

工学硕士学位论文

面向网安试验的大规模虚拟网络映射算
法研究

**Research on Large-scale Virtual Network
Mapping for Cyber Security Testbed**

孙 奥

哈尔滨工业大学
2018 年 6 月

国内图书分类号：TP393.0

学校代码：10213

国际图书分类号：621.3

密级：公开

工学硕士学位论文

面向网安试验的大规模虚拟网络映射算法研究

硕士研究生：	孙 奥
导师：	张伟哲 教授
申 请 学 位：	工学硕士
学科、专业：	计算机科学与技术
所在单位：	计算机科学与技术学院
答辩日期：	2018 年 6 月
授予学位单位：	哈尔滨工业大学

Classified Index: TP393.0

U.D.C: 621.3

Dissertation for the Master Degree in Engineering

**Research on Large-scale Virtual Network
Mapping for Cyber Security Testbed**

Candidate:	Sun Ao
Supervisor:	Prof. Zhang Weizhe
Academic Degree Applied for:	Master of Engineering
Speciality:	Computer Science and Technology
Affiliation:	School of Computer Science and Technology
Date of Defence:	June, 2018
Degree-Conferring-Institution:	Harbin Institute of Technology

摘 要

随着网络空间形势日趋严峻，各个国家在网络层面上的对抗程度越来越深，技术手段也越来越先进。网络仿真与效果评估平台，即网络靶场，作为进行网安试验的设施与工具，变得十分重要。搭建网络靶场进行网安试验已成为各国的重要能力，国外的一些靶场已经拥有了万级节点大规模虚拟网络的构建能力。因此，面向网安试验的大规模虚拟网络映射问题是一个亟待解决的问题。

基于这种情况，本文首先根据网安试验的特点，结合传统的虚拟网络映射问题，提出了面向网安试验的大规模虚拟网络映射问题的定义，详细给出了大规模虚拟网络映射问题下的网络模型，数学定义以及评价函数。

然后分析了传统虚拟网络映射算法，传统虚拟网络映射算法包含两个过程，节点之间的映射过程和链路之间的映射过程，其局限性在于网络规模和链路映射。

由此，提出了针对面向网安试验的大规模虚拟网络映射算法。该算法首先进行基于多层 K 划分的前处理过程，通过粗化处理， K 路划分和细化处理减小虚拟网络请求规模。然后进行基于节点契合度的映射过程，在节点映射时采用新的节点排序方式，并计算物理节点和虚拟节点契合度，放大链路的影响因素，旨在提高链路映射的效果。

最后本文建立了一个虚拟网络映射系统，通过模拟映射实验和真实拓扑映射实验，验证本文提出的算法对于大规模虚拟网络映射有着良好的效果。

关键词：大规模虚拟网络映射；图划分；节点排序；节点契合度；收益开销比

Abstract

With the increasingly serious network space situation, the confrontation degree of each country at the network level is deeper and deeper, and the technical means are more and more advanced. Network simulation and effectiveness evaluation platform, that is, cyber security testbed, is an important infrastructure for network attack and defense drill and new network technology evaluation. It has become an important capability for various countries to build a cyber security testbed for network test. Some foreign target ranges have already built up the capacity of large-scale virtual network of 10000 nodes. Therefore, large scale virtual network mapping problem for cyber security testbed is an urgent problem.

Based on this situation, according to the characteristics of the cyber security testbed and the traditional virtual network mapping problem, the thesis introduces the definition of the large-scale virtual network mapping problem first. The network model, the mathematical definition and the evaluation function of the large-scale virtual network mapping are given in detail as well.

Then we analyze the traditional virtual network mapping algorithm. The traditional virtual network mapping algorithm contains two processes, the mapping process between nodes and the mapping process between links. The traditional algorithm is limited to network scale and link mapping.

Therefore, a large-scale virtual network mapping algorithm for cyber security testbed is proposed. The algorithm first performs a pre-processing process based on multi-level K partition, and reduces the size of virtual network requests by coarsening, K path segmentation and thinning. Then, the mapping process based on the degree of node fit is carried out. A new node sorting method is used in the node mapping, and the correspondence degree of physical nodes and virtual nodes is calculated, and the influence factors of the link are amplified to improve the effect of link mapping.

In the end, a virtual network mapping system is established. Through simulation mapping experiment and real topology mapping experiment, it is proved that the algorithm proposed in this paper has good effect on large-scale virtual network mapping.

Keywords: Large-Scale Virtual Network Mapping, Graphic Partitioning, Node Sort, Node Correspondence Degree, Revenue-Cost ratio

目 录

摘 要	I
ABSTRACT	II
第 1 章 绪 论	1
1.1 课题背景及研究的目的和意义	1
1.2 相关工作	2
1.2.1 网络虚拟化	2
1.2.2 虚拟网络映射算法的分类	3
1.2.3 虚拟网络映射算法的策略	4
1.2.4 虚拟网络映射算法的评价	6
1.2.5 虚拟网络映射算法的进展	7
1.3 主要研究内容	8
1.4 论文的组织结构	9
第 2 章 大规模虚拟网络映射问题定义	10
2.1 基本网络模型	10
2.2 映射问题及数学定义	12
2.3 映射结果评价函数	14
2.4 本章小结	15
第 3 章 基于多层 K 划分的前处理过程	17
3.1 前处理过程概述	17
3.2 粗化处理	19
3.3 K 路划分	21
3.4 细化处理	22
3.5 带有前处理过程的经典映射算法	23
3.6 本章小结	25
第 4 章 基于节点契合度的映射过程	26
4.1 算法思路	27
4.2 节点映射过程	29
4.2.1 节点排序	29
4.2.2 节点契合度匹配	30
4.3 链路映射过程	31

4.4 映射算法整体分析	33
4.5 本章小结	35
第 5 章 虚拟网络映射算法的实验研究	36
5.1 虚拟网络映射系统	36
5.1.1 模块划分	36
5.1.2 工作流程	37
5.2 映射对比实验	38
5.2.1 小规模实验	39
5.2.2 中等规模实验	41
5.2.3 大规模实验	44
5.2.4 超大规模实验	47
5.2.5 真实拓扑映射实验	50
5.2.6 实验结果分析	52
5.3 本章小结	52
结 论	53
参考文献	54
攻读硕士学位期间发表的论文及其它成果	58
哈尔滨工业大学学位论文原创性声明和使用权限	59
致 谢	60

第 1 章 绪 论

1.1 课题背景及研究的目的和意义

网络空间对抗形势日趋严峻，各个国家在网络层面上的对抗程度越来越激烈，技术手段越加复杂。在这种情况下，网络仿真与效果评估平台，即网络靶场，作为网络安全试验和网络攻防技术评测的重要基础设施与工具，变得愈加的重要。搭建网络靶场的首要任务就是进行虚拟网络快速灵活构建，这就涉及到了网络虚拟化，而虚拟网络映射算法作为网络虚拟化的核心，通过面向网安试验的大规模虚拟网络映射算法研究，便可以在有限的物理资源上为虚拟网络构建提供部署策略。

加拿大国家仿真实验室建立的 CASELab 网络靶场，支持大规模网络模拟计算，最大实验虚拟机规模达到 1 万级别，支持破坏性测试，支持重复实验。可以看到网安实验中目标网络规模越来越大，已经达到万级以上规模，而高性能模拟器甚至可以模拟十万节点的巨型网络拓扑图，这就需要人们对硬件资源进行更高效的利用；同时目标网络的需求的特异性差异日益扩大，引来对虚拟网络映射问题的研究，需要将不同需求的虚拟网络拓扑映射至有限资源的物理网络中。不同于之前研究的大部分虚拟网络映射问题，本文的目标是在满足物理网络中的各项约束（CPU 和链路带宽）的前提下充分利用物理节点资源，减少网络通信消耗，在有限的时间内完成映射过程。当前关于大规模虚拟网络映射的研究还并不是很多，有待进一步研究。

本文研究的课题是大规模虚拟网络映射的算法，这作为网络虚拟化的核心技术，主要工作就是通过某种算法，满足虚拟网络资源和物理网络资源约束的同时，找到虚拟网络资源分布到物理网络资源的一种方案，并且使不同的虚拟网络在相互隔离的情况下正常工作。大规模问题下，与传统映射问题的差别主要在于规模，大规模问题的虚拟网络规模是大于物理网络规模的。这也更加符合当下的互联网环境，网络的规模不断增加，物理硬件的性能越来越强，一台物理主机往往容纳多台虚拟主机。

对于传统虚拟网络映射算法的研究已经有很多，其中大部分的研究是针对在网络映射过程中，一个虚拟网络节点只能映射到一个物理网络节点的问题，这是一个多目标多约束问题，因为进行物理资源的分配时，需要同时满足虚拟节点和虚拟链路的约束，同时还受到距离和隔离性等方面的限制。但是这种方式往往解决的虚拟网络规模比较小。

目前还有一部分研究想要将虚拟网络映射的问题转化为图划分问题,借助已有的图划分方面的研究来解决现有问题,这种方法的速度快,可解决的虚拟网络规模足够大,但这种思路由于简化了问题模型,虽然能够得到可行解,划分的质量却比较一般。

