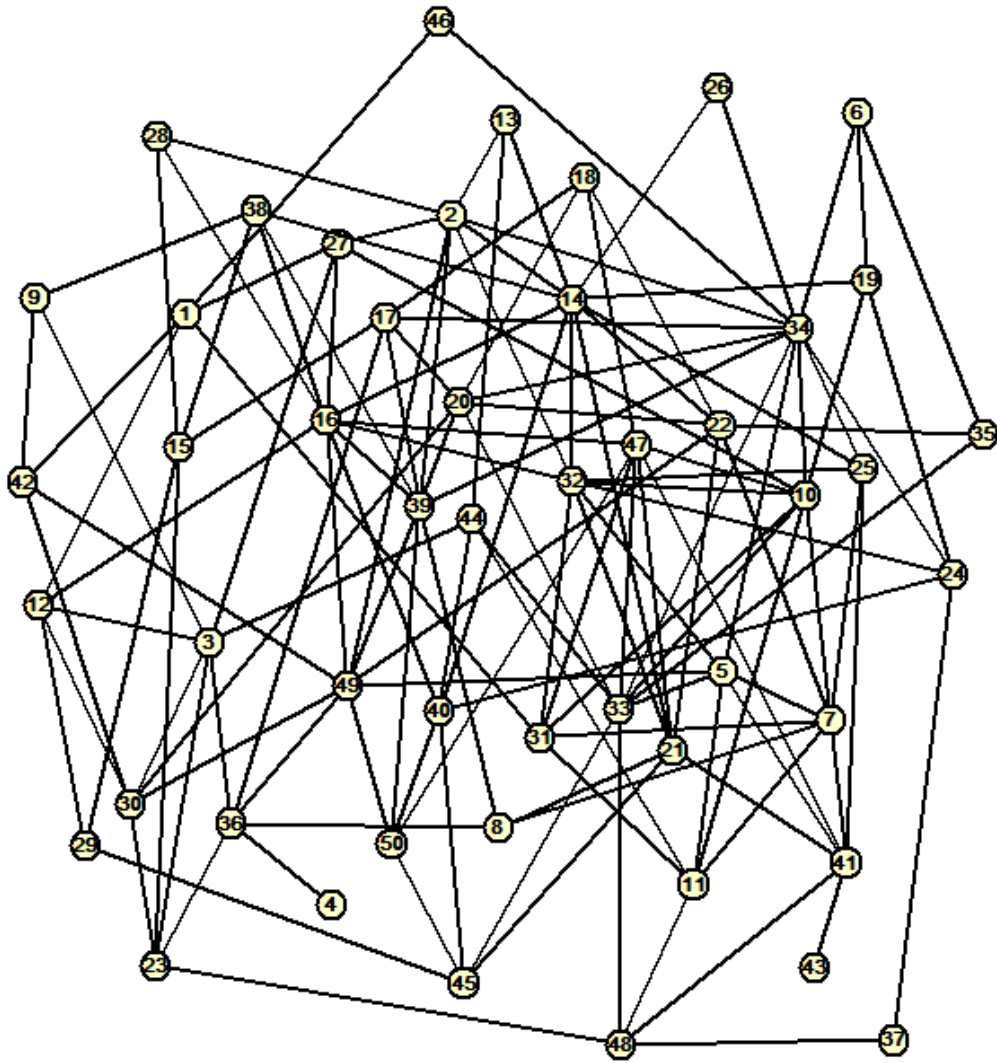


图 4.2: 随机拓扑 (链路长度: 100 km)

采用美国佐治亚理工学院开发的拓扑生成工具 GT-ITM^[95] 产生。在随机拓扑中, 我们假设所有链路的长度都是 100 km。DT 拓扑中每根链路的长度与文献^[94] 中的设置一致。所有的虚拟网络请求也都采用 GT-ITM 工具生成, 其中每一个虚拟光网络的节点个数服从均匀分布, 同时, 同一个虚拟光网络中任何两个虚拟节点之间有边的概率是 0.5, 也就是说 n 个点的虚拟光网络平均将有 $\frac{n \cdot (n-1)}{2}$ 条虚拟光链路。对于动态在线仿真的情形, 我们假设虚拟光网络请求的到达服从泊松过程。表 4.1 总结了我们的仿真中主要的一些参数的值或取值范围。

4.2 对比算法

正如绪论里所说, 由于目前尚没有现存的同时考虑节点和链路映射的透明虚拟光网络映射算法。为了更好的衡量我们算法的性能, 特别是我们提出的分层辅助图的优势, 我们将现存的 2/3 层网络中的虚拟网络映射算法做了适当修改, 作为我们的对比算法。首先, 我们修改了普林斯顿大学 Yu Minlan 等人在文

表 4.1: 主要仿真参数的值或者取值范围

仿真参数	DT 拓扑	随机拓扑
节点个数	14	50
链路条数	23	141
起始计算资源容量	200 units	200 units
起始带宽资源容量	200 FS'	200 FS'
计算资源需求量	[1, 10] units	[1, 20] units
带宽资源需求量	[1, 10] FS'	[1, 20] FS'
一个虚拟光网络中虚拟节点个数	[3, 4]	[2, 10]

输入: 底层光网络 G^s , 虚拟光网络请求 G^r

输出: 节点映射 M_N , 映射状态 F

```

1 计算底层网络中每一个节点  $v^s \in V^s$  的节点度  $\beta_{v^s}$  和资源容量[39], 即
    $c_{v^s}^s \cdot \sum_{(u^s, v^s) \in E^s} b_{(u^s, v^s)}^s$ ;
2 计算每个虚拟节点  $v^r \in V^r$  的节点度  $\beta_{v^r}$ ;
3 按照节点度由大到小对所有虚拟节点进行排序;
4 按照资源容量从大到小对底层网络所有节点进行排序;
5 for 所有虚拟节点  $v^r$ , 按照节点度由大到小 do
6    $F = \text{FAILED}$ ;
7   for 所有未标记的底层物理节点  $v^s$ , 按照资源容量由大到小 do
8     if  $c_{v^s}^s \geq c_{v^r}^r$  AND  $\beta_{v^s} \geq \beta_{v^r}$  then
9        $M_N(v^r) \leftarrow v^s$ ;
10      标记物理节点  $v^s$ ;
11       $F = \text{SUCCEEDED}$ ;
12      break;
13    end
14  end
15  if  $F = \text{FAILED}$  then
16    return( $F$ );
17  end
18 end
19 return( $M_N, F$ );

```

算法 4.1: 基于资源容量的节点映射

献^[39]中提出的基于资源容量和 K-最短路的贪婪虚拟网络映射算法。为了尽可能保证与原算法的相似性，同时也为了使得修改后的算法能够工作在弹性光网络，我们只修改了链路映射算法，算法4.1给出了基于资源容量的节点映射算法的伪代码。实际上，我们设计了两种链路映射算法，即，直接采用最短路径路由结合 Frist-Fit 的频谱分配方案^[22]的链路映射以及基于分层辅助图的链路映射。类似与 LRC-LSP，我们分别将着两个对比算法记作 RC-(SP-FF) 和 RC-LSP，算法4.2和算法4.3分别给出了这两个算法的伪代码。

```

输入: 底层光网络  $G^s$ , 虚拟光网络请求  $G^r$ 
输出: 映射状态 F
1 采用算法4.1进行节点映射;
2 if 节点映射成功 then
3   for 虚拟光链路  $e^r = (u^r, v^r) \in E^r$  do
4     获取虚拟节点  $u^r$  和  $v^r$  映射到的底层节点  $M_N(u^r)$  和  $M_N(v^r)$ ;
5     采用最短路径算法尝试寻找  $M_N(u^r)$  和  $M_N(v^r)$  之间的最短路径  $p$ ;
6     if 找不到路径 then
7       标记  $G^r$  被阻塞;
8        $G^s = G_{temp}^s$ ;
9       return(F = FAILED);
10    end
11     $M_L(e^r) \leftarrow p$ ;
12    在  $G^s$  中删除  $p$  所经过的所有光纤链路;
13  end
14  在所有链路  $e^s \in \{e^s : e^s \in M_L(e^r), \forall e^r \in E^r\}$  上寻找第一段长度为  $n^r$  的可用连续频谱块;
15  if 连续频谱块找到 then
16    | 为  $G^r$  分配资源;
17  else
18    | 标记  $G^r$  被阻塞;
19    |  $G^s = G_{temp}^s$ ;
20    | return(F = FAILED);
21  end
22  return(F = SUCCEEDED);
23 end
24 标记  $G^r$  被阻塞;
25  $G^s = G_{temp}^s$ ;
26 return(F = FAILED);

```

算法 4.2: RC-(SP-FF) 算法

4.3 性能评估参数

在本小节中，我们将从与最优解的间隙（后面部分将直接简称“间隙”）、阻塞率和平均最长光路径的长度三个性能指标来评估我们的算法，并且我们将

```

输入: 底层光网络  $G^s$ , 虚拟光网络请求  $G^r$ 
输出: 映射状态  $F$ 

1  $G_{temp}^s = G^s$ ;
2 采用算法4.1进行节点映射;
3 if 节点映射成功 then
4   for  $k = 1$  to  $B^s - n^r + 1$  do
5     使用算法3.1构建第  $k$  层分层辅助图  $G_k^{sub}$ ;
6     for  $\forall e^r = (u^r, v^r) \in E^r$  do
7       获取虚拟节点  $u^r$  和  $v^r$  映射到的底层节点  $M_N(u^r)$  和
8        $M_N(v^r)$ ;
9       采用最短路径算法尝试寻找  $M_N(u^r)$  和  $M_N(v^r)$  之间的最
10      短路径  $p$ ;
11      if 找不到路径 then
12        标记虚拟光网络请求  $G^r$  被阻塞;
13         $G^s = G_{temp}^s$ ;
14        return( $F = FAILED$ );
15      end
16       $M_L(e^r) \leftarrow p$ ;
17      在  $G_k^{sub}$  中删除  $p$  所经过的所有光纤链路;
18    end
19    为虚拟光网络请求  $G^r$  分配相应的资源;
20    return( $F = SUCCEEDED$ );
21 end
22 标记虚拟光网络请求  $G^r$  被阻塞;
23  $G^s = G_{temp}^s$ ;
24 return( $F = FAILED$ );

```

算法 4.3: RC-LSP 算法

阻塞率作为最主要的性能指标，因为阻塞率直接关系到设备提供商的收益，因此，它将是设备提供商最为关注的指标，其次，由于传输时延以及物理层损伤 (Physical Layer Impairments, PLIs)^[96] 等依赖于光路径的长度 (即，光信号的传输距离)，因此，我们将每个虚拟光网络映射后最长光路径的平均长度也作为一个性能指标。最后，通过与最优解的间隙，我们可以清楚的知道我们的算法与最优解之间的差距，从单个解的质量的角度对我们的算法进行评估。这三个性能指标的定义如下。

定义 4.3.1 (间隙). 给定一个最小化问题 Γ 的一个实例 γ 和一个启发式算法 g ，记 $g(\gamma)$ 为启发式算法 g 运行在实例 γ 上获得的问题 Γ 的解， OPT 为实例 γ 的最优解，那么我们称启发式算法对于实例 γ 的间隙为，

$$\Delta = \frac{g(\gamma) - OPT}{OPT}, \quad (4.1)$$

定义 4.3.2 (阻塞率). 阻塞率定义为在足够长时间内，网络运行过程中被阻塞的

虚拟网络请求的个数与这段时间内总到达的虚拟网络请求的个数的比值，即

$$p_b = \lim_{T \rightarrow \infty} \frac{N_b(T)}{N_a(T)}, \quad (4.2)$$

其中， $N_a(T)$ 和 $N_b(T)$ 分别是 0-T 时段内到达的总的虚拟网络请求的个数和被阻塞的虚拟网络请求的个数。

定义 4.3.3 (平均最长光路径长度). 平均最长光路径长度定义为足够长时间内，网络运行过程中所有被服务的虚拟网络请求映射到的光路中最长光路径的长度的平均值，即

$$\bar{l} = \lim_{T \rightarrow \infty} \frac{\sum_{G^r \in \Omega(T)} \max_{e^r \in E^r} \text{len}(M_L(e^r))}{|\Omega(T)|}, \quad (4.3)$$

其中， $\Omega(T)$ 表示 0-T 时段内被服务的虚拟光网络请求构成的集合， $M_L(e^r)$ 表示虚拟链路 e^r 映射到的底层光路径， $\text{len}(\cdot)$ 是一个函数，返回输入光路径的长度值。

4.4 仿真结果以及结果分析

4.4.1 DT 拓扑下仿真结果

表 4.2: 不同算法单个解的质量对比（底层光网络资源利用率：1%-10%）

算法		LRC-LSP	RC-LSP	RC-(SP-FF)
与最优解的间隙	平均值	26.21%	40.95%	21.05%
	最大值	304.35%	391.30%	83.72%
	最小值	0.00%	0.00%	0.00%
	标准差	0.3151	0.3850	0.2072
求解百分比		99.94%	98.48%	93.18%
到达最优解的百分比		35.56%	21.00%	34.50%

表 4.3: 不同算法单个解的质量对比（底层光网络资源利用率：11%-20%）

算法		LRC-LSP	RC-LSP	RC-(SP-FF)
与最优解的间隙	平均值	7.24%	28.50%	9.84%
	最大值	130.43%	188.46%	62.50%
	最小值	0.00%	0.00%	0.00%
	标准差	0.1454	0.2848	0.1403
求解百分比		95.82%	69.24%	55.38%
到达最优解的百分比		67.22%	19.37%	31.25%

在 DT 拓扑下，我们首先比较了各个算法的解的质量。在这些仿真中，我们将底层网络的初始化计算资源和带宽资源分别设置为 100 个单位和 100 个频谱

带宽槽。我们将底层网络的资源利用率固定在一定范围，对于每一个利用率范围我们生成 100 种不同的资源分布方式，即 100 个资源分布不同的底层光网络。为了方便描述，下面我们假设利用率范围为 μ_1 到 μ_2 。具体而言就是，对于每一个底层节点，其计算资源利用率均服从 μ_1 到 μ_2 的均匀分布，同理，对于每一根光纤链路，其带宽资源的利用率也服从 μ_1 到 μ_2 的均匀分布，并且根据每根链路的带宽资源利用率 μ ，我们随机从 100 个带宽槽中选择 $100 \cdot \mu$ 个带宽槽标记为已用。然后，我们随机生成 50 个有 3-4 个节点，任意两个节点之间有边的概率为 0.5 的虚拟光网络请求，每个节点的计算资源需求量为 1-10 个单位，每根链路的带宽资源需求为 1-10 个带宽槽。然后，我们尝试用算法 *LRC-LSP*、*RC-LSP*、以及 *RC-(SP-FF)* 在这 100 个不同的底层网络上为依次映射这 50 个虚拟光网络请求，在整个仿真过程中每个底层网络的资源使用情况维持不变。表 4.2 和表 4.3 分别统计了资源利用率为 1-10% 和 11-20% 的情况下不同算法的仿真结果，从两个表格我们可以看到，我们提出了算法 *LRC-LSP* 的求解率始终是最高的，但是，从表 4.2 我们发现在资源利用率很低的情况下，我们的算法与最优解的间隙的平均值和最大值均比 *RC-(SP-FF)* 要差一些，而且，我们的算法的最优率（即，求解到最优解的请求占总请求的比例）也略低于 *RC-(SP-FF)*。这主要是由于在资源利用率很低的情况下，分层辅助图拓扑与底层拓扑差别不大，因此，此时 *LRC-LSP* 的节点映射结果并不比对比算法好多少，而在链路映射阶段，算法 *RC-(SP-FF)* 总是采用最短路径来进行链路映射，因此，相对而言，*RC-(SP-FF)* 的解会比较优。从表 4.3 的结果，我们可以看到当资源利用率为 11-20% 时，此时，分层辅助图很容易就被分割成多个连通子图，也就是说算法 *LRC-LSP* 相当于在一个较小的底层拓扑上做映射，因此，从平均意义上说，映射后节点之间的“距离”会比较短，因此，在资源利用率为 11-20% 的情况下，算法 *LRC-LSP* 具有最低的平均间隙，同时拥有最高的最优率和解决率（即，求解出来的虚拟光网络请求的个数与虚拟光网络请求的总个数的比值）。