在大规模虚拟网络映射问题中,研究映射算法的主要目的为,尽可能的映射更多的虚拟网络,并且尽量降低对物理资源的损耗,提高物理资源的使用效率。因为硬件的能力越来越强,一个物理节点往往能够容纳极多的虚拟节点,因此虚拟网络映射成功的关键点往往在于虚拟链路的映射,很多虚拟网络的映射中,都会发生节点的映射成功,但是受制于物理资源的剩余带宽,映射失败。所以本文将提出一种综合考虑节点与链路的映射算法,优化过程,提高映射质量,同时优化映射匹配策略,使物理节点与虚拟节点可以进行重复的映射,使之更加贴合大规模虚拟网络映射的特征,不仅仅满足网安试验的要求,还更符合当前互联网的现状。

1.2 相关工作

解决一个虚拟网络映射问题,需要完成两个过程,一是虚拟节点的映射过程,二是虚拟链路的映射过程,已有很多的研究证明,这是一个 **NP** 难的问题^[1]。即使给定了虚拟节点的映射方案,如何将虚拟链路最优的映射到物理网络这个问题也会归约到不可分流问题上,这也是一个 **NP** 难的问题。因此,这个问题的最优解,仅仅可以在小规模的问题中计算得出,针对大规模的虚拟网络映射问题,主流的解决方案都是采用启发式算法。

接下来,针对虚拟网络映射算法的一些相关研究工作,进行说明和总结,首先于实现网络靶场的核心技术,网络虚拟化,本文给出相关的一些介绍,说明其原理。然后针对虚拟网络映射算法的分类,策略以及评价和进展四个大方面,进行详细的介绍和分析。

1.2.1 网络虚拟化

网络虚拟化^[2-6]的概念早已有之。世界上第一个虚拟局域网大概出现上世纪 90 年代,直到今天,很多国际公司如 Vmware, Google 等都在使用这种不依赖物理设备的技术。最初,网络虚拟技术被设置为一个简单的开关功能,后来也广泛适用于独立的软硬件环境。

一般认为,网络虚拟化就是让一个物理网络能够支持多个虚拟网络,通过在物理设备的基础之上,生成多个相互隔离的虚拟设备,利用虚拟设备来组成虚拟的网

络。对虚拟网络而言，彼此之间是感知不到，每个虚拟网络都认为自己在独享物理资源。因此，对于网络资源如网络地址空间、网络设备容量等，利用网络虚拟化技术，可以提高它们的使用效率。

目前的网络虚拟化技术已经高度的发展，如当下流程的软件定义网络，其实也是一种网络虚拟化的技术手段，这种方法通过软件设置，在一个物理网络上，生成多个虚拟网络，并且虚拟网络的配置随时可以变化。

1.2.2 虚拟网络映射算法的分类

虚拟网络映射问题的特征，根据问题的具体应用场景变化会随之变化。例如，当基础物理设施不变时，虚拟网络映射算法就是面向静态的，考虑了物理设施的变化，虚拟网络映射算法就是动态的；根据虚拟网络请求的分布方式，虚拟网络映射算法又可能是集中于一个节点进行处理的算法，或者是分布式的算法。这些变化的约束条件不是虚拟网络映射算法的特定属性，而是一个具体虚拟网络映射问题的不同变体。因此，在这一小节，根据虚拟网络映射问题的特点，将虚拟网络映射算法分为 6 种类别。

(1) 动态算法或静态算法

根据在映射的过程中，已经映射成功的虚拟网络是否会变化，可以将虚拟网络映射的方式分为动态的或静态的。动态的虚拟网络映射算法再映射时，会根据需要调整已经映射成功的方案寻求最佳映射，而静态的虚拟网络映射算法不会调整已经映射成功的方案，如文献[7-9]。一个动态算法的例子就是 Nguyen^[10]等人提出的动态算法，该算法采用了物理根节点的概念，动态的处理虚拟网络请求，同时没有大幅度的提高复杂度。

随着时间的推移，虚拟请求不断进入或退出，导致物理网络的资源碎片化。或者虚拟网络和物理网络因为自身原因，如拓扑改变，设备升级等等原因，都会导致动态的算法可能效果更好，但本文主要面向网安试验，在这种大规模问题下，不会发生小规模问题中，网络简单重构使得资源空出，映射成功率提高的情况，无法映射的主要原因就是资源不足，通常情况下，只能等待资源。

(2) 集中式算法或分布式算法

在映射时，根据处理虚拟网络请求在一个节点上完成还是在多个节点可以处理请求可以将虚拟网络映射算法分为集中式算法或分布式算法。对于集中式算法，只有一个实体负责虚拟网络嵌入，这种方法的优点在于映射实体在映射的每一步都知道网络的整体情况，即它具有全局拓扑，这有助于获得更好映射的结果，如 A. Razzaq^{[11][12]}等人提出的方式，以及 J. Nogueira^[13]的集中式方案。另一方面，集中算

法存在单点故障，如果映射实体失效，整个映射过程就会失败。此外，当虚拟网络请求量过大时，集中式算法可能处理效率一般。

相对于集中式算法，分布式算法采用多个实体进行嵌入^[14-16]。这种方法的优点在于其更好的可扩展性。并且由于负载分布在多个节点之间，所以每个单独节点将能够更好地处理嵌入。然而，分布式算法必须为分布式处理付出同步开销，因为每个节点都需要关于网络全局状态的信息，当物理网络的规模增加时，同步开销也越发的庞大。对于网安试验，虚拟网络请求的数量不会过多，而且物理网络规模十分庞大，因此采用集中式算法。

(3) 简单式算法或冗余式算法

在虚拟网络请求被映射成功后，某个物理节点的失效可能影响到它所承载的所有虚拟节点，因此，在部署故障敏感的虚拟网络时，可以设置备份物理资源，在部署所使用的主要节点失效的情况下，这些备用节点可以用作备份。实现这一点，虚拟网络请求映射的结果就是冗余的，对于故障是有弹性能力的，方法就是冗余式算法^{[17][18]}。否则，就是简洁式算法^[19]。

对于网安试验而言，这并不是一个对故障敏感的虚拟网络，而且往往为了模拟 DDOS 攻击等，反而会主动造成网络故障来进行模拟实验，因此采用简单式算法。

1.2.3 虚拟网络映射算法的策略

虚拟网络映射算法的策略其实就是解决虚拟网络映射问题的思路，从如何看待问题的角度，可以把虚拟网络映射算法分为两个方面，一是将虚拟网络映射问题分为两个过程的求解，即如何完成虚拟节点的映射过程，以及如何完成虚拟链路的映射过程，本文称这种方案为映射方案。另一反面，虚拟网络映射问题从图论的角度看，也相当于把一个大的拓扑图划分为多个子图，因此，也可以转化为图划分问题，从图划分的角度去解决，本文称这种方案为图划分方案。

(1) 映射方案

如之前所说，因为映射方案将虚拟网络映射的映射问题分为两个过程，显然根据这两个过程的相关性，映射方案可以被划分为两种特点的方案。一是虚拟节点映射与虚拟链路映射两个过程相互独立，这就是一个典型的 Two-Stage（两阶段）映射方案，首先虚拟节点根据各种启发式的排序方案进行排序映射，虚拟节点映射完毕之后，因为任意一条虚拟链路的起点与终点都已经被固定，虚拟链路的映射就被简单化了，虽然仍然无法求得最优的解，但已经有许多成熟的解决方案。二就是虚拟节点映射与虚拟链路映射两个过程相互协调，一起完成，这类方案需要综合考量节点和链路，往往复杂性较高。

对于两阶段映射方案的研究已有很多年的历史,早在 2006 年,针对虚拟网络映射问题和图论问题,Zhu^[20]等人首次建立了转化关系,并提出了基于贪心策略的启发式算法,这种思路被沿用至今。在 2007 年,为了解决链路的分割和迁移问题,Yu^[21]等人在贪心策略的基础上建立了约束,利用多商品流算法进行解决。以上两种算法的不足在于这两个算法要求虚拟网络请求是提前被设计好的,无法针对实时到来的虚拟网络请求进行处理,同时这两个算法对于虚拟节点映射与虚拟链路映射的处理完全是分离的,没有任何相互考量,实用效果比较一般。后来出现从数学角度解决问题的思路,在 2012 年,Chowdhury^[22]再一次提出用数学规划来完成虚拟网络映射问题,提出了元节点和元链路的概念,并利用其进行映射考量。在 2015 年,Ding^[23]等人提出了利用图论中间性的方法,采取新的节点选择方案来联合节点映射和链路映射。2018 年,刘新波^[24]等人将物理学的概念引入映射问题中,提出了拓扑的势能概念,并利用其进行节点的映射。这几个算法较之之前的算法,在节点映射阶段,就一定程度上考量了链路映射,使虚拟链路映射成功率更高,并引入了数学的思维,映射效果较好。

对于一阶段方案,在 2012 年,H. Yu^[25]等人提出一个被称为虚拟基础设施映射算法(VIMA)的算法,这个算法综合考虑了节点映射和链路映射,计算了节点与链路协调额外的映射开销,链路和节点映射阶段之间的协调增加了算法的效率,效果较好,但实际运行上,还是以节点优先运行的。

映射方案在映射时,往往要做遍历虚拟网络和物理网络的乘积次的循环,保证求解的每一步全局最优,这就导致算法的效率往往比较一般。同时对于映射方案,解决节点的映射比较容易,但解决链路的映射,仍然是一个 NP 难的问题,面临链路数较多时,往往效果较差。

(2) 图划分方案

将虚拟网络映射问题看做图划分问题,也已经有多年的研究历史了,在 2011 年,黄敏桓,张尧学^[26]等人提出了基于模拟退火算法进行子网分隔,产生虚拟网络映射结果,以求解近似最优解。2012 年,Leivadreas 等人提出了一个基于局部迭代搜索的划分方法^[16],划分虚拟网络请求,使之适应跨多个物理网络的分布式环境。

2014 年,刘百川^[27]等人提出了一种基于图划分的节点映射策略。该方案主要通过 matis 工具处理,首先将拓扑图转化为 matis 的格式,然后进行分割,节点的映射直接按照分割结果来进行。2015 年,彭利民^[28]提出了一种新的图的分割方法,通过把虚拟网络分为多个子结构,建立新的物理网络和虚拟网络的匹配模型,解决虚拟网络映射问题。

图划分的方法都将拟网络映射问题转化为图划分问题,虽然简化了问题的模

型，划分速度比较好，比较适应大规模环境，但因为约束条件较难反映，因此总体的划分结果比较一般。

1.2.4 虚拟网络映射算法的评价

在这一节，本文主要给出当前对于虚拟网络映射算法的评价指标。对于一个虚拟网络映射算法，清晰的评价指标是十分必要的，只有在有一个清晰地评价指标的前提下，才可以比较不同的虚拟网络映射算法之间的优劣。但与虚拟网络映射问题一样，因为面临的具体场景不同，虚拟网络映射算法的评价指标也有所不同，通过分析大量有关虚拟网络映射算法的论文，下面本文给出一些常见的评价指标。

(1) 服务质量指标

服务质量指标的含义为在进行虚拟网络映射时，映射结果对于服务质量的影响。例如，当一个虚拟网络要承载一些实时应用，如网络游戏时，这个虚拟网络就对延迟和稳定性要求比较高，使用户尽量感受不到网络的波动；如果承载的是一些网盘服务，就对传输速度要求比较高。Adil Razzaq^[29]等人就提出一种叫做最近节点映射的算法，主要思想就是在虚拟节点映射时，选取在物理网络上比较相近的物理节点，从而减小虚拟路径在物理路径上的长度，对于该算法，服务质量指标就是其比较着重考量的因素。

除了上述典型的延迟，稳定性等服务质量指标之外，还有一些其他的服务质量指标，例如在映射结果中，相邻虚拟节点在物理链路之中的平均距离，这个指标可能会影响延迟。此外，还有物理节点的压力程度，这影响着负载的管理。

(2) 资源开销指标

资源开销指标比较容易理解，就是映射一个虚拟网络请求所花费的资源开销。具体而言，包括如下一些指标：

- a) 映射收益，即映射一个虚拟网络能够产生多少收益，具体而言就是整个虚拟网络所需求的资源。
- b) 映射开销，即映射一个虚拟网络消耗了多少物理资源，具体而言包含映射这个虚拟网络消耗了多少物理节点的资源和物理链路的资源。
- c) 收益开销比，即映射收益比上映射开销。
- d) 接受率，即成功映射了多少个虚拟网络。

对于大部分虚拟网络映射算法而言，资源开销指标都是其关注的主要因素，因为这一指标最直接的反映了对于物理资源的利用率，对现实世界而言，这一指标直接反映了经济收益，前面给出的大部分文献都在关注资源开销这一指标，这里不再具体给出。

对于网安试验而言，因为映射的网络规模比较巨大，资源消耗较高，面向网安试验的大规模虚拟网络映射算法主要考量的就是对资源的消耗程度。因此，资源开销指标同样也是本文主要关心的指标，其中收益开销比和接受率反映了资源利用率，是本文对算法评价的重要因素，并将在下一章给出具体映射算法评价函数。

（3） 弹性指标

类似前文提到的冗余式算法，弹性指标就是指映射结果的弹性程度，这项指标反映了一个虚拟网络算法的映射结果对于故障的敏感程度。例如，一个承载工业生产程序的虚拟网络就对弹性指标要求较高，因为在发生故障后，如果虚拟网络不能快速回复，很可能影响到生产的进行，进而影响到经济效益。具体而言，弹性指标包括虚拟节点的备份数量，虚拟路径的冗余程度以及从一个阻塞中恢复所需要的时间等等。一些具体关注弹性指标的虚拟网络映射算法可以参考前面的冗余式算法，冗余式算法的目标就是为虚拟网络请求的映射结果提供弹性。

（4） 其他指标

其他指标主要包含一些特殊场景下的指标，不是之前的普适性指标，本文在这里给出两个例子。其一是算法的运行时间，假设当算法工作在一个虚拟网络请求频繁到来，且生存周期极短，因此频繁退出的环境下时，采用这项指标评判虚拟网络映射算法的好坏就有很重大意义。其二是节点间的消息数量，对于一个分布式算法而言，因为在映射的过程中，其需要不断的同步信息，获取物理网络全局状态，因此消息开销是一个很大的影响因素，节点间的消息数量就是一个比较重要的指标。总体而言，根据要解决的具体问题所提出的一些具体指标都可以划入此范围内。

1.2.5 虚拟网络映射算法的进展

在了解虚拟网络映射算法的分类，策略以及评价方式之后，简单阐述一下虚拟网络近期进展状况。

在 2016 年，Xu^[30]等人提出了一种考量时序的虚拟网络映射算法，该算法面向跨多个自治域的云端，提出了预测和分配周期性资源需求的方法。2017 年，Haeri^[31]等人提出了一种基于马尔科夫决策过程的映射算法，利用蒙特卡洛树进行搜索，对于跨多个自治域的云端服务提供商，可以提高其收益。2018 年，Rost^[32]提出了一个新的线性规划公式，并将其引入跨多个自治域的云上虚拟网络映射，提高映射收益。

纵观近三年的虚拟网络映射算法研究，其研究热点都在于跨多个自治域的云端映射，想办法提高云端服务提供商的收益，基本上都属于分布式算法，并提供了一定的冗余能力，网络规模也比较大，这符合当下云计算时代的特征。但本文所着

重关注的是面向网安试验的大规模虚拟网络映射问题，环境是集中的，不会跨自治域的，因此这些算法对本文有一定的借鉴意义，适当参考。在未来，随着互联网的不断扩大与发展，显然虚拟网络映射算法会朝向云计算与大规模两个大方向继续进展下去。

1.3 主要研究内容

本文在了解了网络虚拟化技术并分析了虚拟网络映射问题的本质之后，结合网安试验的特殊性，在大规模网安试验中，虚拟网络的规模不再是简单的几个或几十个节点，而是几千个节点，甚至上万，上十万级别的节点规模，本文给出这种新问题的定义与其目标函数，并结合以前的算法的解决思路，提出一种新算法，通过大量的模拟实验验证所提出的新算法，在新的目标环境下表现良好，在收益开销比和接受率方面均有很大进步。

具体而言，本文的主要内容如下：

(1) 分析传统的虚拟网络映射问题的特点，并结合网安试验的现状，提出相应的大规模虚拟网络映射问题，并给出其数学定义和目标函数。

(2) 分析已有的虚拟网络映射算法发现，映射方案的映射效果较好，但速度慢，面对链路数过多时失败率高，图划分思路的速度快，但映射质量一般，本文将结合这两种思路，给出一个先利用图划分进行前处理，在进行节点映射的虚拟网络映射算法。

(3) 前处理过程基于多层 K 划分算法，这是一种图划分的解决方案，通过制定的粗化处理， K 路划分和细化处理，可以缩小虚拟网络的规模，同时因为节点的第一步聚合，减少了虚拟网络请求的链路的数量，因为映射过程对边的规模十分敏感，这相当于加快了映射过程的处理速度和处理质量。

(4) 分析节点映射过程面临的问题，结合网络拓扑结构的特点，综合考量节点和链路，给出新的节点权重计算方法，并提出了节点契合度计算公式，根据节点契合度计算公式可以更好的进行节点映射的选择，根据这两个公式，给出一个新的节点映射过程。

(5) 实现一个虚拟网络映射系统，利用虚拟网络映射系统，把本文提出的大规模虚拟网络映射算法和目前已有的经典算法作对比，验证本文提出的算法在收益开销比和接受率方面的效果。

1.4 论文的组织结构

本文的组织结构如下：

第一章是绪论部分，主要介绍了面向网安试验的大规模虚拟网络映射问题的背景及研究意义，说明了这项研究的重要性和合理性，并对所涉接到的相关工作做一个总结，分析已有研究的优缺点，从中吸取和借鉴，最后给出文章的主要内容和结构安排