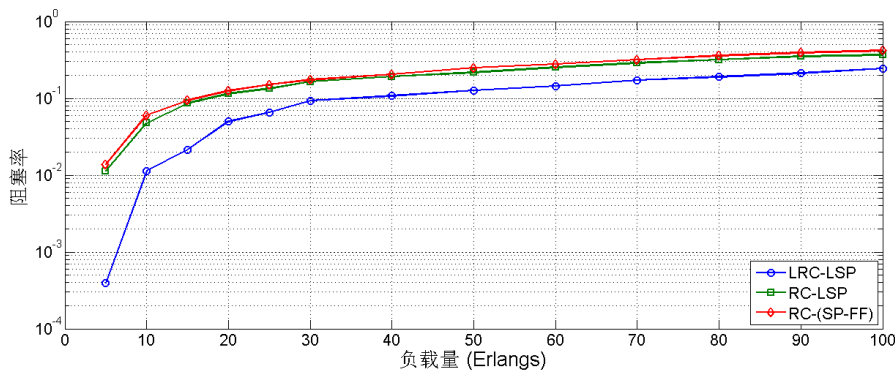


图 4.3: DT 拓扑下的阻塞率

如表 4.1 所示，在 DT 拓扑下，底层网络的初始化的计算和带宽资源均设置为 200 个单位，同时，每个虚拟光网络请求有 3-4 个虚拟节点，任意两个虚拟

节点之间有边的概率是 0.5，每个节点需求的计算资源为 1 – 10，服从均匀分布，每根虚拟光链路的带宽需求为 1 – 10 个频谱带宽槽，也服从均匀分布。

图4.3展示了三个算法在 DT 拓扑作为底层光网络的情况下阻塞率随负载量的变化情况。从图中，我们可以看到算法 *LRC-LSP* 总是拥有最低的阻塞率，算法 *RC-LSP* 的阻塞率略低于算法 *RC-(SP-FF)*。这说明通过分层辅助图的方式在节点映射阶段考虑部分链路资源的相关信息，在链路映射阶段将路由与频谱分配两个子问题整合起来考虑可以极大的提高透明虚拟光网络映射的效果，从而极大的降低阻塞率。

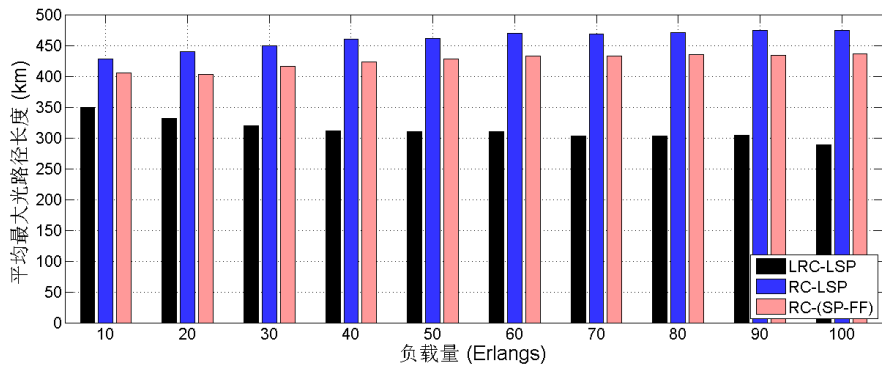


图 4.4: DT 拓扑下的平均最长光路径长度

图4.4统计了所有映射成功的虚拟光网络请求映射后最长光路径的长度的平均值。从图中我们可以看到，算法 *LRC-LSP* 的平均最长光路径的长度也是三个算法中最小的，大约是算法 *RC-(SP-FF)* 的 73%，算法 *RC-LSP* 的 68%。并且从图4.4我们可以看到算法 *LRC-LSP* 得到的平均最长光路径的长度最大也不超过 313 km，远远小于了采用 QPSK 调制的 500 Gb/s 光正交频分复用 (Optical Orthogonal Frequency Division Multiplexing, O-OFDM) 信号的传输极限距离^[97]。

4.4.2 随机拓扑下的仿真结果

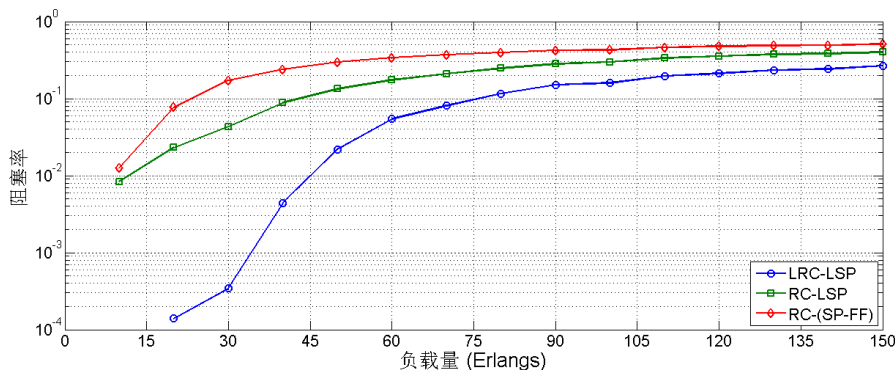


图 4.5: 随机拓扑下的阻塞率

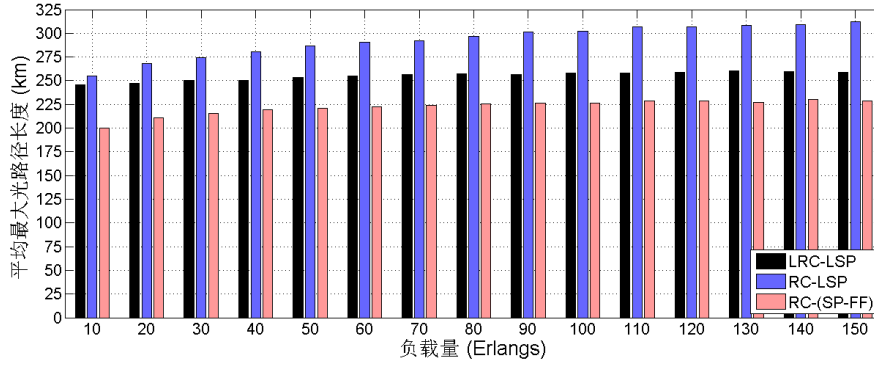


图 4.6: 随机拓扑下的平均最长光路径长度

由于 DT 拓扑的规模的限制，在本章第4.4.1小节中 DT 拓扑作为底层光网络时，我们的虚拟光网络的规模很难超过 4 个节点。为了进一步评估我们算法的性能，我们在图4.2所示的 50 个节点 141 条链路的随机随机拓扑上进行了规模更大的虚拟光网络请求的仿真。在这些仿真中我们的虚拟网络请求有 2-10 个节点，两个节点之间有边的概率依然是 0.5，同时，每个节点的计算资源需求量为 1-20 个单位，每根虚拟链路的带宽需求量为 1-20 个频谱带宽槽。

图4.5和图4.6分别展示了在该随机拓扑做为底层光网络的情况下三个算法的阻塞率和平均最长光路径长度。从图4.5中我们看到，算法 *LRC-LSP* 的阻塞率还是三个算法中最低的，但是，此时算法 *RC-LSP* 的阻塞率却比算法 *RC-(SP-FF)* 的低了很多。此外，从图4.6中，我们发现此时算法 *LRC-LSP* 的平均最长光路径长度要比算法 *RC-(SP-FF)* 的长了一些。这是由于在随机拓扑作为底层光网络拓扑的情况下，底层网络的规模大了很多，同时，虚拟光网络请求的规模大了很多，这就导致平均每个虚拟光网络请求拥有更多的虚拟光链路，而算法 *RC-(SP-FF)* 在节点和链路映射中均没有考虑弹性光网络中频谱资源的特性，这就决定了它很难服务两个端节点映射的底层节点距离很远的虚拟链路，于是，它映射成功的虚拟光网络请求的最长光路径的长度相对而言就会比较小，这也解释了为什么在随机拓扑作为底层光网络拓扑时，算法 *RC-(SP-FF)* 和 *RC-LSP* 阻塞率的差别比在 DT 拓扑里大了很多。

第五章 试验平台

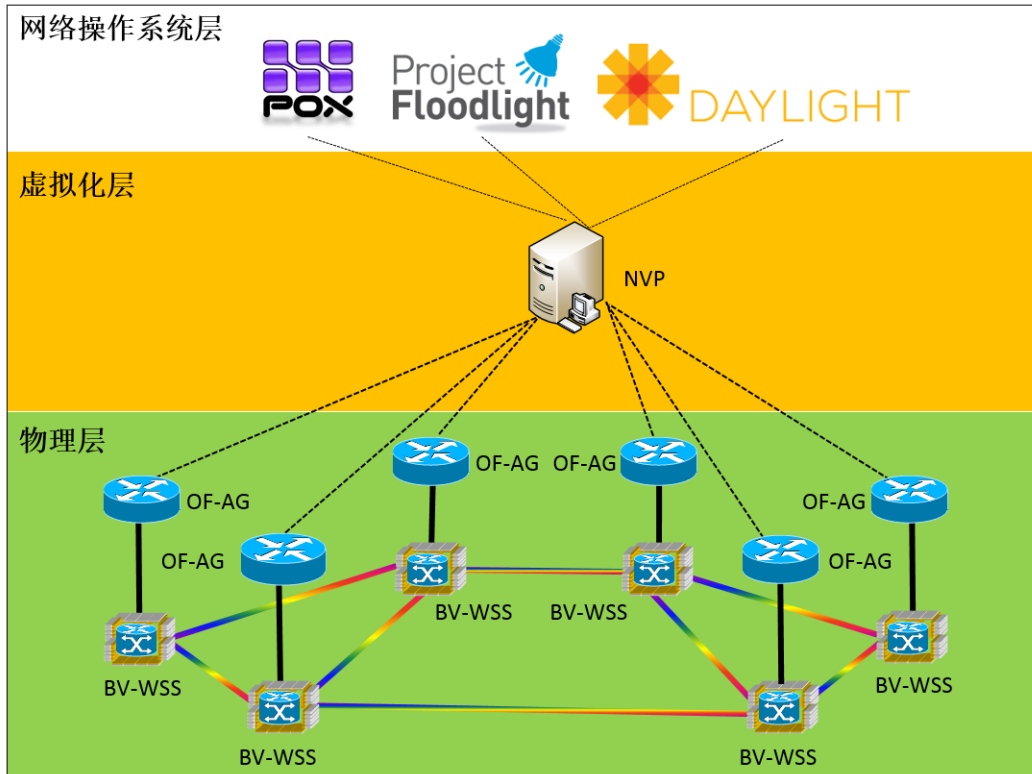


图 5.1: 系统框架

5.1 系统架构

图 5.1 描述了基于软件定义弹性光网络（Software-Defined Elastic Optical Networking, SD-EON)^[98,99] 架构的网络虚拟化系统框架。从图中我们可以看出，在该框架中有物理层，虚拟化层以及网络操作系统层。其中物理层就是一个跨数据中心的软件定义弹性光网络，主要由可变带宽波长选择交换机（Bandwidth-Variable Wavelength Selective Switches, BV-WSS^{*}），连接这些 BV-WSS 的光纤和为 BV-WSS 翻译 OpenFlow 消息的 OpenFlow 代理（OpenFlow Agent, OF-AG）构成^{*}。虚拟化层则是基于 OpenFlow 协议的一个网络管理程序，实际上就是物理网络和用户网络控制器之间的代理，对于底层网络而言，它相当于控制面，而对于用户的网络控制器而言，它则相当于数据面。具体而言，对于每一个虚拟网络，它重写用户网络控制器与底层光网络之间的 OpenFlow 消息。而网络操作系统层则是用户用于控制自己的虚拟网络的 OpenFlow 控制器，如 POX^[100]，Floodlight^[101] 等等。

图 5.2 以一个虚拟网络的生命周期为例展示了该系统的工作流程。

^{*}注意，这里同样省去了每个 BV-WSS 上直连的数据中心。

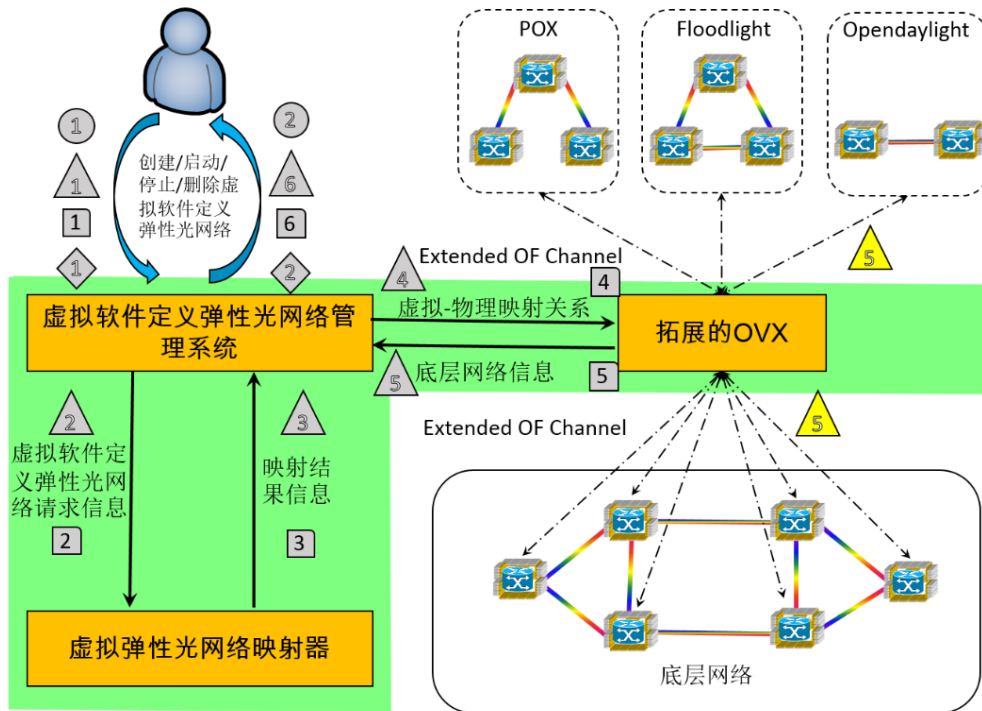


图 5.2: 虚拟化系统工作流程实例

1. 创建

- (a) 用户通过虚拟软件定义弹性光网络管理系统创建自己的虚拟软件定义网络。
- (b) 虚拟软件定义弹性光网络管理系统接收到用户的创建请求之后，将用户的虚拟网络信息存储到数据库，同时，刷新 web 页面跳转到该用户的用户中心页面。

2. 启动

- (a) 用户通过虚拟软件定义弹性光网络管理系统的用户中心页面启动在上一步创建的虚拟网络。
- (b) 虚拟软件定义弹性光网络管理系统从数据库获取该虚拟软件定义网络的信息以及底层网络的拓扑与资源信息，并按照虚拟弹性光网络映射器的要求将这些信息发送给虚拟弹性光网络映射器进行虚拟网络映射。
- (c) 虚拟弹性光网络映射器将映射的结果反馈给虚拟软件定义弹性光网络管理系统。
- (d) 虚拟软件定义弹性光网络管理系统从映射结果中获取虚拟到物理的映射关系，并将这些映射关系通过 E-OVX 提供的 API 配置到底层网络。

- (e) E-OVX 完成配置任务之后，将底层网络状态信息反馈给虚拟软件定义弹性光网络管理系统。与此同时，创建的虚拟软件定义网络开始运作。
- (f) 虚拟软件定义弹性光网络管理系统刷新 web 页面，将该虚拟网络的运行状态改为“Running”。

3. 停止

- (a) 用户通过虚拟软件定义弹性光网络管理系统的用户中心页面停止该虚拟网络。
- (b) 虚拟软件定义弹性光网络管理系统从数据库获取该虚拟软件定义网络的信息以及底层网络的拓扑与资源信息，并按照虚拟弹性光网络映射器的要求将这些信息发送给虚拟弹性光网络映射器进行网络资源的预更新。
- (c) 虚拟弹性光网络映射器将底层网络资源状态反馈给虚拟软件定义弹性光网络管理系统。
- (d) 虚拟软件定义弹性光网络管理系统从数据库获取该虚拟网络的 ID，并通过 E-OVX 提供的 API 发送停止该虚拟网络的配置。
- (e) E-OVX 执行停止网络配置，删除该虚拟网络对应的虚拟到物理的映射关系，并将底层网络状态信息反馈给虚拟软件定义弹性光网络管理系统。与此同时，该虚拟软件定义网络停止运作。
- (f) 虚拟软件定义弹性光网络管理系统更新底层网络资源信息并写入数据库，将该虚拟网络的运行状态改为“Awaiting”。

4. 删除

- (a) 用户通过虚拟软件定义弹性光网络管理系统的用户中心页面删除该虚拟网络。
- (b) 虚拟软件定义弹性光网络管理系统接收到用户的删除请求之后，从数据库中将该用户的该虚拟网络信息删除，同时，刷新 web 页面，通知用户该网络已成功删除。

5.2 协议拓展

正如前文所述，我们的平台主要基于 OpenFlow v1.0^[91]，而该版本的 OpenFlow 协议并不支持光传输网络，更不用说新兴的弹性光网络，为了使其支持弹性光网络我们对其做了一些拓展。根据 OpenFlow 的原理，我们准备采用 OF 中由匹配项、行为以及相关计数器构成的流表来标示一条光路。于是，我们在 OF 协议中添加了如下三个与弹性光网络相关的新的匹配项以及行为。

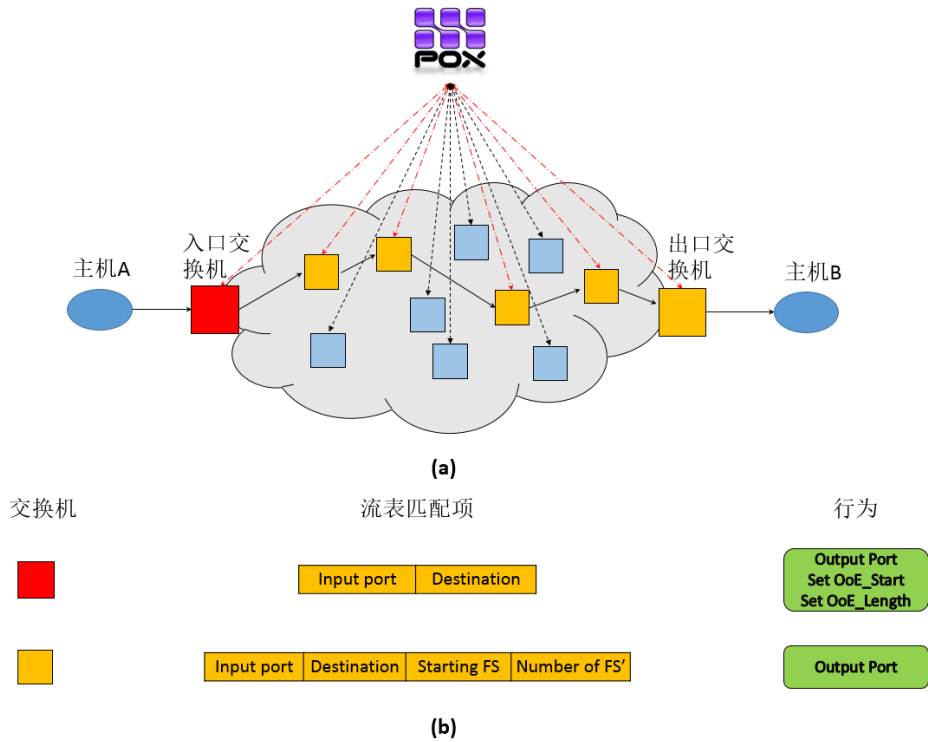


图 5.3: 支持弹性光网络的 OF 的工作原理: (a) 主机 A 到主机 B 的光路 (b) 对应交换机上的流表信息

- 目的节点 (Optical over Ethernet Destination, OoE_DST): 该匹配项的引入主要是为了支持同一个源节点到同一目的节点在同一时刻希望建立多条光路的情形。实际上, 该匹配项包含两个子项: 一个是请求编号, 另一个是真正的目的地址编号。其对应的行为是 Set OoE_DST。
- 起始频谱带宽槽 (Optical over Ethernet Starting Frequency Slot, OoE_Start): 该光路的起始带宽槽下标, 其对应行为为 Set OoE_Start。
- 频谱带宽槽个数 (Optical over Ethernet Length of Frequency Slots, OoE_Length): 该光路频谱带宽槽的个数, 其对应行为是 Set OoE_Length。

图5.3中以主机 A 到主机 B 的光路的实现展示了拓展后的 OF 协议是如何实现对弹性光网络的支持的, 当 POX (或其他 OF 控制器) 接收到建立主机 A 到主机 B 的请求之后, 它会计算一条主机 A (直连的交换机) 到主机 B (直连的交换机) 的一条光路 (即, 图中红色和黄色交换机连成的路径) 以及其上分配的频谱资源。然后, POX 控制器向对应交换机发送 FlowMod 消息 (图5.3(a) 中红色虚线所示)。图5.3(b) 中展示了每个相关交换机上的流表信息以及相关行为。我们可以看到, 入口交换机的 OF-AG 会“指导”对应的 BV-T 操作一段指定的频谱块, 同时将光信息发送到指定的出端口。而中间交换机和出口交换机只需要将从对应入端口接收的光信息发送到对应出端口即可。这里需要注意的是, 实际上, 从主机 A 到对应入口交换机的过程实际上传输的是电信号, 因此, 需要

在入口交换机上将电信号转换成光信号，同时，在出交换机到主机 B 的过程中也是传输的电信号，故，需要在出口交换机上将光信号转换回电信号。

5.3 主要功能模块

图5.2中的绿色阴影部分展示了虚拟化平台的主要功能模块。从图中我们可以看到该虚拟化平台主要有三个功能模块：虚拟软件定义弹性光网络管理系统（Virtual Software-Defined Elastic Optical Network Management System, vSD-EONMS），虚拟弹性光网络映射器（Virtual Elastic Optical Network Embedder, VEONE）和拓展的 OpenVirteX（E-OVX）。下面我们将一一介绍这三个功能模块。

5.3.1 虚拟软件定义弹性光网络管理系统

虚拟软件定义弹性光网络管理系统是基于 Bootstrap^[102] 和 Django^[103] 开发的一个图形化用户接口。用户可以很容易地通过该接口查看和管理自己的虚拟软件定义网络。其主要功能如下：

1. 创建和配置虚拟节点（即，虚拟交换机和虚拟数据中心）以及虚拟光链路。
2. 管理每个虚拟软件定义弹性光网络的控制器。
3. 管理（启动/停止/删除）其软件定义弹性光网络。
4. 以图形化的方式展示用户的虚拟网络。

图5.4展示了虚拟软件定义弹性光网络管理系统的虚拟网络创建页面。从图中我们可以看到，该页面主要包含 7 个模块。

模块 1 用户菜单。用户可以通过该菜单访问虚拟网络创建页面（new virtual network），自己的用户中心页面（my profile），更改账号密码（change password）以及注销登录（logout）。

模块 2 虚拟网络描述信息。该模块需要用户选择虚拟网络的类型（type），填写虚拟网络的名字（name）以及该虚拟网络的详细描述（description）。

模块 3 虚拟网络规模信息与控制器信息。该模块需要用户填写待创建虚拟网络中的交换机个数和链路条数以及控制器相关信息。

模块 4 节点信息。通过该模块，用户需要填写每个交换机上直连的数据中心上需求的资源量，此外，用户还可以根据需求动态添加或者删除节点。

模块 5 链路信息。该模块准许用户指定待创建虚拟网络的每一条光链路的直连交换机及其端口号和该虚拟光链路上需求的频谱带宽槽个数，需要注意的是如果用户在模块 2 中选择的是透明虚拟光网络，那么在链路资源那一列只用填写任何一根链路的资源即可。

new virtual network

type: Transparent Optical Network
 name: identical name of your VN (no blank space)
 description: description of your VN

general configuration

node number: 4
 link number: 6
 controller type: Default
 controller ip: localhost
 controller port: 6633

switch configuration

id	name	resource	action
1	sw1		Delete
2	sw2		Delete
3	sw3		Delete
4	sw4		Delete

Reset Add

link configuration

id	name	bandwidth	source (switch/port)	destination (switch/port)	action
1	link1				Delete
2	link2				Delete
3	link3				Delete
4	link4				Delete
5	link5				Delete
6	link6				Delete

Reset Add

host configuration

id	name	switch connected (switch/port)	action
1	h1	sw1 / port1	Delete
2	h2	sw2 / port1	Delete
3	h3	sw3 / port1	Delete
4	h4	sw4 / port1	Delete

Reset Add

Preview Submit

图 5.4: 虚拟软件定义弹性光网络管理系统新建页面

模块 6 主机信息。该模块准许用户自助添加或者减少待创建虚拟网络中的主机数目以及主机连接到的交换机及其端口。

模块 7 预览和提交。该模块准许用户在任何时刻通过预览按钮以图片形式查看自己创建的虚拟光网络，同时在用户填写完所有信息之后，“提交”按钮会被激活，用户此时可以通过它提交自己创建的虚拟光网络。在用户点击“提交”按钮之后，页面会自动跳转到用户中心页面（图5.5）。

The screenshot shows a web interface for managing virtual networks. At the top, there is a navigation bar with 'EON-NVP', 'Home', and 'About'. Below it, the page title is 'demo nvp's virtual network'. A table lists two virtual networks:

id	name	switch number	link number	host number	status	action
1	demo_net_02	3	2	3	Awaiting	Devirtualize More
2	demo_net_01	4	4	4	Awaiting	Virtualize More Start Delete Edit

Below the table, a network diagram is shown. It consists of three switches: sw1, sw2, and sw3. sw1 is connected to host h1. sw2 is connected to host h2. sw3 is connected to host h3. There are also connections between sw1 and sw2, and between sw2 and sw3.

图 5.5: 虚拟软件定义弹性光网络管理系统用户中心页面

图5.5展示了用户中心页面的一个实例。从该页面用户可以看到它的所有虚拟网络的基本信息（例如，节点数和链路数），同时它还可以通过“action”里的按钮以图片形式查看每个虚拟网络的详细信息（visualize），改变该虚拟网络的状态，例如，对于“Awaiting”状态的虚拟网络，它可以启动（start）或者删除（delete）该虚拟网络。

5.3.2 虚拟弹性光网络映射器

虚拟弹性光网络映射器就是本虚拟化平台中用于虚拟光网络映射的模块。类似于文献^[90]，我们选择将该功能作为一个独立的模块。主要有如下原因：

1. 方便设备提供商升级或者更换虚拟光网络映射算法。
2. 方便该功能的单独调试。
3. 设备提供商可以根据用户需求的不同自适应的采用相应的虚拟网络映射算法。

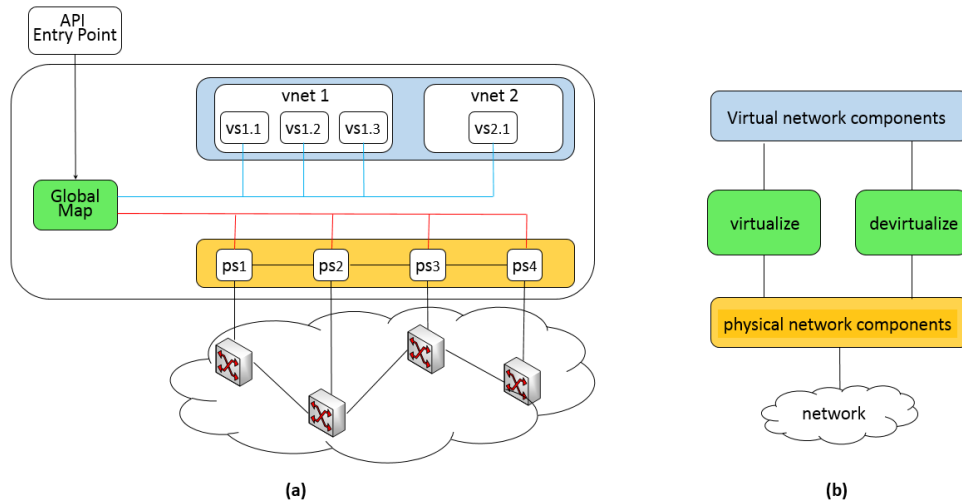


图 5.6: (a) 通过 E-OVX，底层物理网络（橙色）与虚拟网络（蓝色）从逻辑上被分离开，同时，不同的虚拟网络在逻辑上也是隔离的，每个用户只能“看到”自己的虚拟网络。虚拟网络 vnet 1 和 vnet 2 的拓扑信息以及通过 API 配置的它们与物理网络之间的映射都被存储在 Global Map（绿色）中^[104] (b) 在 Global Map 中，虚拟网络与物理网络之间的分割通过 OpenFlow 消息的虚拟化以及解虚拟化实现^[104]

5.3.3 拓展的 OpenVirteX

拓展的 OpenVirteX (E-OVX) 是本虚拟化平台的核心。OpenVirteX 是 Stanford 大学开发的基于 OpenFlow 协议的网络虚拟化平台。E-OVX 为每一个虚拟网络以及物理网络建立并维护其逻辑拓扑信息。对于物理网络的逻辑拓扑信息，E-OVX 通过拓扑发现协议来实现，而对于虚拟网络，E-OVX 则借助其 API 提供的配置信息来完成。

如图5.6所示，E-OVX 将物理网络和虚拟网络在逻辑上的耦合限制在：1) 用于存储物理网络和虚拟网络之间映射的一部分共享的全局结构（即，Global Map）；2) 在 OpenFlow 消息事件流中对控制信道 OF 消息的重写和解复用。这样整个控制信道就被分成如下两个部分。

- 用户控制器与 E-OVX 的北向接口之间的虚拟部分（图5.6中蓝色部分）。
- 底层物理网络与 E-OVX 的南向接口之间的物理部分（图5.6中橙色部分）。

耦合部分主要用于“翻译”穿越虚拟和物理部分的 OF 消息。由于物理部分与虚拟部分在逻辑上是解耦合的，因此，即使没有虚拟层的用户网络控制器甚至没有虚拟层，OVX 自身也能够建立并维持与底层数据面之间的连接。

在本节接下来的部分，我们将详细介绍 E-OVX 的几大主要功能：拓扑虚拟化、地址虚拟化、控制功能虚拟化和频谱资源虚拟化。实际上，拓展前的 OVX 也就有前三大功能，不过其中拓扑虚拟化和控制功能虚拟化在 E-OVX 中做了相应拓展。

5.3.3.1 拓扑虚拟化

正如前面所述，E-OVX 准许用户根据自身需求建立“任意”虚拟拓扑*。实际上，这些虚拟拓扑不必完全与底层物理拓扑一一对应（即，不需要一定是底层物理拓扑的子图），它们可以完全根据用户的实际需求构建。E-OVX 通过解析并回复从用户控制器发送来的 LLDP 消息向用户控制提供虚拟拓扑信息。具体而言就是，当一个在消息体中明确了出端口的 LLDP 消息到达一个虚拟组件时，E-OVX “知道”对应虚拟链路的另一端在哪，因此，E-OVX “伪造” LLDP 响应消息，并将其发送到相应的用户控制器，从而为用户控制器制造了“链路”的假象。