第二章介绍了大规模虚拟网络映射问题，并给出了这个问题的问题模型和定义，同时给出映射结果评价函数，作为评价解决方案好坏的指标。

第三章介绍了基于多层 K 划分的前处理过程，详细描绘了前处理过程的主要阶段，它包含粗化处理， K 路划分和细化处理三个阶段，并针对每一个阶段，给出相应的算法和计算公式。

第四章介绍了基于节点契合度的映射过程，首先给出了算法的整体思路，然后给出了节点映射过程的两个重要公式，节点排序公式和节点契合度计算公式，然后说明如何根据公式进行节点映射，最后修改经典的链路映射过程，给出了新的链路映射过程。

第五章则介绍了实验部分，首先说明了实验系统的是如何搭建的，然后根据虚拟网络的规模，给出从十级别到十万级别的对比实验结果，分析实验结果，验证本算法的在接受率和收益开销比方面的优越性。

第 2 章 大规模虚拟网络映射问题定义

分析当前的大部分虚拟网络映射算法，可以发现目前的算法所应对的问题往往都是较小规模的，集中在几个节点，几十个节点这个范围，这些算法所提到的大规模，也仅仅达到了几百个节点这种规模。与此同时，著名的美国网络靶场^[33]（*National Cyber Range, NCR*）已经达到了万级虚拟网络的实验规模，依照要求可以迅速的进行靶场的环境恢复，重新生成节点进行网安试验，而且这些操作在小时级别内就可以全部完成，效率非常高。

根据上述数据，显然当下虚拟网络映射算法的目标规模，与网安试验动辄几百个虚拟节点，一个大规模试验往往达到了上万虚拟节点的规模之间存在着较大的差距。

在存在这种巨大差距的情况下，原有的虚拟网络映射问题的定义在一些方面已经不再适用，比如大部分的虚拟网络映射算法中，物理网络拓扑是大于虚拟网络拓扑的，因此都聚焦于虚拟节点和物理节点的一一映射，这在当前的环境下就不再适用，在当前环境下，往往会发生虚拟网络拓扑大于物理拓扑的情况，一个物理节点往往需要承载数个虚拟节点。因此，本文将针对网安试验的规模特点，提出相应的大规模虚拟网络映射问题，并给出相应的描述和定义。

在本章中，2.1 节将分析虚拟网络映射问题的特点，给出相应的基本网络模型；2.2 节将对大规模虚拟网络映射问题给出描述和约束条件，并同时给出相应数学的形式化描述；2.3 节则结合网安试验的关注点，针对大规模虚拟网络映射问题的特点提出相应的映射结果评价函数，用于评价算法的优劣；2.4 节将对本章节的内容作出总结。

2.1 基本网络模型

对于基本的网络模型而言，传统的虚拟网络映射问题和大规模虚拟网络映射问题并无区别，因为实际上要解决的问题都是将一个拓扑图放入另一个拓扑图当中，因此涉及到的模型就是虚拟网络拓扑图和物理网络拓扑图，所以大规模虚拟网络映射问题的基本网络模型可以沿用原有的网络模型，不需要作出什么修改。

首先看一个简单的虚拟网络映射的图示，如图 2-1，在图中，两个虚拟网络请求需要被映射到一个基础物理网络之中，其中涉及到两个过程，虚拟网络节点的查找过程，虚拟网络链路的查找过程。在这里主要涉及到了两个实体，虚拟网络拓扑和物理网络拓扑，同时还有它们的一些基本信息，即相应的节点约束条件和链路约

束条件。

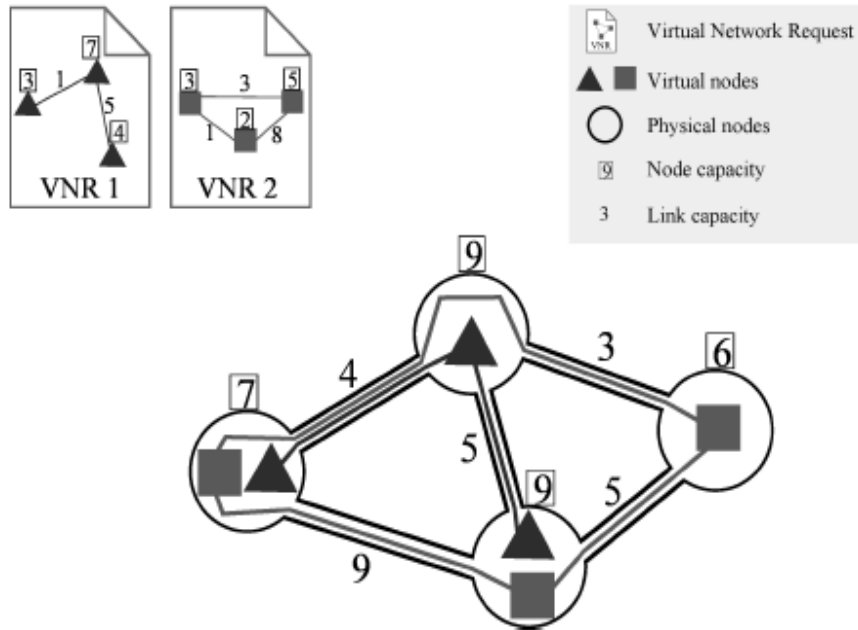


图 2-1 虚拟网络映射简图^[15]

由此，对于大规模虚拟网络映射问题而言，所涉及到的网络模型，包含两个模型，即物理网络模型和虚拟网络模型，物理网络模型描绘了物理网络拓扑及其约束条件，虚拟网络模型则，描绘了虚拟网络拓扑及其约束条件，具体定义如下：

物理网络模型

物理网络模型（*Substrate Network Model*）是一个无向拓扑图，链路和节点带有权重，本文用 G_s 指代， $G_s = (V_s, E_s, C_s, W_s)$ ，其中 V_s 代表物理节点的集合，对于任意一个节点 v_s ，都具有一个约束条件，节点的处理能力 $C(v_s)$ ，代表这个节点所能提供的最大处理能力，它所构成的集合就是 C_s 。 E_s 代表物理链路的集合，对于任意一条链路 e_s ，都具有一个带宽约束 $W(e_s)$ ，代表这条链路所能提供的最大带宽，带宽约束所构成的集合就是 W_s 。

虚拟网络模型

虚拟网络模型（*Virtual Network Model*）同样是一个拓扑图，本文用 G_v 指代， $G_v = (V_v, E_v, C_v, W_v)$ ，也是带权重的无向图，其中 V_v 代表虚拟请求中节点的集合，对于任意一个节点 v_v ，都具有一个约束条件，节点的处理能力 $C(v_v)$ ，代表这个虚拟节点所需要的处理能力，它所构成的集合就是 C_v 。 E_v 代表虚拟请求中链路的集合，对于任意一条链路 e_v ，都具有一个带宽约束 $W(e_v)$ ，代表这条链路所需要的最大带宽，它所构成的集合就是 W_v 。

然后，在这里给出本文常用的一些函数的说明。 $\min()$ 函数的含义为求一个集

合的最小值， $\max()$ 函数的含义为求一个集合的最大值， $\text{sum}()$ 函数的含义为对一个集合求和， $\text{avg}()$ 函数的含义为求一个集合的平均值。

2.2 映射问题及数学定义

大规模虚拟网络映射问题简单理解，就是一个虚拟拓扑图放入物理拓扑图的过程，但又和传统映射有所不同，如图 2-2，在大规模虚拟网络映射问题中，左边的虚拟网络往往是大于物理网络的，因此，根据鸽巢原理，物理节点与虚拟节点是一个一对多的关系。这就是大规模虚拟网络映射问题的核心所在，普通的虚拟网络映射如图 2-1，物理拓扑的规模是大于虚拟拓扑的规模的，因此，在这种情况下，物理节点与虚拟节点关系是一一对应的。但在大规模虚拟网络映射中，虚拟网络请求会有大于物理网络规模的情况，甚至虚拟网络请求的规模要比物理网络规模大一甚至三个数量级。