5.3.3.2 地址虚拟化

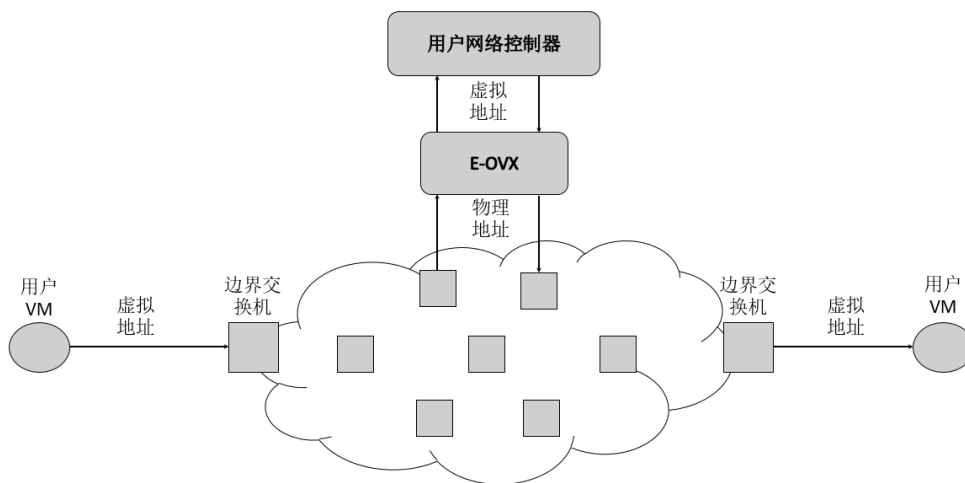


图 5.7: E-OVX 地址改写机制实例^[90]

E-OVX 准许用户为其端主机选择地址，准许多个可能重叠地址在底层物理网络中存在。为了区分不同的端主机，E-OVX 为每个虚拟网络分配一个唯一的 ID，这样，对于每一个端主机，E-OVX 为其生成一个采用这个 ID 来编码其从属关系的物理地址。

*由于透明虚拟网络映射的频谱连续性要求，实际上，虚拟拓扑并不能真的“任意”，至少虚拟节点的节点度是不能超过对应物理节点的。

E-OVX 通过地址改写机制来实现地址虚拟化，从而避免地址冲突。具体而言，在下发流表时，E-OVX 通过边界交换机上地址改写机制，即在入口交换机上将用户分配的地址转换成物理网络的地址，而在出口交换上执行相反的改写。图5.7展示了该地址改写机制的一个实例。

需要注意的是，上述地址改写机制对于用户控制器而言是“不可见”的。也就是说，用户的控制器不需要做任何对应的修改。

5.3.3.3 控制功能虚拟化

E-OVX 准许每个虚拟网络拥有自己的控制器，因此，用户可以根据自己的需求来管控自己的虚拟交换机。为了支持这些管控功能，E-OVX 必须能将虚拟网络中的这些管控功能合理地映射到物理网络之中。因此，E-OVX 有时可能需要将虚拟网络中的一条相对简单的管控操作“翻译”成多条不同的 OF “行为”。E-OVX 能够实现这种“翻译”，主要是由于它处在用户控制器和底层网络的“中间”，因此，它可以“截获”来自用户控制器的管控信息，从而将不同用户的多个不同控制面复用在同一个控制面之上。这样，E-OVX 就可以让每个用户控制器都产生一种整个系统中只有它在管控的“假象”。

5.3.3.4 频谱资源虚拟化

E-OVX 使用频谱下标更改机制来实现频谱资源的虚拟化。具体而言就是，为了使得用户控制器“感觉”它是整个系统中唯一的控制器，E-OVX 准许每一根虚拟光链路的起始带宽槽下标从一开始，显然，这样可能导致物理网络中某些光纤链路上频谱资源的“重叠”。为了避免这种情形，在每个边界交换机上进行频谱下标改写，也就是在每个入口交换机将每个虚拟光链路的起始下标改写成物理网络中的下标，然后，在每一个出口交换机再改写回去。

与地址改写机制类似，频谱下标改写对于用户控制器而言也是“透明”的。因此，用户控制器不需要做任何修改。

5.4 试验结果

5.4.1 试验环境

我们采用 6 台“PC”搭建了如图5.8所示的实验环境。其中，每台机器的详细配置如下。

虚拟软件定义弹性光网络管理系统（图5.8(a)）在一台 2 G 内存安装 64 位 Ubuntu 12.04 操作系统的 VirtualBox^[105] 虚拟机，它通过桥接模式连接到外部网络，因此，与物理主机在同一个网段。对应物理机器是一台 8 G 内存搭载 3.40 GHz 英特尔第二代酷睿 i3-2130 双核 CPU 安装 64 位 Windows 7 专业版的台式电脑。我们采用 MySQL^[106] 做 django 数据库，采用 Apache^[107] 来部署 django。

拓展的 OpenVirteX(图5.8(b)) 运行在一台 4 G 内存搭载 3.10 GHz 英特尔酷睿 i5-2400 四核 CPU 安装 64 位 Ubuntu 12.04 操作的系统的台式机。

虚拟弹性光网络映射器(图5.8(c))的配置与拓展的 OpenVirteX 的相同。我们采用 MATLAB R2011 来实现我们的 LRC-LSP 算法,为了能够和虚拟软件定义弹性光网络管理系统通信,我们同时采用 MATLAB 的 web 模块搭建了一个 web 服务器监听在 4000 端口。

底层软件定义弹性光网络(图5.8(d))采用 Mininet^[108]来模拟。实际上,Mininet 内集成的软件交换机 Open vSwitch^[109]的本身并不支持弹性光网络,我们对其做了相应的拓展。运行 Mininet 的是一台 1 G 内存安装 64 位 Ubuntu 12.04 操作系统的 VirtualBox 虚拟机,它也通过桥接模式连接到外部网络。对应物理主机是一台 4 G 内存搭载 3.00 GHz 英特尔奔腾 E5700 双核 CPU 安装 64 位 Windows 7 旗舰版的台式电脑。

用户控制器(图5.8(e))采用拓展的 POX 控制器^[98,99]。它们运行在一台 4 G 内存搭载 3.00 GHz 英特尔奔腾 E5700 双核 CPU 安装 64 位 Windows 7 旗舰版的台式电脑。实验中,我们采用 Python 中的多进程模块(即, Multiprocessing)来实现用户 POX 控制器的动态启动与关闭。同时,为了实现与虚拟软件定义弹性光网络管理系统的通信,我们将 POX 模块封装在一个 web 服务器中,该 web 服务器采用开源事件驱动网络引擎 Twisted^[110]实现,监听 8888 端口。具体而言,当它从虚拟软件定义弹性光网络管理系统接收到启动 POX 的命令和待启动 POX 的端口信息之后,便通过多进程模块启动一个新的进程运行 POX 并将该进程加入运行进程队列,而当它收到用户停止某个 POX 控制器的请求之后,它会从运行进程队列中将对应进程取出,并停止该进程。

测试用户(图5.8(f))采用 Safari 浏览器^[111]来访问虚拟软件定义弹性光网络管理系统。

5.4.2 系统功能测试

在这一部分,我们采用图5.9所示的 DT 拓扑作为底层软件定义弹性光网络以一个虚拟软件定义网络的生命周期为例来测试我们的系统确实实现了虚拟化的功能。该实验的流程如下。

Step 1 用户通过虚拟软件定义弹性光网络管理系统的虚拟网络创建页面创建一个如图5.10的透明虚拟弹性光网络。其中 sw_i , $i = 1, 2, 3$ 和 h_j , $j = 1, 2, 3$ 分别表示第 i 个交换机和第 j 个主机。图中红色数字表示(虚拟)端口号,交换机和链路上的数字分别表示与对应虚拟节点需求的计算资源量 and 对应链路需求的频谱带宽槽个数。

Step 2 启动该虚拟网络。

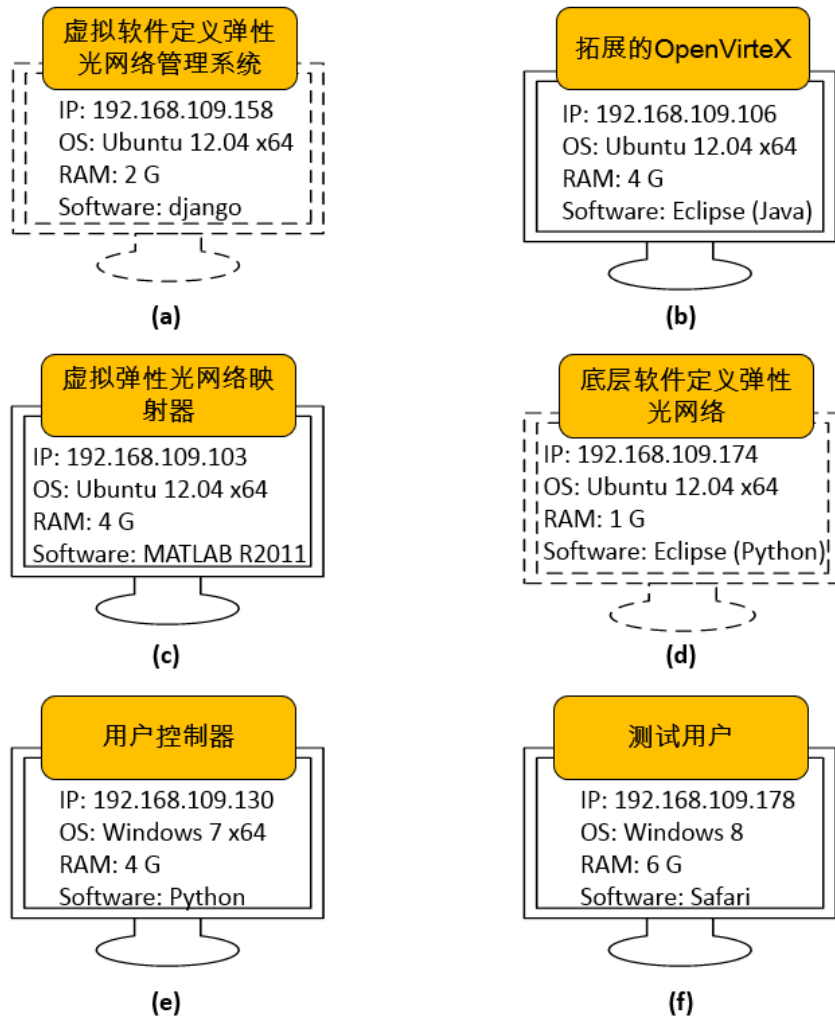


图 5.8: 实验环境（实线：物理机器，虚线：虚拟机）

Step 3 通过虚拟软件定义弹性光网络管理系统的用户中心页面以图形方式查看一下该虚拟网络，看是否与图5.10描述的一致。

Step 4 从E-OVX提供的API接口获取 h_1 和 h_3 对应的“VM”（实际上是Mininet里创建的host）。然后，通过xterm连接到对应主机。

Step 5 从 h_1 向 h_3 发送数据。

Step 6 数据发送完之后，停止该虚拟网络。

Step 7 再次尝试从 h_1 向 h_3 发送数据。

Step 8 删除该虚拟网络。

图5.11是上述实验过程中虚拟软件定义弹性光网络管理系统机器上Wireshark抓包的结果。从图中容易看出，实验结果与本章第5.1描述的流程是一致的。

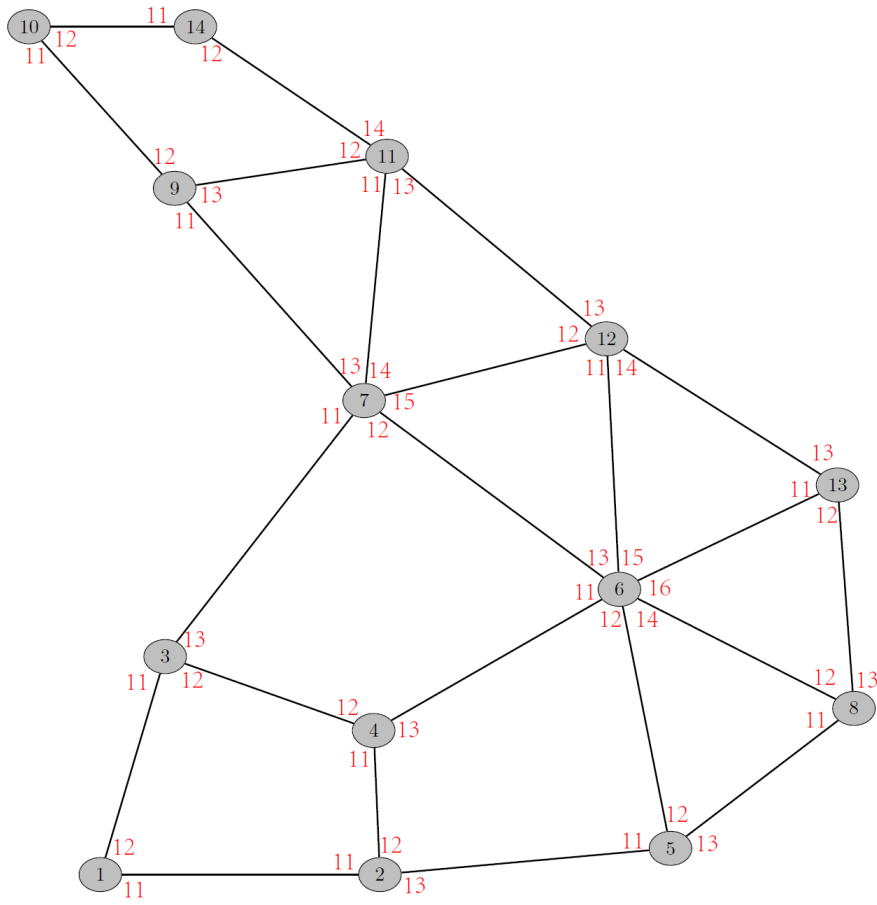


图 5.9: 系统功能验证试验中使用的底层网络拓扑 (其中红色数字表示端口号)

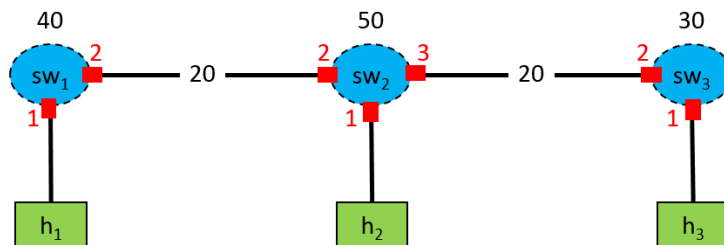


图 5.10: 系统功能验证试验中使用的虚拟网络拓扑

表 5.1: 虚拟软件定义弹性光网络管理系统抓包结果详解

红框标号	用户操作	虚拟软件定义弹性光网络管理系统
1	访问虚拟网络新建页面，开始创建虚拟网络	向用户展示虚拟网络创建页面，在用户创建过程中适时给予提示，并检测用户填写虚拟网络信息的合法性
2	提交虚拟网络	将用户提交的虚拟网络存储到数据库，同时跳转到用户中心页面
3	点击“start”按钮	
4		将虚拟网络信息发送给虚拟弹性光网络映射器进行虚拟网络映射，并获取映射结果
5		向用户控制器模块发送端口询问请求，获取可用的 POX 端口
6		向 E-OVX 发送映射关系，并向用户控制器模块发送启动 POX 控制器的请求
7		刷新用户中心页面，向用户展示虚拟网络启动结果
8	点击“virsualize”按钮	向用户展示该虚拟网络的图形
9	点击“stop”按钮	
10		将该虚拟网络资源分配信息发送到弹性光网络映射器进行资源释放
11		向 E-OVX 发送停止虚拟网络请求
12		向用户控制器模块发送停止 POX 控制器的请求
13		刷新用户中心页面，向用户展示虚拟网络停止结果
14	点击“delete”按钮	检测该虚拟网络的状态，如果处于运行状态，则先停止虚拟网络，然后从数据库删除该网络；否则，直接删除，并将结果返回给用户