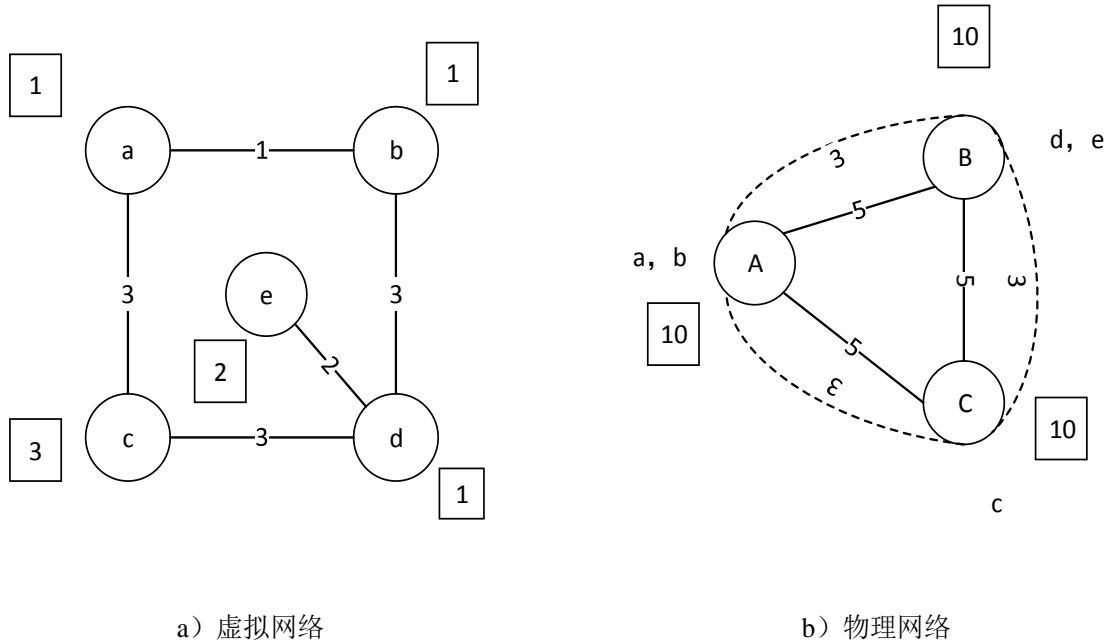


图 2-2 大规模映射简图

在这种情况下，大规模虚拟网络映射问题的基本形式化定义如下：

定义 2-1：基础物理资源 (*Substrate Resource, SR*)，基础物理资源定义为一个三元数据集 $SR = \{SN, SC, SW\}$ ，其中 SN 就是上文提到的物理网络模型，描述了基础物理网络的属性，包含基础物理网络的拓扑形式，各个物理节点的处理能力以及各条物理链路的最大通信带框限制， SN 在映射中视为不变的量，除非发生了基础物理网络的改变。 SC 是所有物理节点当前的处理能力构成的集合，其初始状态为

SN 中的 C_s ，随着虚拟网络的不断部署，在不断地改变。 SW 是所有物理链路当前剩余的通信带宽能力，其初始状态为 SN 中的 W_s ，随着虚拟网络的不断部署，也在不断地改变。

定义 2-2: 虚拟网络请求 (*Virtual Network Request, VNR*)，虚拟网络请求定义为一个三元数据集合，包含虚拟网络拓扑信息，各个虚拟节点的处理能力要求以及各条虚拟链路的通信带宽要求。 $VNR = \{VN, T_{start}, T_{end}\}$ ，其中， VN 为上文提到的虚拟网络模型， T_{start} 和 T_{end} 分别为该虚拟网络所要求的进入时间和退出时间，虚拟网络需要在 T_{start} 前被映射，在 T_{end} 后被移除， T_{start} 至 T_{end} 这段时间就是虚拟网络的生存周期。

定义 2-3: 映射关系 (*Mapping Relationship, MR*)，映射关系定义为一个二元组，它保存着虚拟网络请求 VNR 到基础资源 SR 的映射关系，有 $MR = \{f, g\}$ ，其中 f 保存着节点之间的对应关系， $f: V_v \rightarrow V_s$ ， g 保存着链路之间的对应关系， $g: E_v \rightarrow E_s$ 。

定义 2-4: 映射约束条件 (*Mapping Limit Condition, MLC*)，映射约束条件定义为基于物理网络资源为了容纳嵌入的虚拟网络请求，需要满足的约束条件，它是一个二元组 $MLC = \{VLC, ELC\}$ ，其中 VLC 的含义为了基础物理网络为了容纳虚拟网络请求，节点的约束条件， ELC 的含义为了基础物理网络为了容纳虚拟网络请求，链路的约束条件。

根据上述定义，一个大规模虚拟网络映射问题的具体描述如下：

在给定一个基础物理资源 SR 的情况下，每当来临一个虚拟网络请求 VNR 时，在满足映射约束条件 MLC 的前提下，寻求一个映射关系 MR ，使得任意一个虚拟节点被映射至物理节点，任意一条虚拟链路被映射至物理链路。

即计算 $f: V_v \rightarrow V_s$ 和 $g: E_v \rightarrow E_s$ ，要求计算过程始终满足 VLC, ELC ，将 VLC 和 ELC 具体展开成公式 (2-1)

$$\begin{cases} SC(f(v_v)) > C(v_v), \forall v_v \in V_v \\ \min(SW(g(e_v))) > W(e_v), \forall e_v \in E_v \end{cases} \quad (2-1)$$

如上述公式，第一项为节点约束条件，即将任意一个节点映射至某一物理节点时，要求该物理节点的当前处理能力剩余值大于虚拟节点所需求的处理能力。第二项为链路约束条件，即将任意一条虚拟链路映射至物理链路时，因为虚拟链路可能会通过多条物理链路，要求这多条物理链路的最小带宽剩余值大于虚拟链路所需求的通信带宽。

为了便于理解，公式 (2-1) 也可以表达成如下的公式 (2-2)

$$\begin{cases} \text{sum}(C(v_v)) > C(v_s), \forall v_v \in v_s, \forall v_s \in V_s \\ \text{sum}(W(e_v)) > W(e_s), \forall e_v \in e_s, \forall e_s \in E_s \end{cases} \quad (2-2)$$

其具体含义为，在任意一个物理节点上的全部虚拟节点的处理能力之和小于物理节点的处理能力，通过任意一条物理链路的虚拟链路的带宽要求之和，小于物理链路的通信带宽上限。公式（2-2）相当于公式（2-1）的一个变形，因为只要在映射的过程中保证选择的物理节点的当前处理能力剩余值大于当前虚拟节点所需求的处理能力和选择的物理链路的最小带宽剩余值大于虚拟链路所需求的通信带宽，就一定能保证任意一个物理节点和任意一条物理链路资源不超限，反之亦然。

2.3 映射结果评价函数

在得到大规模虚拟网络映射问题的形式化表述之后，还有一个问题就是如何来判断映射结果的好坏，如果没有清晰的评价体系，也无法比对算法之间的优劣，因此在这一小节，本文给出用于对比不同算法映射结果的评价函数。

如前文提到的，对于一个虚拟网络映射算法而言，可以从很多方面评价它的好坏，如服务质量，资源开销等等。但考虑到网安试验的特殊性，小则几百个节点，大则成千上万的节点规模，同时网安试验往往会进行一些对于网络的破坏性实验如 DDOS 攻击等，本文将评价算法好坏的重点聚焦于资源开销指标。

在给出在这里给出评价函数公式（2-3），此函数就是前文提到的虚拟网络映射的收益开销比（*Ratio of Cost to Revenue*），函数的分子代表本次映射的收益，函数的分母，代表本次映射的开销，函数的值越大，代表映射效果越好。接受率的计算十分简单，就是成功映射的虚拟网络请求比上全部的虚拟网络请求，这里不再赘述。

目标函数：

$$\begin{aligned} \frac{R}{C} &= \frac{\sum R(G_v)}{\sum \text{Cost}(G_v)} \\ &= \frac{\sum_{e_v \in E_v} W(e_v) + \alpha \cdot \sum_{v_v \in V_v} C(v_v)}{\sum_{e_v \in E_v} (W(e_v) \cdot \text{length}(M^E(e_v))) + \beta \cdot \sum_{v_v \in V_v} C(v_v)} \\ &= \frac{\sum_{e_v \in E_v} W(e_v) + \alpha \cdot \text{avg}(S_N(V_s)) \cdot |V_s|}{\text{avg}(S_L(E_s)) \cdot |E_s| + \beta \cdot \text{avg}(S_N(V_s)) \cdot |V_s|} \end{aligned} \quad (2-3)$$

公式（2-3）中：

（1）目标函数上值 $\sum R(G_v)$ 为映射一个虚拟网络的整体收益，具体而言，相当于虚拟网络的节点处理能力和边带宽求和。下值 $\sum \text{Cost}(G_v)$ 为映射这个虚拟网络带来的开销，具体而言，相当于物理网络为了映射该虚拟网络消耗的节点处理能力和

边带宽求和。

(2) $\sum_{e_v \in E_v} W(e_v)$ 由虚拟网络拓扑而定，就是所有虚拟链路的权重求和，为固定值，不需讨论。

(3) $\sum_{v_v \in V_v} C(v_v) = avg(S_N(V_s)) \cdot |V_s|$ 同样也是固定值，相当于所有虚拟节点权重求和。 $|V_s|$ 为已经被映射的物理节点的数量， $avg(S_N(V_s))$ 为已经被映射的物理节点的平均压力，为了尽可能提高物理节点资源利用效率，只需要让该值尽可能的高。

(4) $\sum_{e_v \in E_v} (W(e_v) \cdot length(M^E(e_v)))$ 含义为全部虚拟链路的权重乘以虚拟链路在物理链路中的长度，代表物理网络为了映射虚拟链路所产生的开销，等于 $avg(S_L(E_s)) \cdot |E_s|$ ，其中 $avg(S_L(E_s))$ 为物理网络中所有链路的平均压力， $|E_s|$ 为物理网络中的链路数量。