No.	Time	Source	Destination	Protocol	Length	Info
1	802.65.262922	192.168.109.178	192.168.109.158	HTTP	508	GET /new/ HTTP/1.1
	806.65.281492	192.168.109.158	192.168.109.178	HTTP	74	HTTP/1.1 200 OK (text/html)
	1497.120.218141	192.168.109.178	192.168.109.158	HTTP	74	POST /new/ HTTP/1.1 (application/x-www-form-urlencoded)
2	1499.120.272275	192.168.109.158	192.168.109.178	HTTP	408	HTTP/1.1 302 FOUND (text/html)[Malformed Packet]
	1503.120.286702	192.168.109.178	192.168.109.158	HTTP	536	GET /user/nvp_demo/ HTTP/1.1
	1506.120.304910	192.168.109.158	192.168.109.178	HTTP	74	HTTP/1.1 200 OK (text/html)
3	1928.155.479786	192.168.109.178	192.168.109.158	HTTP	69	POST /vnManager/ HTTP/1.1 (application/x-www-form-urlencoded)
	1936.155.524156	192.168.109.158	192.168.109.103	HTTP	500	POST /vnem/tvonem.m HTTP/1.1 (application/json)
4	1956.156.578013	192.168.109.103	192.168.109.158	HTTP	993	HTTP/1.0 200 OK (text/html)
	1971.156.656117	192.168.109.158	192.168.109.130	HTTP	265	POST / HTTP/1.1 (application/json)
5	1972.156.663637	192.168.109.130	192.168.109.158	HTTP	235	HTTP/1.1 200 OK (application/json)
	1979.156.666173	192.168.109.158	192.168.109.105	HTTP	417	POST /tenant HTTP/1.1 (application/json-rpc)
	1982.156.669023	192.168.109.105	192.168.109.158	HTTP	591	HTTP/1.1 401 Unauthorized (text/html)
6						
	2360.156.856007	192.168.109.158	192.168.109.105	HTTP	370	POST /tenant HTTP/1.1 (application/json-rpc)
	2363.156.868210	192.168.109.105	192.168.109.158	HTTP	66	HTTP/1.1 200 OK (application/json)
	2370.156.918171	192.168.109.158	192.168.109.178	HTTP	74	HTTP/1.1 200 OK (text/html)
7	2372.156.926721	192.168.109.178	192.168.109.158	HTTP	544	GET /user/nvp_demo/ HTTP/1.1
	2376.156.962180	192.168.109.158	192.168.109.178	HTTP	74	HTTP/1.1 200 OK (text/html)
	2379.157.021294	192.168.109.178	192.168.109.158	HTTP	573	GET /static/bootstrap/css/bootstrap.min.css HTTP/1.1
	2380.157.023502	192.168.109.158	192.168.109.178	HTTP	207	HTTP/1.1 304 NOT MODIFIED
8						
	2449.157.382795	192.168.109.158	192.168.109.178	HTTP	208	HTTP/1.1 304 NOT MODIFIED
9	7206.593.212047	192.168.109.178	192.168.109.158	HTTP	68	POST /vnManager/ HTTP/1.1 (application/x-www-form-urlencoded)
	7212.593.258338	192.168.109.158	192.168.109.103	HTTP	1428	POST /vnem/tvonemReleaseResource.m HTTP/1.1 (application/json)
10	7223.594.150057	192.168.109.103	192.168.109.158	HTTP	637	HTTP/1.0 200 OK (text/html)
	7230.594.285884	192.168.109.158	192.168.109.105	HTTP	340	POST /tenant HTTP/1.1 (application/json-rpc)
11	7232.594.287559	192.168.109.105	192.168.109.158	HTTP	591	HTTP/1.1 401 Unauthorized (text/html)
	7238.594.288423	192.168.109.158	192.168.109.105	HTTP	371	POST /tenant HTTP/1.1 (application/json-rpc)
	7241.594.320713	192.168.109.105	192.168.109.158	HTTP	66	HTTP/1.1 200 OK (application/json)
12	7252.594.360955	192.168.109.158	192.168.109.130	HTTP	273	POST / HTTP/1.1 (application/json)
	7253.594.369540	192.168.109.130	192.168.109.158	HTTP	255	HTTP/1.1 200 OK (application/json)
	7260.594.371781	192.168.109.158	192.168.109.178	HTTP	74	HTTP/1.1 200 OK (text/html)
13	7262.594.382787	192.168.109.178	192.168.109.158	HTTP	544	GET /user/nvp_demo/ HTTP/1.1
	7266.594.421716	192.168.109.158	192.168.109.178	HTTP	74	HTTP/1.1 200 OK (text/html)
14	8304.684.588085	192.168.109.178	192.168.109.158	HTTP	70	POST /vnManager/ HTTP/1.1 (application/x-www-form-urlencoded)
	8307.684.627273	192.168.109.158	192.168.109.178	HTTP	74	HTTP/1.1 200 OK (text/html)

图 5.11: 虚拟软件定义弹性光网络管理系统上 Wireshark 抓包截图

No.	Time	Source	Destination	Protocol	Length	Info
14	123.819936	00:00:00_00:07:02	00:00:00_00:00:00	OoE	75	0 > 0 [Request to set up a new lightpath to 0x3: Bandwidth requirement = 0x1e GHz]
15	124.998103	00:00:00_00:11:02	00:00:00_00:07:02	OoE	34	0 > 0 [lightpath set up successfully, Req_id = 0x0]
16	130.084441	00:00:00_00:07:02	00:00:00_00:11:02	OoE	256	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
18	135.086566	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
19	140.081686	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
21	145.085713	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
22	150.088648	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
24	155.082098	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
25	160.077309	00:00:00_00:07:02	00:00:00_00:11:02	OoE	223	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
27	165.082180	00:00:00_00:07:02	00:00:00_00:11:02	OoE	244	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]
28	170.086263	00:00:00_00:07:02	00:00:00_00:11:02	OoE	244	0 > 0 [Dst = 0x3 0x0, Slot = (0, -1), Modulation = BPSK]

Frame 14: 75 bytes on wire (600 bits), 75 bytes captured (600 bits)						
Ethernet II, Src: 00:00:00_00:07:02 (00:00:00:00:07:02), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)						
OoE Protocol, REQ_TYPE: LIGHTPATH_SETUP_REQ(1) Dst: 0x3(3) Req_id: 0x0(0) frequency_slot_start_index: 0x0(0) frequency_slot_number: 0x0(0)						
0000 00 00 00 00 00 00 00 00 00 07 02 09 08 00 00						
0010 00 03 00 00 00 00 00 00 00 00 1e 10 00 00 00						
0020 00 00 49 20 61 6d 20 61 20 6c 69 67 68 74 70 61 ..I am a lightpa						
0030 74 68 20 73 65 74 75 70 70 72 65 71 75 65 73 74 th setup request						
0040 20 6f 6f 65 20 70 61 63 6b 65 74 ooe pac ket						

图 5.12: 主机 h₁ 上 Wireshark 抓包截图

No.	Time	Source	Destination	Protocol	Length	Info
15	142.681371	00:00:00_00:07:02	00:00:00_00:11:02	OoE	256	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
17	147.684074	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
18	152.676253	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
20	157.683075	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
21	162.687105	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
23	167.676207	00:00:00_00:07:02	00:00:00_00:11:02	OoE	286	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
24	172.683369	00:00:00_00:07:02	00:00:00_00:11:02	OoE	223	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
26	177.676194	00:00:00_00:07:02	00:00:00_00:11:02	OoE	244	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
27	182.687209	00:00:00_00:07:02	00:00:00_00:11:02	OoE	244	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
29	187.686920	00:00:00_00:07:02	00:00:00_00:11:02	OoE	244	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]
30	192.679496	00:00:00_00:07:02	00:00:00_00:11:02	OoE	244	0 > 0 [Dst = 0x3 0x0, Slot = (1, 3), Modulation = BPSK]

Frame 15: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)	
Ethernet II, Src: 00:00:00_00:07:02 (00:00:00:00:07:02), Dst: 00:00:00_00:11:02 (00:00:00:00:11:02)	
OoE Protocol, REQ_TYPE: DATA_SENDING (0) Dst: 0x3 (3) Req_id: 0x0 (0) frequency_slot_start_index: 0x1 (1) frequency_slot_number: 0x3 (3)	


```

0000 00 00 00 00 11 02 00 00 00 00 07 02 09 08 00 00 .....
0010 00 03 00 00 00 01 00 00 00 03 00 00 00 00 00 .....
0020 00 00 09 09 09 09 09 09 09 20 20 20 0d 0a 20 20 .....
0030 20 20 2e 45 45 45 45 45 45 45 45 45 45 45 45 ..EEEEEEEE
0040 45 45 20 20 20 20 4b 45 45 45 45 45 45 45 45 EE KE EEEEEEE
0050 45 45 20 20 20 0d 0a 20 20 20 20 45 45 45 EEE .. EEE
0060 45 45 45 45 45 45 45 45 45 45 20 20 20 20 FFFFFFFF FFFF

```

图 5.13: 主机 h_3 上 Wireshark 抓包截图

的。表 5.1 对每一个红色框对应的用户操作以及虚拟软件定义弹性光网络管理系统的操作做了详细总结。

图 5.12 和图 5.13 分别展示了主机 h_1 和主机 h_3 上 Wireshark 抓包的结果。从图 5.12 我们可以看到，主机 h_1 首先向控制器发送了一个光路建立请求（LIGHT_PATH_SETUP_REQ）希望建立一条到节点 3 带宽为 30 GHz 的光路（即，图 5.12 中的第 1 个数据包），然后，收到了来自控制器的光路建立成功的回复*。光路建立成功之后，主机 h_1 便开始了数据的发送。从图 5.13 可以看出，主机 h_3 成功接收到了主机 h_1 的数据。

图 5.14 展示了试验中图 5.10 虚拟软件定义网络的映射关系，从图中我们可以看到，虚拟交换机 sw_1 、 sw_2 和 sw_3 分别被映射到物理交换机 7、6 和 11，虚拟光链路 (sw_1, sw_2) 、 (sw_2, sw_3) 分别被映射到底层物理光路 $7 \rightarrow 6$ 和 $6 \rightarrow 12 \rightarrow 11$ 。

图 5.15 展示了实验过程中 E-OVX 机器上 Wireshark 抓包的结果[†]。从图中我们可以看到 h_1 开始发送数据时，由于网络中没有流表因此触发了一个 OF PacketIn 消息（即，图中第 3 个数据包）到 E-OVX，E-OVX 检测数据包的合法性，找到该数据包归属的虚拟网络，然后，将入端口改为虚拟网络中对应入端口后 PacketIn 到对应的用户控制器（即，图中第 4 个数据包）。用户控制器拿到数据包之后，发现 h_1 希望建立一条到 h_3 的光路，于是，获取网络频谱资源信息和拓扑信息进行路由和频谱分配，通过计算该用户控制器找到一条从 h_1 到 h_3 的光路，即 $sw_1 \rightarrow sw_2 \rightarrow sw_3$ 并在该光路上分配 1-3 三个频谱带宽槽，然后，它以 FlowMod 消息的形式将流表信息下发给对应“交换机”。E-OVX 截获这些 FlowMod 消息检测这些消息的合法性，然后对它们进行相应改写（即，虚

*图 5.12 中第 2 个数据包虽然是控制器直接发给主机 h_1 ，但是，由于 E-OVX 中不支持以控制器为源给主机发包，因此，我们采用了用控制器以目的节点（即，主机 h_3 ）的 MAC 为源地址给主机 h_1 回复了一条光路建立成功的数据包。