根据评价函数可以分析出，在给出具体虚拟网络请求之后，函数的 R 值即分子就是固定的了，为了让整个收益开销比尽可能大，应该使 C 值即分母越小越好，所以应该尽可能的让 $avg(S_L(E_s))$ 即物理链路的平均压力尽可能的小，或者说让每个对应虚拟链路 e_v 的物理路径长度 $length(M^E(e_v))$ 尽可能短，由此可以使得整体的收益开销比变大，显然，在每个对应虚拟链路的物理路径长度尽可能短的条件下，资源的消耗也会尽可能的小，符合预期。

在得到映射结果评价函数之后，对于算法的优化思路也确定的下来，显然映射一个虚拟网络请求的收益是无法更改的，因为它由虚拟网络拓扑本身决定，想要获得更好的收益开销比，要办的办法就是降低开销，同时，因为节点的映射是比较固定，要想降低开销，就只能从链路的映射下手。

2.4 本章小结

在本章中，主要给出了研究大规模虚拟网络映射问题需要了解的一些相关基础概念，并对比传统的虚拟网络映射问题，结合网安试验的特殊性，分析为什么要这么处理。

首先是给出了基本网络模型，在网络模型方面，传统虚拟网络映射问题和大规模虚拟网络映射问题并没有什么本质区别，基本上可以直接沿用已有模型的定义和概念。

然后分析出传统虚拟网络映射问题和大规模虚拟网络映射问题的主要差别在于网络规模，由此说明，在大规模虚拟网络映射问题中的一个重要思路就是节点的可重复映射，即物理节点与虚拟节点是一个一对多的关系，这是与传统虚拟网络映射问题不同的，在此基础上，提出了大规模虚拟网络映射问题的数学定义。

最后，分析网安试验的特点，结合第一章提出的虚拟网络映射算法的评价，给出了针对大规模虚拟网络映射算法的评价函数，并通过分析评价函数，找出优化大规模虚拟网络映射算法的大致思路。

第3章 基于多层 K 划分的前处理过程

分析已有的虚拟网络映射算法可以发现，图划分方法解决虚拟网络映射映射问题的优点在于可以简单快速的产生一个映射结果，但是纵观已有一些的图划分方案，如 2015 年，郭艳来^[34]等人提出分割方法，可以在数秒内产生万级别的虚拟网络的划分，但对于如何选择划分的数量，往往基本上就选择物理网络的节点数量，这十分的粗糙，同时，对于划分的过程中，如何控制各个划分区域的负载，也处理的较差，导致最后的映射效果虽然可用，但十分粗糙，收益开销比也很低。

然后考虑已有的映射方案，映射方案的主要问题就在于解决了节点的映射之后，链路的映射仍然是一个 NP 难的问题，当虚拟网络请求的链路数过多时，链路映射失败率较高。

因此，本文结合图划分方案和映射方案两种思路，提出一种新算法，先图划分，再映射。首先利用图划分方法给出一个简单的切分，在保证映射效果的同时，是减小链路数量，提高映射过程的成功率和质量，同时避免复杂的约束控制。然后针对简单的切分，提出新的映射过程，体现约束，追求更优化的收益开销比。

本章主要针对前处理过程作出详细的说明。在 3.1 节说明前处理过程的大概过程，为何可以使用多层 K 划分来进行前处理，以及如何保证映射结果较好。3.2 节到 3.4 节详细的说明前处理过程的三个流程粗化处理，K 路划分和细化处理的处理方式和控制条件。3.5 节提出一个带有前处理过程的经典映射算法。最后，对本章的主要内容进行总结。

3.1 前处理过程概述

首先对前处理过程作出整体上的描述。整个前处理过程基于图划分方案的经典解决算法，多层 K 路划分方案^[35-36]。在进行节点划分时，拓扑图的规模往往是很大的，因此直接将拓扑图划分为 K 个部分是十分困难的事情，各个部分之间的平衡也很难保证。因此想到，先将大拓扑图的规模减小，处理一个较小的拓扑，对其进行划分，然后逐步的还原这个拓扑图，修改划分方案，迭代产生这个大图划分结果。这就是多层 K 路划分思路的出发点。

多层 K 路划分的目标是将拓扑图划分为 K 各部分，同时使得各部分之间关联性尽可能的小，或者说是划分出的部分，之间的链路尽可能的小。这个目标和映射结果评价函数所追求的不谋而合，这也是能够使用这种方法进行虚拟网络映射的前提，在这里对这一点作出说明。多层 K 路划分的目标是各部分之间的总通信量

最小化,换句话说,就是希望各个划分区域之间的内部带宽大,对外的通信带宽小。而如映射结果评价函数提到的,大规模虚拟网络映射算法中,虚拟节点的映射代价是不变的,大规模虚拟网络映射算法所追求的,就是虚拟链路的映射代价更低,这可以降低整体映射开销,多层 K 路划分总体上,使得大的虚拟链路被隐藏在节点内,对外通信的虚拟链路小,正是对更优映射结果的追求。因此可以用多层 K 路划分的办法处理大规模虚拟网络映射问题,更低的总通信量对应着更低的链路映射开销。

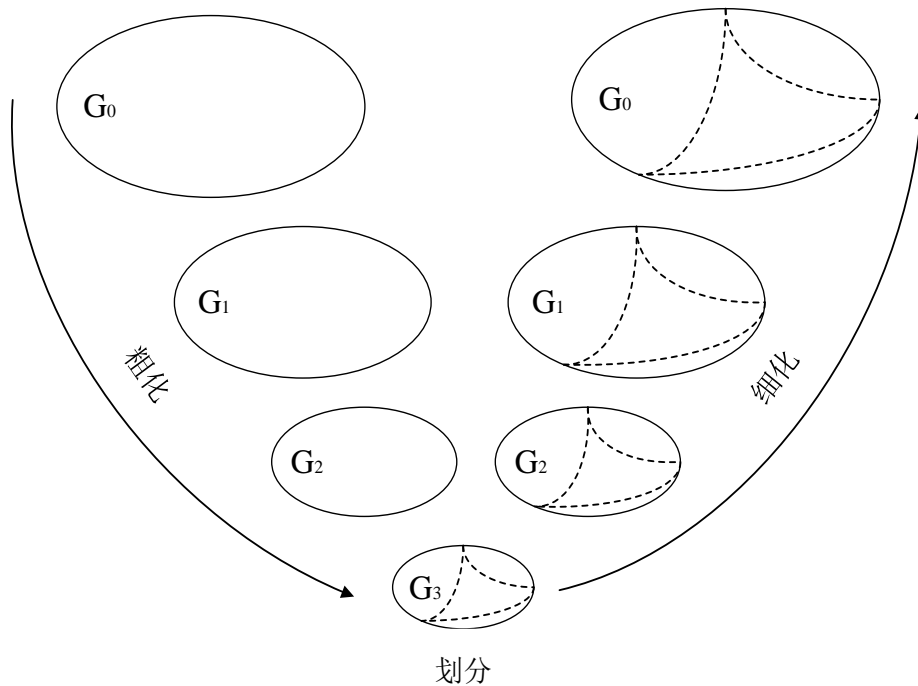


图 3-1 多层 K 路划分

然后说明前处理过程的具体流程,如之前所描述的,先将大拓扑图的规模减小,划分后在逐渐还原优化,如图示 3-1。基于多层 K 路划分的前处理过程就可以被分为三个阶段。首先是粗化阶段,粗化阶段的目标就是压缩虚拟网络拓扑图的大小,通过不断合并虚拟网络拓扑图中的顶点和边,将大图转化为一个足够小的图。然后是 K 路划分,这一步的目标就是把粗化阶段得到的虚拟网络拓扑图进行划分,分成 K 个部分,同时,使得各个部分之间的通信量最小化,或者说在划分时,使得切割的链路的权重之和最小,因为链路的权重代表着节点间的通信带宽。最后是细化阶段,因为 K 路划分后的结果仅仅是针对粗化后的虚拟网络拓扑图得,在逐渐还原成原图的过程中,调整节点和链路,会产生更好的划分结果,这就在细化阶段完成,根据第一步粗化压缩的信息和第二步 K 路划分的信息,还原优化虚拟网络拓扑图的划分方案。

同时多层 K 路划分算法的时间复杂度是 $O(|E|)$ 的，和拓扑图的链路数是线性的，因此可以它的处理速度也快于映射过程，至于映射过程的时间复杂度，将在说明映射过程的时候分析给出。

3.2 粗化处理

前处理过程的粗化阶段，总体来说，就是一个减小拓扑图规模的过程，对于拓扑图 $G_i = (V_i, E_i, C_i, W_i)$ 而言，经过一次粗化过程的压缩处理，得到新拓扑图 $G_{i+1} = (V_{i+1}, E_{i+1}, C_{i+1}, W_{i+1})$ ，要求新拓扑图的节点数小于原来拓扑图的节点数。

在一次处理过程中，对于节点来说， G_i 中的一些节点聚合到一起，形成新的节点，新的节点继承原来节点的属性，也就是新节点的处理能力等于所有形成它的节点的处理能力之和，或者从图的角度来讲，新节点的权重就是聚合产生它的节点的权重之和。对于链路来说，节点聚合到了一起，节点所连接的链路也聚合到了一起，新链路的通信带宽等于聚合产生它的链路的带宽之和，或者从图的角度讲，新链路的权重等于聚合产生它的链路的权重之和，但是有一些链路存在于聚合节点的内部，不对外体现出来，因此，链路的数量就相当于减少了，对于 G_{i+1} 来说，链路数 $|E_{i+1}|$ 小于 G_i 的链路数 $|E_i|$ 。

然后再分析拓扑图权重的变化。对于节点的权重而言，因为新节点的权重等于聚合产生它的节点的权重之和，因此 G_{i+1} 的节点的总权重等于 G_i 节点总权重，即 $\text{sum}(C_{i+1})$ 等于 $\text{sum}(C_i)$ 。对于链路的权重的而言，因为一些链路在聚合的过程中被隐藏在节点内部了，因此 $\text{sum}(W_{i+1})$ 小于 $\text{sum}(W_i)$ 。

所以，一次处理过程就是对于拓扑图 $G_i = (V_i, E_i, C_i, W_i)$ 而言，经过一次处理，得到拓扑图 $G_{i+1} = (V_{i+1}, E_{i+1}, C_{i+1}, W_{i+1})$ ，其中 $|V_{i+1}|$ 小于 $|V_i|$ ， $|E_{i+1}|$ 小于 $|E_i|$ ， $\text{sum}(C_{i+1})$ 等于 $\text{sum}(C_i)$ ， $\text{sum}(W_{i+1})$ 小于 $\text{sum}(W_i)$ 。整个粗化过程就是不断的迭代处理过程，通常当直到节点数到达跟 15K 或者一次粗化过程的压缩比例小于 0.8，结束粗化阶段。

因为找到一个具有最小边切割的划分方案是整个前处理过程的目标，因此直观考虑，权重较大的链路不被分割而隐藏在划分区域的内部可以使划分结果的边切割较小，同时针对映射结果评价函数，较大的边被隐藏起来，不需要被实际映射，整体映射开销也会变小。因此，本文采用粗化过程的经典算法重边匹配算法的改进版。

经典的重边匹配算法步骤如下：

- (1) 对于一个拓扑图，随机的访问一个尚未被访问节点 v_i 。
- (2) 选取与节点 v_i 相连，并同样没有被访问的节点 v_j ，要求节点 v_i 与节点 v_j

的边的权重是节点 v_i 的所有的边中最大的，标记 v_i 与 v_j 为已访问，且记录匹配信息。

- (3) 如果(2)步骤失败，节点 v_i 不存在没有被访问的邻居节点，同样标记 v_i 为已访问。
- (4) 重复(1)至(3)步骤。直至所有节点均被访问，根据记录的匹配信息，进行节点聚合。

重边匹配算法的优点如图 3-2 匹配对比图，和简单的随机匹配算法（节点间完全随机进行匹配）对比，重边匹配效果更好。经过以往的大量实验验证，重边匹配往往能取得较好的效果。

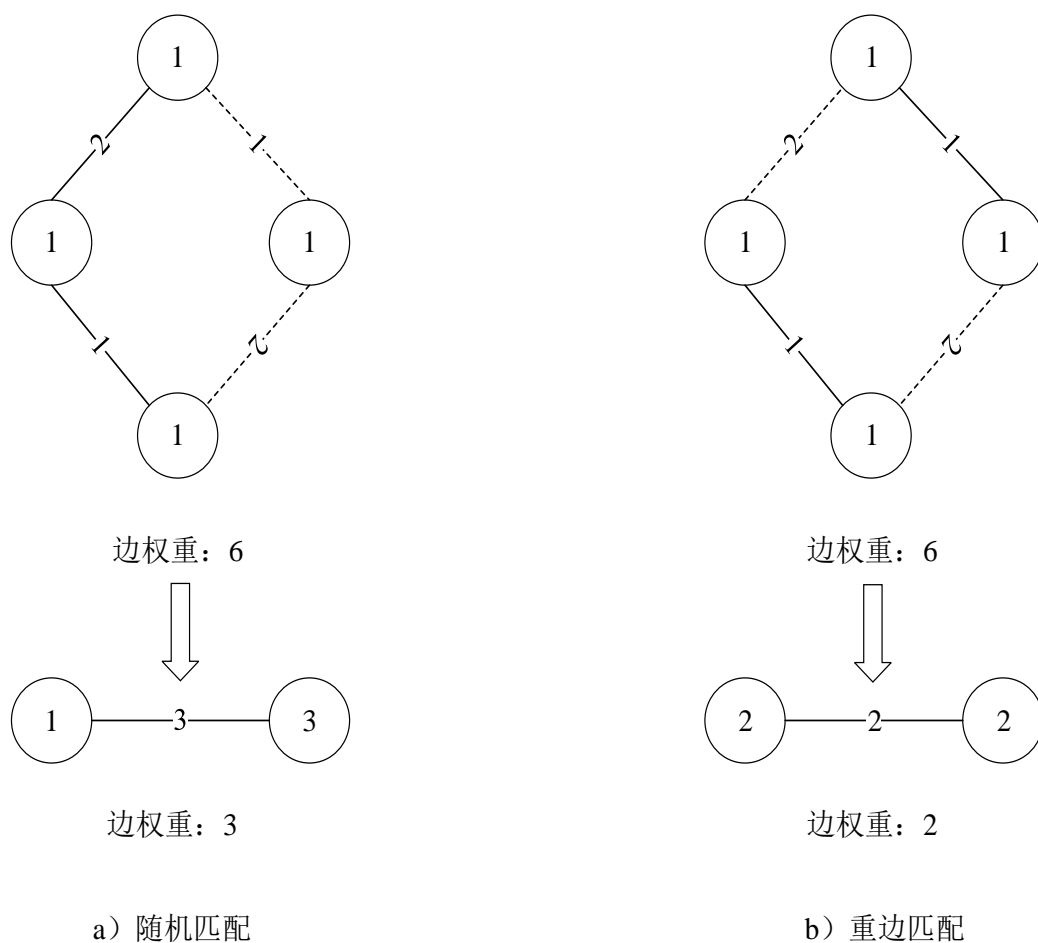


图 3-2 匹配对比

对于重边匹配算法的改进版本，主要在于改进它面临相同重边时的选择。因为一个虚拟网络拓扑图往往是比较大，因此在逐渐粗化的过程中，很可能面临一个节点进行节点匹配时，发现和多个邻居节点的所连接的链路权重是相同的，这时候，传统的重边匹配算法的办法就是随机选择一个节点进行匹配，但这种随机方法效

果比较一般, 因此, 提出一个改进方案, 当节点 v_i 与 v_j 和 v_k 节点之间的链路权重相等, 且均为最大时, 设 V_j 为 v_j 的邻接节点集合, V_k 为 v_k 的邻接节点集合, 计算 v_i 与 V_j 之间的直接相连的链路权重之和 w_{ij} , 同理, 计算 v_i 与 V_k 之间的直接相连的链路权重之和 w_{ik} , 选择 w_{ij} 和 w_{ik} 之间较大的节点, 与 v_i 进行匹配。

有研究表明, 采取改进的重边匹配算法, 可以获得更佳的粗化结果, 同时可以得到整体上更好的划分结果^[35-37]。同时, 因为该算法每一轮会遍历所有的链路, 其时间复杂度为 $O(|E|)$ 。

3.3 K 路划分

前处理过程的 K 路划分阶段, 主要完成对粗化阶段产生的粗图的 K 路划分, 对于粗化阶段产生的粗图 $G_i = (V_i, E_i, C_i, W_i)$, K 路划分将 G_i 分为 K 个部分, 并使得每个部分的权重大致为 $\text{sum}(C_i)/K$ 。基于对粗化阶段的信任, 只要粗化阶段进行的足够好, 粗图能够反映出原始图的特征, 只要对粗图的划分是平衡且边切割最小的, 那么这个划分结果大体上对原始图就是平衡且边切割最小的。同时, 这一部分的重点在于 K 的计算, 因为已有的图划分方案在 K 的设置方面都是十分粗糙的, 本文将在这一部分提出一个基于原始虚拟网络拓扑和物理网络拓扑的 K 的计算方法。

首先说明 K 路划分所采用的方法。有两种思路来产生这个划分, 第一种是持续的对图进行粗化处理, 直到拓扑图被粗化的只剩下 K 个顶点。如此粗化产生的结果就完成了初始划分过程。但是这种思路存在两个问题, 首先, 每一轮粗化, 都相当于失去原始拓扑图的一部分特征, 不断的对图进行粗化, 对原始拓扑图的特征损失很大。其次, 也是最主要的问题, 在不断粗化的过程中, 平衡度是很难控制的, 最后产生的 K 个节点的权重可能非常的不均衡。因此, 本文采用递归二分法^[36]来产生初始划分, 即对于粗化后的图, 首先将其二分为两部分, 然后二分出的部分再次不断二分, 已有实验表明, 该方法当图较大时, 能产生良好的初始划分效果, 具体可以采用贪心图增长划分算法, 其时间复杂度是 $O(E)$ 。