[†]为了方便展示，此截图将大部分无关数据包省略了。

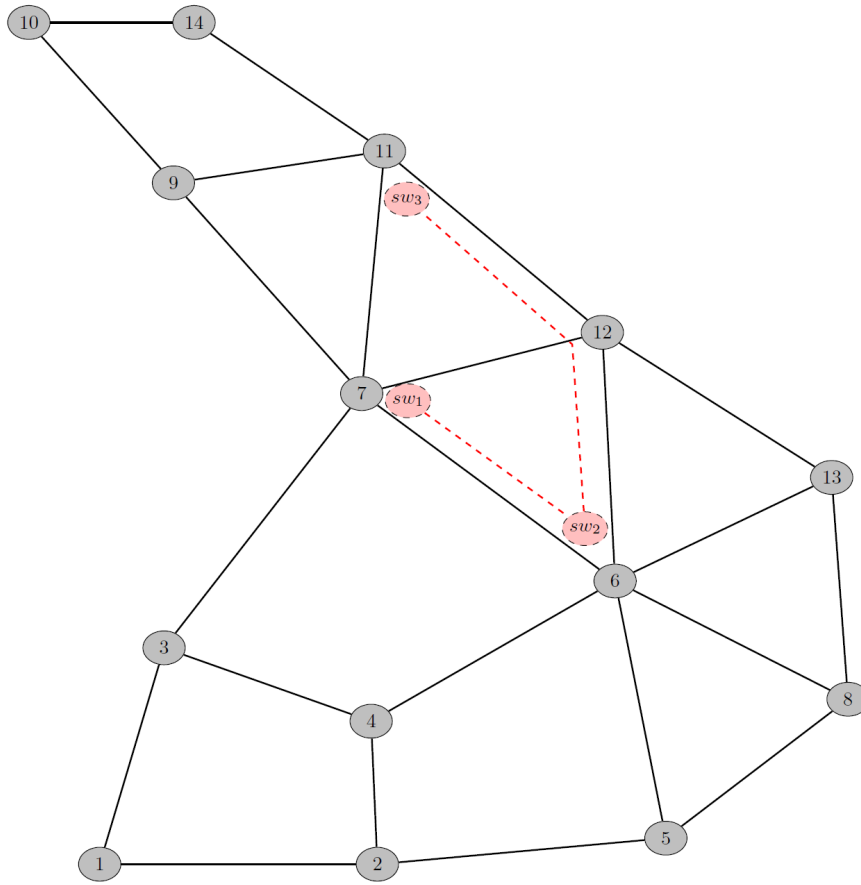


图 5.14: 图5.10所示虚拟软件定义网络的映射关系

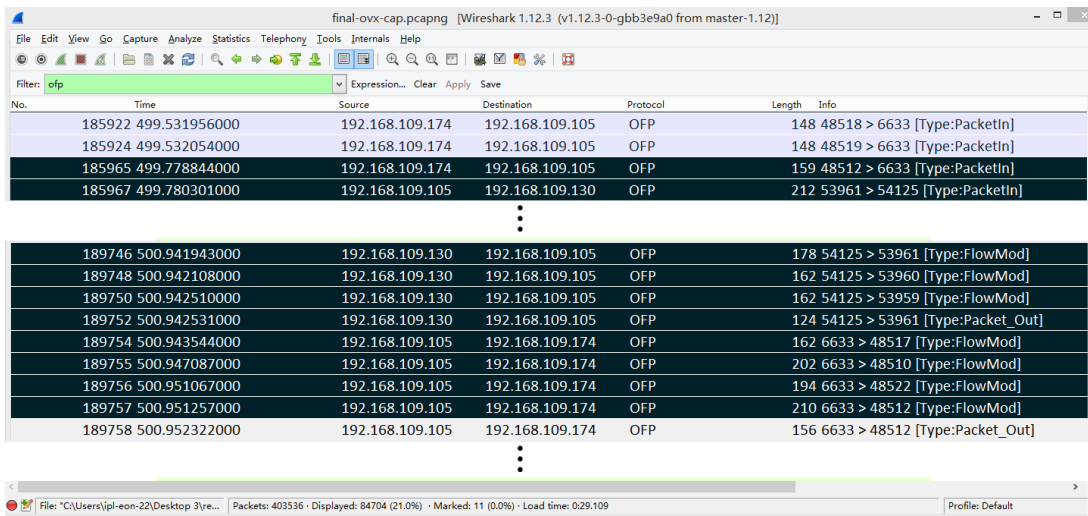


图 5.15: E-OVX 上 Wireshark 抓包截图

拟化)，然后发送给对应底层交换机。

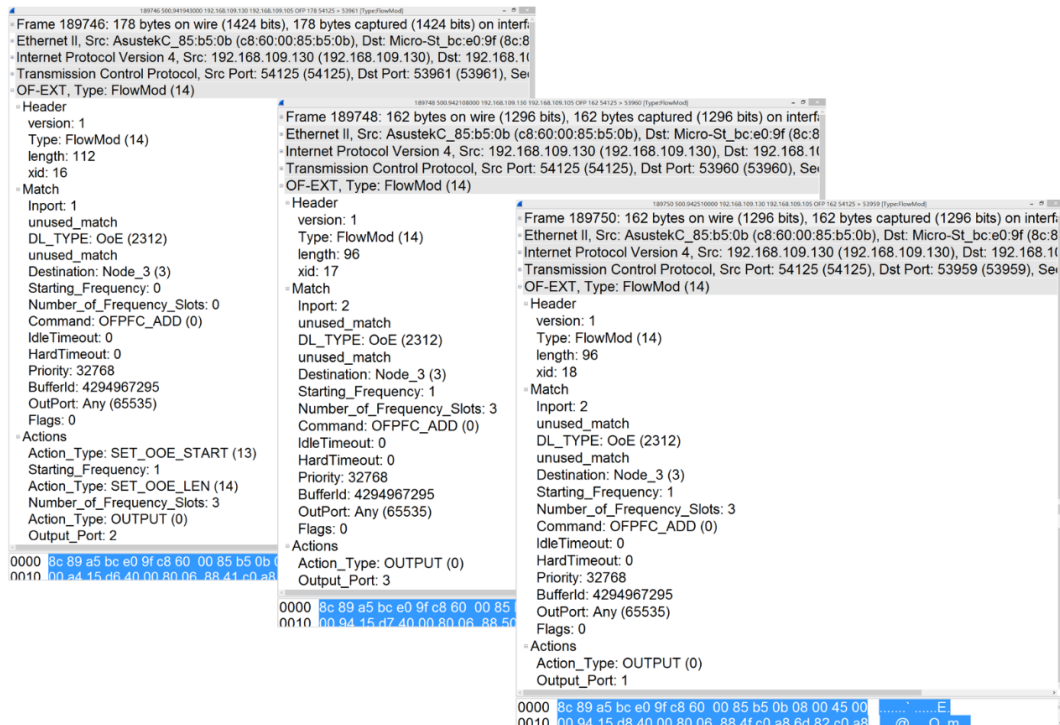


图 5.16: POX 发送给 E-OVX 的 FlowMod 消息

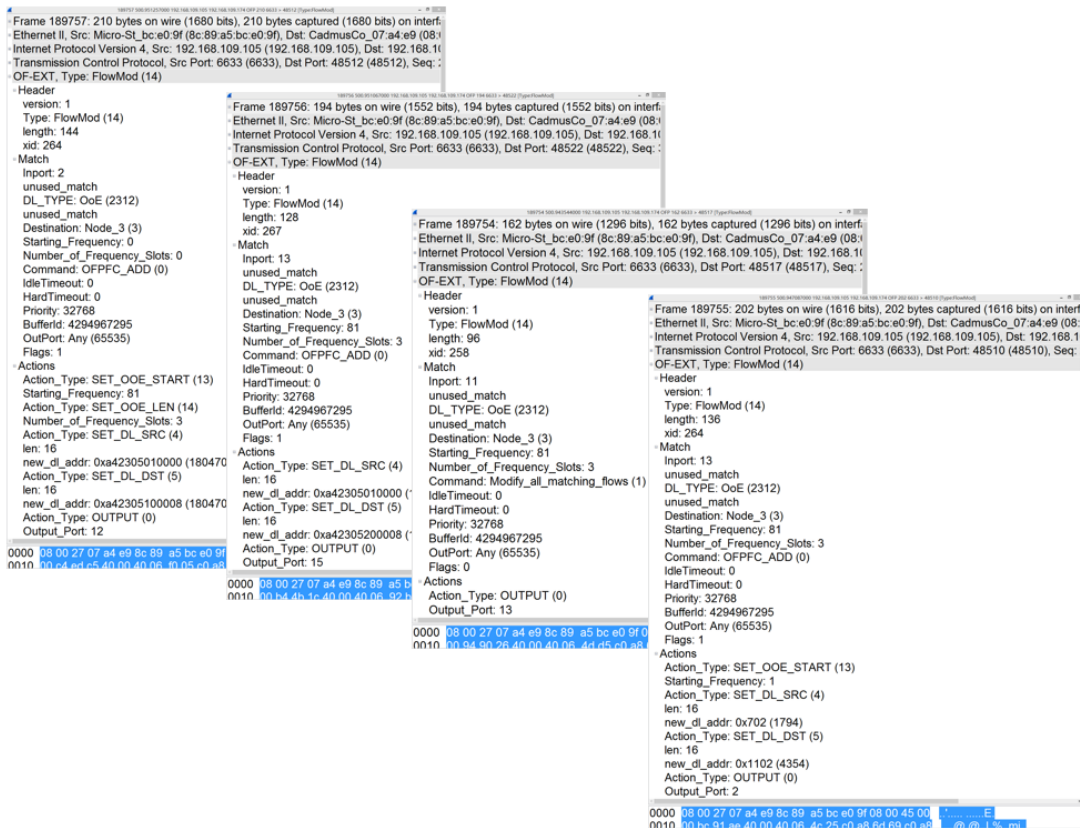


图 5.17: E-OVX 发送给底层网络的 FlowMod 消息

```

mininet> dpctl dump-flows
*** sw_1 -----
NXST_FLOW reply (xid=0x4):
*** sw_2 -----
NXST_FLOW reply (xid=0x4):
*** sw_3 -----
NXST_FLOW reply (xid=0x4):
*** sw_4 -----
NXST_FLOW reply (xid=0x4):
*** sw_5 -----
NXST_FLOW reply (xid=0x4):
*** sw_6 -----
NXST_FLOW reply (xid=0x4):
*** sw_7 -----
NXST_FLOW reply (xid=0x4):
*** sw_8 -----
NXST_FLOW reply (xid=0x4):
*** sw_9 -----
NXST_FLOW reply (xid=0x4):
*** sw_10 -----
NXST_FLOW reply (xid=0x4):
*** sw_11 -----
NXST_FLOW reply (xid=0x4):
*** sw_12 -----
NXST_FLOW reply (xid=0x4):
*** sw_13 -----
NXST_FLOW reply (xid=0x4):
*** sw_14 -----
NXST_FLOW reply (xid=0x4):
mininet> dpctl dump-flows
*** sw_1 -----
NXST_FLOW reply (xid=0x4):
*** sw_2 -----
NXST_FLOW reply (xid=0x4):
*** sw_3 -----
NXST_FLOW reply (xid=0x4):
*** sw_4 -----
NXST_FLOW reply (xid=0x4):
*** sw_5 -----
NXST_FLOW reply (xid=0x4):
*** sw_6 -----
NXST_FLOW reply (xid=0x4):
cookie=0x100000002, duration=50.729s, table=0, n_packets=10, n_bytes=2641, ooe_in_port=13, dl_src=a4:23:05:01:00:00, dl_dst=a4:23:05:10:00:08, ooe_dst=0x3, ooe_start=81, ooe_length=3 actions=mod_dl_src:a4:23:05:01:00:00, mod_dl_dst:a4:23:05:20:00:08, output:15
*** sw_7 -----
NXST_FLOW reply (xid=0x4):
cookie=0x100000002, duration=50.805s, table=0, n_packets=10, n_bytes=2641, ooe_in_port=2, dl_src=00:00:00:00:07:02, dl_dst=00:00:00:00:11:02, ooe_dst=0x3 actions=mod_ooe_start:81, mod_ooe_length:3, mod_dl_src:a4:23:05:01:00:00, mod_dl_dst:a4:23:05:10:00:08, output:12
*** sw_8 -----
NXST_FLOW reply (xid=0x4):
*** sw_9 -----
NXST_FLOW reply (xid=0x4):
*** sw_10 -----
NXST_FLOW reply (xid=0x4):
*** sw_11 -----
NXST_FLOW reply (xid=0x4):
cookie=0x100000002, duration=51.098s, table=0, n_packets=10, n_bytes=2641, ooe_in_port=13, dl_src=a4:23:05:01:00:00, dl_dst=a4:23:05:20:00:08, ooe_dst=0x3, ooe_start=81, ooe_length=3 actions=mod_ooe_start:1, mod_dl_src:00:00:00:00:07:02, mod_dl_dst:00:00:00:00:11:02, output:2
*** sw_12 -----
NXST_FLOW reply (xid=0x4):
cookie=0x100000000, duration=51.154s, table=0, n_packets=10, n_bytes=2641, ooe_in_port=11, dl_src=a4:23:05:01:00:00, dl_dst=a4:23:05:20:00:08, ooe_dst=0x3, ooe_start=81, ooe_length=3 actions=output:13
*** sw_13 -----
NXST_FLOW reply (xid=0x4):
*** sw_14 -----
NXST_FLOW reply (xid=0x4):
mininet> dpctl dump-flows
*** sw_1 -----
NXST_FLOW reply (xid=0x4):
*** sw_2 -----
NXST_FLOW reply (xid=0x4):
*** sw_3 -----
NXST_FLOW reply (xid=0x4):
*** sw_4 -----
NXST_FLOW reply (xid=0x4):
*** sw_5 -----
NXST_FLOW reply (xid=0x4):
*** sw_6 -----
NXST_FLOW reply (xid=0x4):
*** sw_7 -----
NXST_FLOW reply (xid=0x4):
*** sw_8 -----
NXST_FLOW reply (xid=0x4):
*** sw_9 -----
NXST_FLOW reply (xid=0x4):
*** sw_10 -----
NXST_FLOW reply (xid=0x4):
*** sw_11 -----
NXST_FLOW reply (xid=0x4):
*** sw_12 -----
NXST_FLOW reply (xid=0x4):
cookie=0x100000000, duration=265.554s, table=0, n_packets=26, n_bytes=6726, ooe_in_port=11, dl_src=a4:23:05:01:00:00, dl_dst=a4:23:05:20:00:08, ooe_dst=0x3, ooe_start=81, ooe_length=3 actions=output:13
*** sw_13 -----
NXST_FLOW reply (xid=0x4):
*** sw_14 -----
NXST_FLOW reply (xid=0x4):

```

图 5.18: 实验过程中流表信息

图5.16和图5.17分别展示了从用户 POX 控制器发送到 E-OVX 的 FlowMod 消息和从 E-OVX 到底层交换机的 FlowMod 消息。图5.18展示了整个实验过程中所有底层交换机上的流表信息*。其中，蓝色方框 1 是在主机 h_1 向 h_3 开始发包之前的流表状态，我们可以看到此时整个网络中所有交换机上都没有流表。蓝色方框 2 展示了 h_1 发包过程中的流表信息，此时我们可以看到物理交换机 sw_7 , sw_6 , sw_{12} 和 sw_{11} 上出现了流表，从流表信息我们可以得出该虚拟网络的映射关系，容易验证该映射关系就是我们从虚拟网络映射器上得到的映射关系（即，图5.14所示的映射关系）。结合图5.16和图5.17的 FlowMod 消息以及图5.18中蓝色方框 2 中的流表信息，我们可以看到用户控制器 POX 在接收到 PacketIn 消息并计算好路由和频谱分配之后，为了建立光路 $sw_1 \rightarrow sw_2 \rightarrow sw_3$ 并在其上分配 1-3 的连续频谱块，对三台虚拟交换机下发了图5.16所示的 FlowMod 消息，而 E-OVX 截获这些 FlowMod 消息之后，根据存储的映射关系将这些 FlowMod 消息改写之后，发送到对应物理交换机。具体而言，E-OVX 首先将发送给入口交换机（即，物理交换机 sw_7 ）的 FlowMod 消息中的行为 Set OoE_Start 的数值改写成了物理网络中对应的值（即,81），此外，它还改写了出、入端口以及源、目的 MAC 地址和匹配项中的 OoE_Start 的匹配值。对于发送给中间交换机 sw_6 的 FlowMod 消息，它则只是改写了匹配项中的源、目的 MAC 地址以及 OoE_Start 的匹配值。对于目的交换机 sw_{11} ，它则增加了行为 Set OoE_Start 将物理网络中的起始频谱带宽槽改写回虚拟网络中对应的值。此外，它还在虚拟链路经过的非映射物理交换机 sw_{12} 上，下发了一个修改流表的 FlowMod 消息。这就证明了我们的系统确实实现了弹性光网络虚拟化的功能。

5.4.3 系统性能测试

在上一小节我们通过一个简单的试验验证了我们的系统能够实现弹性光网络的虚拟化，这本小节，我们采用图4.2所示的 50 点随机拓扑为底层软件定义弹性光网络来测试我们系统的性能。实验中，底层网络中的资源设置为，每个节点和每根光纤链路的初始计算资源容量和带宽资源容量分别为 200 单位和 200 FS'。对于虚拟网络，我们也采用 GT-ITM 工具随机生成，节点数目为 2-10，服从均匀分布，任意两个节点之间有边的概率均为 0.5，每个节点和链路的计算资源和带宽资源需求分别为 1-20 单位和 1-20 FS'，均服从均匀分布，为了保证实验结果的可靠性，我们一共生成了 9000 个虚拟网络。我们通过测试我们系统所提供的四种主要操作（即，创建、启动、停止和删除虚拟网络）的耗时长度来衡量我们系统的性能。整个实验过程如下，我们以 10 个虚拟网络为一组，首先依次创建 10 个虚拟网络，然后，再依次启动它们，接下来再依次停止，最后再将它们删除。一组完成之后，继续进入下一组，直至 9000 个虚拟网络请求全部处理完。

*实验开始之前，我们先将所有流表清空了。

需要注意的是，正如本章第5.3小节所述，删除操作实际上可能包括停止操作，而从实验流程很容易知道，上述实验的所有删除操作都是不包括停止操作的。

表 5.2: 弹性光网络虚拟化平台四种主要操作所需时间（单位：毫秒）

操作 指标	创建	启动	停止	删除
平均时间	129.650	5105.110	4215.880	161.488
最短时间	94.247	1616.887	1539.228	108.545
最长时间	601.168	22034.953	14446.369	1226.696
标准差	27.852	2533.605	1916.246	73.625

表格5.2统计了这四种主要操作所需时间的平均值、最小值、最大值以及标准差。图5.19展示了这四种主要操作所需要的时间（即，从用户按下对应按钮到用户收到反馈）的累积概率分布。结合表格5.2和图5.19, 我们可以得到，该系统创建、启动、停止和删除一个虚拟网络平均各需要约 130 毫秒，5 秒，4 秒和 160 毫秒，最快分别需要约 90 毫秒，2 秒，1.5 秒和 100 毫秒，最慢需要约 600 毫秒，22 秒，14 秒和 1.2 秒。其中，90% 以上的请求均可以在 170 毫秒内创建，9 秒内启动，7 秒内停止，175 毫秒内删除。另外，通过观察四条曲线的走向和四种操作的标准差，我们可以看到由于启动和删除操作只涉及管理系统不需要与其他模块通信同时也不需要进行任何计算，因此时间相比于启动和停止而言短了很多，而且它们的曲线非常陡，它们的标准差也相对较小，说明这两个操作与虚拟网络的规模关系很小。然而，对于启动和停止，均需要管理系统、E-OVX 用户控制器以及虚拟网络映射器共同完成，而启动虚拟网络映射器上的虚拟网络映射以及 E-OVX 根据映射关系进行相关操作的过程中都与虚拟网络的规模有关系，因此，启动与停止操作的概率累积分布曲线相对比较平缓，标准差也相对较大。

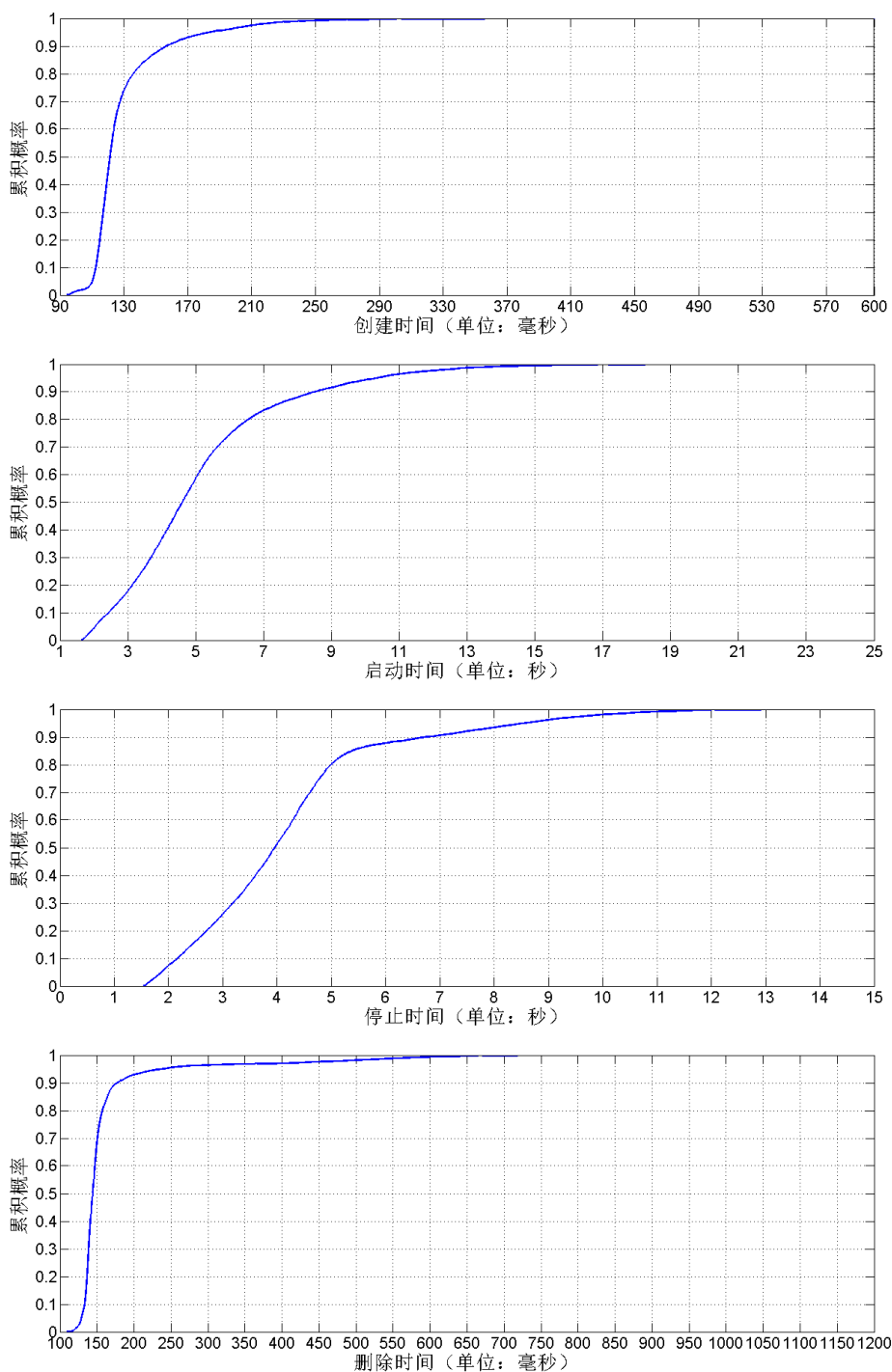


图 5.19: 弹性光网络虚拟化平台四种主要操作所需时间的累积概率分布图

第六章 总结与未来工作展望

6.1 总结

本课题研究了跨数据中心弹性光网络环境下的透明虚拟光网络映射的问题。考虑到该问题本质上是一个带有额外限制的多品流问题，我们为这个问题建立了一个基于多商品流的整数线性规划模型。然后，我们为这个问题设计了一个高效的启发式算法。为了实现节点与链路映射的协作以及链路映射阶段路由与频谱分配的整合，我们引入了分层辅助图的概念，根据虚拟光网络的带宽需求以及底层光网络的频谱使用情况，我们将底层网络分成多个分层辅助图，借助分层辅助图，我们提出了一个基于分层资源容量和分层最短路径路由的透明虚拟光网络映射算法。通过大量的数值仿真，我们评估了该算法的性能，验证了该算法相比于对比算法能够极大地降低虚拟网络请求的阻塞率。

最后，基于软件定义网络的架构，我们搭建了一个弹性光网络虚拟化平台，并在该平台上实现了我们的启发式算法，通过实验验证了我们的平台确实实现了弹性光网络虚拟化的功能。此外，我们还通过一系列实验测试了该平台的性能。

6.2 未来工作展望

本文对跨数据中心弹性光网络中透明虚拟光网络映射问题展开了深入研究，为该问题建立了数学模型，提出了高效的启发式算法，搭建了试验平台。但是，该工作还有很多不足之处。

- 首先，正如第二章第2.1.2小节所述，本文为了简化问题，假设每一个透明虚拟光网络的所有虚拟光链路都需要相同的带宽资源，并且必须分配同一段频谱资源。这一假设很大程度上限制了我们的算法的应用场景，在未来的工作中，我们将进一步优化问题模型，考虑建立适用性更广的问题模型。
- 其次，正如第四章第4.3小节所述，物理层损伤会严重影响虚拟光网络的性能好坏，而本文并没有考虑物理层的损伤。在未来的工作中，我们将考虑通过结合灵活的调制级别选择来实现对物理层损伤的考虑。
- 另外，正如第一章第1.2小节所述，由于光纤的巨大带宽以及数据中心相对比较“脆弱”，因此，考虑虚拟光网络的生成性显得尤为必要。然而，本文却没有考虑可生成的虚拟光网络映射。在未来的工作中，我们将从恢复和保护两个角度来考虑可生成虚拟光网络映射问题。
- 最后，从本文第五章的系统描述和试验结果，我们可以看出，我们的试验平台虽然实现了弹性光网虚拟化的功能，但是，该平台并没有实现对数据

中心的支持，另外，该平台的性能还有待提高。在未来的工作中，我们准备从以下几个方面来进一步完善该实验平台。

- 将云平台 Openstack^[112] 整合到我们的虚拟化平台实现对数据中心的支持。
- 优化数据结构，降低各个模块之间的通信开销，从而提高系统性能。
- 升级其中 OF 的版本，提高该平台的适用性。

参考文献

- [1] Armbrust M, Fox A, Griffith R, et al. A View of Cloud Computing. *Communications of the ACM*, 2010, 53(4):50–58.
- [2] Amazon Elastic Compute Cloud (EC2) - Scalable Cloud Hosting. http://aws.amazon.com/ec2/?nc2=h_ls.
- [3] Google Cloud Computing, Hosting Services & Cloud Support – Google Cloud Platform. <https://cloud.google.com/>.
- [4] Microsoft Azure: 云计算平台与服务. <http://azure.microsoft.com/>.
- [5] IBM Cloud Services. <http://www-935.ibm.com/services/us/en/it-services/cloud-services/>.
- [6] 阿里云 -打造数据分享第一平台. <http://www.aliyun.com/>.
- [7] 云服务 - 百度开放云平台. <http://developer.baidu.com/cloud>.
- [8] 新浪云计算. <http://www.sinacloud.com/>.
- [9] Greenberg A, Hamilton J, Maltz D A, et al. The Cost of a Cloud: Research Problems in Data Center Networks. *ACM SIGCOMM Computer Communication Review*, 2008, 39(1):68–73.
- [10] Chen Y, Jain S, Adhikari V, et al. A first look at inter-data center traffic characteristics via Yahoo! datasets. *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2011. 1620–1628.
- [11] Liu C, Kind A, Vasilakos A. Sketching the data center network traffic. *IEEE Network*, 2013, 27(4):33–39.
- [12] Gerstel O, Jinno M, Lord A, et al. Elastic optical networking: a new dawn for the optical layer? *IEEE Communications Magazine*, 2012, 50:S12–S20.
- [13] Develder C, De Leenheer M, Dhoedt B, et al. Optical Networks for Grid and Cloud Computing Applications. *Proceedings of the IEEE*, 2012, 100(5):1149–1167.
- [14] Chowdhury N M K, Boutaba R. Network virtualization: state of the art and research challenges. *IEEE Communications Magazine*, 2009, 47(7):20–26.
- [15] Layec P, Morea A, Vacondio F, et al. Elastic optical networks: The global evolution to software configurable optical networks. *Bell Labs Technical Journal*, 2013, 18(3):133–151.
- [16] Zhang G, De Leenheer M, Morea A, et al. A Survey on OFDM-Based Elastic Core Optical Networking. *IEEE Communications Surveys & Tutorials*, 2013, 15(1):65–87.
- [17] Mukherjee B. *Optical WDM Networks (Optical Networks)*. Springer-Verlag New York, Inc., 2006.
- [18] Takara H, Goh T, Shibahara K, et al. Experimental demonstration of 400 Gb/s multi-flow, multi-rate, multi-reach optical transmitter for efficient elastic spectral routing. *Proceedings of IEEE/OSA European Conference and Exhibition on Optical Communication (ECOC)*, 2011. 1–3.
- [19] Bocoï A, Schuster M, Rambach F, et al. Reach-Dependent Capacity in Optical Networks Enabled by OFDM. *Proceedings of IEEE/OSA Optical Fiber Communication Conference (OFC)*, 2009. 1–3.

- [20] Zang H, Jue J P. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 2000, 1:47–60.
- [21] Christodoulopoulos K, Tomkos I, Varvarigos E. Elastic Bandwidth Allocation in Flexible OFDM-Based Optical Networks. *IEEE/OSA Journal of Lightwave Technology*, 2011, 29:1354–1366.
- [22] Gong L, Zhou X, Lu W, et al. A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks. *IEEE Communications Letters*, 2012, 16:1520–1523.
- [23] Wei Y, Shen G, Bose S. Span-Restorable Elastic Optical Networks Under Different Spectrum Conversion Capabilities. *IEEE Transactions on Reliability*, 2014, 63(2):401–411.
- [24] Zhu Z, Chen X, Ji F, et al. Energy-Efficient Translucent Optical Transport Networks With Mixed Regenerator Placement. *IEEE/OSA Journal of Lightwave Technology*, 2012, 30(19):3147–3156.
- [25] Ramamurthy B, Feng H, Datta D, et al. Transparent vs. opaque vs. translucent wavelength-routed optical networks. *Proceedings of IEEE/OSA Optical Fiber Communication Conference (OFC)*, volume 1, 1999, 59–61 vol.1.
- [26] Turner J, Taylor D. Diversifying the Internet. *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, volume 2, 2005, 1–6.
- [27] Anderson T, Peterson L, Shenker S, et al. Overcoming the Internet impasse through virtualization. *IEEE Computer*, 2005, 38(4):34–41.
- [28] Feamster N, Gao L, Rexford J. How to Lease the Internet in Your Spare Time. *ACM SIGCOMM Computer Communication Review*, 2007, 37(1):61–64.
- [29] Network Virtualization vs. Software-Defined Networking (SDN). <http://www.vmware.com/software-defined-datacenter/networking-security>.
- [30] Network Virtualization Solutions. <http://www.cisco.com/c/en/us/solutions/enterprise-networks/network-virtualization-solutions/index.html>.
- [31] HP Software-defined Networking and Network Virtualization. <http://h17007.www1.hp.com/us/en/networking/solutions/technology/sdn/network-virtualization.aspx#.VR6aifyUeVg>.
- [32] Windows Network Virtualization. [https://msdn.microsoft.com/zh-cn/library/windows/desktop/jj673986\(v=vs.85\).aspx](https://msdn.microsoft.com/zh-cn/library/windows/desktop/jj673986(v=vs.85).aspx).
- [33] Chowdhury N, Boutaba R. A survey of network virtualization. *Computer Networks*, 2010, 54:862–876.
- [34] Scott C, Wolfe P, Erwin M. *Virtual private networks*. O'Reilly Media, Inc., 1999.
- [35] Fischer A, Botero J, Beck M, et al. Virtual Network Embedding: A Survey. *IEEE Communications Surveys & Tutorials*, 2013, 15:1888–1906.
- [36] Zhang S, Wu J, Lu S. Virtual network embedding with substrate support for parallelization. *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2012, 2615–2620.
- [37] Zhao J, Subramaniam S, Brandt-Pearce M. Virtual topology mapping in elastic optical networks. *Proceedings of IEEE International Conference on Communications (ICC)*, 2013, 3904–3908.

-
- [38] Zhu Y, Ammar M. Algorithms for Assigning Substrate Network Resources to Virtual Network Components. *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2006. 1–12.
- [39] Yu M, Yi Y, Rexford J, et al. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. *ACM SIGCOMM Computer Communication Review*, 2008, 28:17–29.
- [40] Houidi I, Louati W, Zeglache D. A Distributed Virtual Network Mapping Algorithm. *Proceedings of IEEE International Conference on Communications (ICC)*, 2008. 5634–5640.
- [41] Lischka J, Karl H. A Virtual Network Mapping Algorithm Based on Subgraph Isomorphism Detection. *Proceedings of ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA)*, New York, NY, USA: ACM, 2009. 81–88.
- [42] Farooq Butt N, Chowdhury M, Boutaba R. Topology-Awareness and Reoptimization Mechanism for Virtual Network Embedding. In: Crovella M, Feeney L, Rubenstein D, et al., (eds.). *Proceedings of IFIP International Conference on Networking*. Springer Berlin Heidelberg, 2010: 27–39.
- [43] Houidi I, Louati W, Zeglache D, et al. Adaptive Virtual Network Provisioning. *Proceedings of ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA)*, New York, NY, USA: ACM, 2010. 41–48.
- [44] Yu H, Qiao C, Anand V, et al. Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures. *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2010. 1–6.
- [45] Chowdhury M, Samuel F, Boutaba R. PolyViNE: Policy-based Virtual Network Embedding Across Multiple Domains. *Proceedings of ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA)*, New York, NY, USA: ACM, 2010. 49–56.
- [46] Fajjari I, Aitsaadi N, Pujolle G, et al. VNE-AC: Virtual Network Embedding Algorithm Based on Ant Colony Metaheuristic. *Proceedings of IEEE International Conference on Communications (ICC)*, 2011. 1–6.
- [47] Yu H, Anand V, Qiao C, et al. Cost Efficient Design of Survivable Virtual Infrastructure to Recover from Facility Node Failures. *Proceedings of IEEE International Conference on Communications (ICC)*, 2011. 1–6.
- [48] Guo T, Wang N, Moessner K, et al. Shared Backup Network Provision for Virtual Network Embedding. *Proceedings of IEEE International Conference on Communications (ICC)*, 2011. 1–5.
- [49] Cheng X, et al. Virtual network embedding through topology-aware node ranking. *ACM SIGCOMM Computer Communication Review*, 2011, 41:39–47.
- [50] Houidi I, Louati W, Ameer W B, et al. Virtual network provisioning across multiple substrate networks. *Computer Networks*, 2011, 55(4):1011 – 1023. Special Issue on Architectures and Protocols for the Future Internet.
- [51] Chowdhury M, Rahman M, Boutaba R. ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping. *IEEE/ACM Transactions on Networking*, 2012, 20:206–219.
- [52] Papagianni C, Leivadreas A, Papavassiliou S, et al. On the optimal allocation of virtual resources in cloud computing networks. *IEEE Transactions on Computers*, 2013, 62:1060–1071.

-
- [53] Hu Q, Wang Y, Cao X. Resolve the virtual network embedding problem: A column generation approach. Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2013. 410–414.
- [54] Habib M, Tornatore M, Mukherjee B. Fault-tolerant virtual network mapping to provide Content Connectivity in optical networks. Proceedings of IEEE/OSA Optical Fiber Communication Conference (OFC), 2013. 1–3.
- [55] Gu F, Peng M, Khan S, et al. Virtual network reconfiguration in optical substrate networks. Proceedings of IEEE/OSA Optical Fiber Communication Conference (OFC), 2013. 1–3.
- [56] Zhang S, Qian Z, Wu J, et al. Virtual Network Embedding with Opportunistic Resource Sharing. IEEE Transactions on Parallel and Distributed Systems, 2014, 25:816–827.
- [57] Yu H, Qiao C, Wang J, et al. Regional failure-resilient virtual infrastructure mapping in a federated computing and networking system. IEEE/OSA Journal of Optical Communications and Networking, 2014, 6(11):997–1007.
- [58] Yoshinari M, Ohsita Y, Murata M. Virtual network reconfiguration with adaptability to traffic changes. IEEE/OSA Journal of Optical Communications and Networking, 2014, 6(6):523–535.
- [59] Gong L, Wen Y, Zhu Z, et al. Toward Profit-Seeking Virtual Network Embedding Algorithm via Global Resource Capacity. Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2014. 1–9.
- [60] Nejabati R, Escalona E, Peng S, et al. Optical network virtualization. Proceedings of IEEE Conference on Optical Networking Design and Modeling (ONDM), 2011. 1–5.
- [61] Jinno M. Virtualization in optical networks: from elastic networking level to sliceable equipment level. Proceedings of IEEE International Conference on Optical Internet (COIN), 2012. 61–62.
- [62] Peng S, Nejabati R, Simeonidou D. Role of optical network virtualization in cloud computing [invited]. IEEE Network, 2013, 5(10):A162–A170.
- [63] Belbekkouche A, Hasan M, Karmouch A. Resource Discovery and Allocation in Network Virtualization. IEEE Communications Surveys & Tutorials, 2012, 14:1114–1128.
- [64] Wang Z, Han Y, Lin T, et al. Virtual network embedding by exploiting topological information. Proceedings of IEEE Global Communications Conference (GLOBECOM), 2012. 2603–2608.
- [65] Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. Proceedings of International Conference on World Wide Web, 1998. 107–117.
- [66] Freeman L. Centrality in social networks: Conceptual clarification. Social Networks, 1979, 1:215–239.
- [67] Ahuja R, Magnanti T, Orlin J. Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., 1993.
- [68] Schrijver A. Theory of Linear and Integer Programming. New York, NY, USA: John Wiley & Sons, Inc., 1986.
- [69] Barnhart C, Johnson E, Nemhauser G, et al. Branch-and-price: Column generation for solving huge integer programs. Operations Research, 1998, 46:316–329.

- [70] Rahman M, Boutaba R. SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization. *IEEE Transactions on Network and Service Management*, 2013, 10(2):105–118.
- [71] Jarray A, Song Y, Karmouch A. p-Cycle-based node failure protection for survivable virtual network embedding. *Proceedings of International Federation for Information Processing (IFIP)*, 2013. 1–9.
- [72] Jarray A, Karmouch A. Column generation approach for one-shot virtual network embedding. *Proceedings of IEEE Global Communications Conference (GLOBECOM) Workshops*, 2012. 863–868.
- [73] Grover W, Stamatelakis D. Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration. *Proceedings of IEEE International Conference on Communications (ICC)*, volume 1, 1998. 537–543.
- [74] Yu H, Qiao C, Anand V, et al. Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures. *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2010. 1–6.
- [75] Yu H, Anand V, Qiao C, et al. Cost Efficient Design of Survivable Virtual Infrastructure to Recover from Facility Node Failures. *Proceedings of IEEE International Conference on Communications (ICC)*, 2011. 1–6.
- [76] Guo B, Qiao C, Wang J, et al. Survivable Virtual Network Design and Embedding to Survive a Facility Node Failure. *IEEE/OSA Journal of Lightwave Technology*, 2014, 32(3):483–493.
- [77] Zhang S, Shi L, Vadrevu C, et al. Network virtualization over WDM networks. *Proceedings of IEEE International Symposium on Advanced Networks and Telecommunication Systems (ANTS)*, 2011. 1–3.
- [78] Elmirghani J, Mouftah H. All-optical wavelength conversion: technologies and applications in DWDM networks. *IEEE Communications Magazine*, 2000, 38(3):86–92.
- [79] Peng S, Nejabati R, Azodolmolky S, et al. An impairment-aware virtual optical network composition mechanism for future Internet. *Optics Express (OpEx)*, 2011, 19:B251–B259.
- [80] Pages A, Perello J, Spadaro S. Virtual network embedding in optical infrastructures. *Proceedings of IEEE International Conference on Transparent Optical Networks*, 2012. 1–4.
- [81] Pages A, Perello J, Spadaro S, et al. Optimal allocation of virtual optical networks for the future Internet. *Proceedings of IEEE Optical Networks Design and Modelling (ONDM) Conference*, 2012. 1–6.
- [82] Hammad A, Nejabati R, Simeonidou D. Novel approaches for composition of online virtual optical networks utilizing O-OFDM technology. *Proceedings of IEEE/OSA European Conference and Exhibition on Optical Communication (ECOC)*, 2013. 1–3.
- [83] Shakya S, Pradhan N, Cao X, et al. Virtual network embedding and reconfiguration in elastic optical networks. *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2014. 2160–2165.
- [84] Chen B, Zhang J, Xie W, et al. Minimum-cost survivable virtual optical network mapping in flexible bandwidth optical networks. *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2014. 2023–2028.
- [85] Xie W, Jue J, Zhang Q, et al. Survivable virtual optical network mapping in flexible-grid optical networks. *Proceedings of IEEE International Conference on Computing, Networking and Communications (ICNC)*, 2014. 221–225.

- [86] Feamster N, Rexford J, Zegura E. The Road to SDN. *Queue*, 2013, 11(12):20:20–20:40.
- [87] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2):69–74.
- [88] Rob Sherwood K K Y G A M C N M, Parulkary G. FlowVisor: A Network Virtualization Layer. Technical report, Stanford University, 2009.
- [89] Drutskey D, Keller E, Rexford J. Scalable Network Virtualization in Software-Defined Networks. *IEEE Internet Computing*, 2013, 17(2):20–27.
- [90] Al-Shabibi A, De Leenheer M, Gerola M, et al. OpenVirteX: Make Your Virtual SDNs Programmable. Proceedings of the Third Workshop on Hot Topics in Software Defined Networking (HotSDN), New York, NY, USA: ACM, 2014. 25–30.
- [91] OpenFlow Switch Specification Version 1.0.0. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf>.
- [92] Basta A, Blenk A, Lai Y T, et al. HyperFlex: Demonstrating Control-plane Isolation for Virtual Software-Defined Networks. Proceedings of IFIP/IEEE Symposium on Integrated Network and Service Management, 2015. 1–9.
- [93] Pages A, Perello J, Spadaro S, et al. Strategies for Virtual Optical Network Allocation. *IEEE Communications Letters*, 2012, 16:268–271.
- [94] Azodolmolky S, et al. Experimental Demonstration of an Impairment Aware Network Planning and Operation Tool for Transparent/Translucent Optical Networks. *IEEE/OSA Journal of Lightwave Technology*, 2011, 29:439–448.
- [95] Zegura E, Calvert K, Bhattacharjee S. How to model an internetwork. Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 1996. 594–602.
- [96] Peng S, Nejabati R, Simeonidou D. Impairment-Aware Optical Network Virtualization in Single-Line-Rate and Mixed-Line-Rate WDM Networks. *IEEE/OSA Journal of Optical Communications and Networking*, 2013, 5:283–293.
- [97] Klekamp A, Dischler R, Buchali F. Transmission reach of Optical-OFDM superchannels with 10 - 600 Gb/s for transparent bit-rate adaptive networks. Proceedings of IEEE/OSA European Conference and Exhibition on Optical Communication (ECOC), 2011. 1–3.
- [98] Zhu Z, Chen X, Chen C, et al. OpenFlow-assisted online defragmentation in single-/multi-domain software-defined elastic optical networks [invited]. *IEEE/OSA Journal of Optical Communications and Networking*, 2015, 7(1):A7–A15.
- [99] Zhu Z, Chen C, Chen X, et al. Demonstration of Cooperative Resource Allocation in an OpenFlow-Controlled Multidomain and Multinational SD-EON Testbed. *IEEE/OSA Journal of Lightwave Technology*, 2015, 33(8):1508–1514.
- [100] POX Wiki. <https://openflow.stanford.edu/display/ONL/POX+Wiki>.
- [101] Floodlight. <http://www.projectfloodlight.org/floodlight/>.

- [102] Bootstrap: The world's most popular mobile-first and responsive front-end framework. <http://getbootstrap.com/>.
- [103] Django: The Web framework for perfectionists with deadlines. <https://www.djangoproject.com/>.
- [104] OpenVirtX: A High-level View. <http://ovx.onlab.us/wp-content/uploads/2014/04/Figure1-2.png>.
- [105] Oracle VM VirtualBox. <https://www.virtualbox.org/>.
- [106] MySQL: The world's most popular open source database. <https://www.mysql.com/>.
- [107] The Apache HTTP Server Project. <http://httpd.apache.org/>.
- [108] Lantz B, Heller B, McKeown N. A Network in a Laptop: Rapid Prototyping for Software-defined Networks. Proceedings of ACM SIGCOMM Hot Topics in Networks (HotNets) workshops. ACM, 2010. 19:1–19:6.
- [109] Open vSwitch. <http://openvswitch.org/>.
- [110] Twisted. <https://twistedmatrix.com/trac/>.
- [111] Safari: Apple - Support - Downloads. <https://support.apple.com/downloads/#safari>.
- [112] OpenStack: Open Source Cloud Computing Software. <https://www.openstack.org/>.

致 谢

时光飞逝，转眼间硕士生活就将结束。在此论文完成之际，请准许我对我的导师，我的朋友和我的亲戚以及其他帮助过我的人表示最诚挚的感谢。

首先，我想感谢我的导师朱祖劼老师，是他把我带到了未来网络这一研究领域，是他让我懂得了什么是科研，让我体会到了科研的乐趣。他的勤奋、主动和精益求精的“科研三原则”将使我受益终身。

其次，我要感谢新加坡南洋理工大学的文勇刚老师，是他带我走进了网络虚拟化的世界，文老师一丝不苟的严谨作风，踏踏实实的精神，不仅授我以文，而且还教我做人。

再次，我要感谢实验室的师兄、师姐、师弟、师妹们。他们经常和我分享科研思路，帮我解决科研和生活中遇到的各种难题。

然后，我要感谢我的室友对我的学习和生活的关心与帮助。

此外，我要感谢中国科学技术大学为我提供这么优异的科研、学生和生活环境。

最后，向我的父母和亲人致谢，感谢他们对我的理解和支持。

龚龙

2015年4月29日

在读期间发表的学术论文与取得的研究成果

研究工作：

1. 弹性光网络下的单播、组播路由、调制与频谱资源分配。
2. 2/3 网络下的虚拟网络映射。
3. 跨数据中心弹性光网络环境下虚拟网络映射。
4. 可生成虚拟网络映射。

已发表论文：

1. **Long Gong** and Zuqing Zhu, “Virtual Optical Network Embedding (VONE) over Elastic Optical Networks”, IEEE/OSA Journal of Lightwave Technology, vol. 32, no. 3, pp. 450-460, Feb. 2014.
2. **Long Gong**, Xiang Zhou, Wei Lu, Zuqing Zhu, “A Two-Population based Evolutionary Approach for Optimizing Routing, Modulation and Spectrum Assignments (RMSA) in O-OFDM Networks,” IEEE Communications Letters, vol. 16, no. 9, pp. 1520-1523, Sept. 2012.
3. **Long Gong**, Wenwen Zhao, Yonggang Wen, Zuqing Zhu, “Dynamic Transparent Virtual Network Embedding over Elastic Optical Infrastructures,” in Proceedings of IEEE International Conference on Communications (ICC), pp. 1-5, Jun. 2013.
4. **Long Gong**, Xiang Zhou, Xiahe Liu, Wenwen Zhao, Wei Lu, and Zuqing Zhu, “Efficient Resource Allocation for All-Optical Multicasting Over Spectrum-Sliced Elastic Optical Networks”, IEEE/OSA Journal of Optical Communications and Networking, vol. 5, no. 8, 2013.
5. **Long Gong**, Yonggang Wen, Zuqing Zhu, and Tony Lee, “Revenue-Driven Virtual Network Embedding Based on Global Resource Information,” in Proceedings of Global Telecommunications Conference (GLOBECOM), pp. 1-6, Dec. 2013.
6. **Long Gong**, Yonggang Wen, Zuqing Zhu, and Tony Lee, “Toward Profit-Seeking Virtual Network Embedding Algorithm via Global Resource Capacity”, to be appeared in IEEE International Conference on Computer Communications (INFOCOM), 2014.

7. Xiahe Liu, **Long Gong**, and Zuqing Zhu, “Design Integrated RSA for Multicast in Elastic Optical Networks with a Layered Approach” , in Proceedings of IEEE Global Communications Conference (GLOBECOM), pp. 2346-2351, Dec. 2013.
8. Xiahe Liu, **Long Gong**, and Zuqing Zhu, “Spectrum- and Energy-Efficient Multicasting over Multicast-Incapable EONs with Member-Only Flexible Relay” , in Proceedings of IEEE/OSA Asia Communications and Photonics Conference (ACP), Paper AW4H.6, Nov. 2013.
9. Xiahe Liu, **Long Gong**, and Zuqing Zhu, “On the Spectrum-Efficient Overlay Multicast in Elastic Optical Networks Built with Multicast-Incapable Switches” , IEEE Communications Letters, vol. 17, no. 9, pp. 1860-1863, Sept. 2013.
10. Huihui Jiang, **Long Gong**, and Zuqing Zhu, “Efficient Joint Approaches for Location-Constrained Survivable Virtual Network Embedding” , in Proceedings of IEEE Global Communications Conference (GLOBECOM), pp. 1810-1815, Dec. 2014.
11. Mingyang Zhang, Weiran Shi, **Long Gong**, Wei Lu, and Zuqing Zhu, “Bandwidth Defragmentation in Dynamic Elastic Optical Networks with Minimum Traffic Disruptions” , in Proceedings of IEEE International Conference on Communications (ICC), pp. 3894-3898, Jun. 2013.
12. Jingjing Yao, Ping Lu, **Long Gong** and Zuqing Zhu, “On Fast and Coordinated Data Backup in Geo-Distributed Optical Inter-Datacenter Networks” , accepted for publication in IEEE/OSA Journal of Lightwave Technology Apr. 2015
13. Wei Lu, Xiang Zhou, **Long Gong**, Mingyang Zhang, and Zuqing Zhu, “Dynamic Multi-Path Service Provisioning under Differential Delay Constraint in Elastic Optical Networks” , IEEE Communications Letters, vol. 17, no. 1, pp. 158-161, Jan. 2013.
14. Xiang Zhou, Wei Lu, **Long Gong**, and Zuqing Zhu, “Dynamic RMSA in Elastic Optical Networks with an Adaptive Genetic Algorithm,” in Proceedings of IEEE Global Communications Conference (GLOBECOM), pp. 2912-2917, Dec. 2012.
15. Wei Lu, Xiang Zhou, **Long Gong**, and Zuqing Zhu, “Scalable Network Planning for Elastic Optical Orthogonal Frequency Division Multiplexing (OFDM) Networks” , in Proceedings of 8th IEEE/IET International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), pp. 1-4, Jul. 2012.

16. Sheng Shen, Wei Lu, Xiahe Liu, **Long Gong**, and Zuqing Zhu, “Dynamic Advance Reservation Multicast in Data Center Networks over Elastic Optical Infrastructure” , in Proceedings of IEEE/OSA European Conference and Exhibition on Optical Communication (ECOC), Paper P.5.12, Sept. 2013.
17. Wenjian Fang, Minhua Lu, Xiahe Liu, **Long Gong**, and Zuqing Zhu, “Joint Defragmentation of Optical Spectrum and IT Resources in Elastic Optical Datacenter Interconnections” , accepted for publication in IEEE/OSA Journal of Optical Communications and Networking, Feb. 2015.