然后说明 K 的计算方法, 已有的图划分方案在 K 的设置通常为物理网络节点的数量, 然后把虚拟网络划分为 K 个部分, 粗暴的每个部分塞入一个物理节点当中, 因为物理节点的负载是不同的, 有的物理节点性能强或负载低, 能容纳更多的虚拟节点, 而有的物理节点性能较差或负载较高, 这就导致了划分出来的结果放入物理节点后可能会因为不满足约束要求而失败, 就算成功, 收益开销比方面也比较一般。因此, 需要先将 K 的规模放的比较大, 减少每一部分的权重, 相当于把大的虚拟网络拓扑图变为一个小的虚拟网络拓扑图, 然后用映射方案的方法去映射这个小的虚拟网络拓扑图, 相当于完成了整个映射。

K 的具体计算方法采用综合考虑节点负载和链路负载进行估计，对于一个虚拟网络拓扑 G_v 和物理网络拓扑 G_s ，具体计算公式为公式 (3-1)：

$$K = \varepsilon \cdot \max \left(\frac{C(G_v) \cdot |V_s|}{SC(G_s)}, \frac{W(G_v) \cdot |V_s|}{SW(G_s)} \right) \quad (3-1)$$

其含义为，对于物理网络拓扑 G_v 和虚拟网络拓扑 G_s ， $\max()$ 函数的第一项为，计算虚拟网络的节点权重之和 $C(G_v)$ ，将其除以物理网络的节点权重之和 $SC(G_s)$ ，再乘以虚拟网络节点数 $|V_s|$ 。 $\max()$ 函数的第二项为，计算虚拟网络的链路权重之和 $W(G_v)$ ，将其除以物理网络的链路权重之和 $SW(G_s)$ ，再乘以虚拟网络节点数 $|V_s|$ 。然后这两项取较大值，相当于计算平均需要多少的物理节点才能容纳这个虚拟网络请求，然后用得到的值乘以系数 ε 得到 K 的计算值，系数 ε 为放大系数，其含义为一条虚拟链路的映射通常需要多条物理链路才能完成，因此对于计算出的数值应当放大，这个系数在运行中是可变的，随着压力的增大，系数也不断放大。参考 Beck^[15]等人的方法， ε 过小会导致节点的过度聚合，网络规模过小，导致映射过程的灵活性降低，收益开销比较差，而 ε 过大，会导致节点聚合程度太低，链路规模减小效果较差，影响映射过程的成功率，因此本文设置 ε 的初始值 5，最大可增大到 10。

3.4 细化处理

前处理过程的细化处理的主要进行的就是在 K 路划分之后，根据粗化过程所保存的压缩匹配信息，不断的还原虚拟网络拓扑图，同时在还原的过程中，调整不同划分区域之间的一些点，使得划分方案更加的优化。

经过之前映射结果评价函数那一节的分析可得，优化所追求的目标为 $avg(S_L(E_p))$ 最小，对前处理过程而言，就相当于大的链路被隐藏在节点内部不被处理，而小的链路被分割，即为最小的边切割，因此这一步的主要操作就是，对于粗化后被 K 路划分的虚拟网络拓扑图，在根据信息一层一层还原为原始虚拟网络拓扑图的同时，尝试性的将一些点从当前划分区域迁移至别的划分区域，不断的计算最优的边切割。

参考以往的一些细化算法，贪心细化算法 (*Greedy Refinement, GR*) 从运行时间和处理效果的角度都是比较好的，因此本文使用贪心细化算法，来完成细化过程。

贪心细化算法的思路非常简单，即首先按照随机的方法去访问各个划分区域的边界节点，如果把这个边界节点移动到别的划分区域，能够减小边切割或者不增大边切割的情况下使不同划分区域更加均衡，那么就移动这个节点。这个算法是迭代工作的，每一轮迭代，访问全部的边界节点，然后一轮一轮进行迭代。

贪心细化算法工作流程如下, 对于一个拓扑图 $G_t = (V_t, E_t, C_t, W_t)$, 和一个划分结果 $P = \{P_i | i = 1, 2, \dots, K\}$, 设节点 v_t 是一个划分区域的边界节点, 且 v_t 在第 i 个划分区域中, 即 $v_t \in P_i$, v_t 和 P_i 内部的节点的链路的权重之和设为 $Cin_i(v_t)$, v_t 和任意一个划分区域如 P_j 的节点的链路权重之和为 $Cout_j(v_t)$, 如果要把 v_t 节点移动到别的划分区域 P_j , 需要满足如下任何一个条件即可:

$$(1) \quad sum(W(P_j)) + W(v_t) < W_{max} \quad \text{且} \quad sum(W(P_i)) - W(v_t) > W_{min} \quad ,$$

$$Cout_j(v_t) > Cin_i(v_t), \quad \text{且} \quad Cout_j(v_t) = \max(Cout_a(v_t)), a = 1, 2, \dots, K$$

$$(2) \quad Cout_j(v_t) = Cin_i(v_t), \quad sum(W(P_j)) - sum(W(P_i)) > W(v_t)$$

也就是说, 在满足平衡的约束的前提下, 如果移动节点 v_t 到别的划分区域, 能最大程度的减小边切割, 就移动这个节点。或者移动节点 v_t 到别的划分区域不增加边切割, 但可以减小划分区域间的不平衡也移动这个节点。在这里, W_{max} 与 W_{min} 为划分的约束条件, W_{max} 为一个划分区域所能接受的最大权重值, W_{min} 为一个划分区域所能接受的最小权重值。具体而言, W_{min} 的取值通常为 $0.9 \cdot sum(C_i)/K$, W_{max} 的取值通常为 $1.03 \cdot sum(C_i)/K$ 。在这里 W_{max} 与 W_{min} 是可以变化的, W_{max} 越大或者 W_{min} 越小, 产生划分结果的速度越快, 但是平衡程度也越差。

如之前所说, 贪心细化算法是一个迭代式算法, 在进行几次迭代之后就会收敛。对于大多数的虚拟网络拓扑图而言, 使用贪心细化算法可以在 8 到 12 个轮次之内细化处理完毕, 因此其时间复杂度小于 $O(|E|)$ 。

至此, 就完成了整个细化处理过程, 这里要说明一下为什么细化处理过程也要一轮轮进行处理, 因为在根据粗化阶段保存的压缩匹配信息还原虚拟网络拓扑图时, 即使对 G_i 而言, 已经是全局最优的划分方案, 在还原为 G_{i-1} 之后, 因为压缩的节点被展开, 在 G_i 中的一条链路, 在 G_{i-1} 可能是多条链路组成的, 因此在 G_{i-1} 会有更多的边, 所以可能会存在更优的划分方法。这时候, 就需要利用刚才给出的贪心细化算法, 再次细化, 直至贪心细化算法迭代结束, 无法继续优化, 然后将虚拟网络拓扑图继续还原, 继而再次贪心细化, 直至虚拟网络拓扑图被还原为最开始的规模, 无法继续还原, 这时候也就得到了初始虚拟网络拓扑图的前处理结果。

3.5 带有前处理过程的经典映射算法

在给出前处理过程后, 结合经典映射算法 Node-opt, 在这里给出一个带有本文前处理过程和经典映射算法的新算法 Pre-node-opt, Pre-node-opt 首先进行前处理过程, 然后用经典的映射方式进行映射, 如算法 3-1。

算法 3-1 Pre-node-opt Algorithm

Input: 基础物理资源 $SR = \{SN, SC, SW\}$, 其中 SN 为 $G_s = (V_s, E_s, C_s, W_s)$

虚拟网络拓扑 $G_v = (V_v, E_v, C_v, W_v)$

Output: 节点映射集合 $R(V)$, 链路映射集合 $R(E)$

```

1  for  $v_s$  in  $SR$  do
2       $PS[v_s] = SC(v_s) \cdot \sum_{e \in E(v_s)} W(e)$ 
3  end for
4  对  $G_v$  进行前处理过程
5  for  $v_v$  in  $G_v$  do
6       $PV[v_v] = SC(v_v) \cdot \sum_{e \in E(v_v)} W(e)$ 
7  end for
8  对  $PS$  数组和  $PV$  数组排序
9  for  $v_{vt}$  in  $PV[]$  do
10     for  $v_{st}$  in  $PS[]$  do
11         if (  $v_{st}$  满足  $v_{vt}$  约束 )
12              $R(V) : v_{vt} \rightarrow v_{st}$ 
13             更新  $PS$  数组
14             break
15         end if
16     end for
17     if (  $v_{vt}$  未被映射 )
18         拒绝  $G_v$  并退出
19     end if
20 end for
21 if ( 使用 Dijkstra 算法计算链路映射成功 )
22     链路映射结果保存在  $R(E)$  中
23 else
24     拒绝  $G_v$  并退出
25 end if
    
```

算法进行映射时, 前处理过程的时间复杂度为 $O(|E_v|)$, 前处理之后, 虚拟网络节点数变为 K , 排序部分的时间复杂度 $O(K^2)$, 匹配部分是一个双重循环, 时间

复杂度为 $O(K \times V_s)$ ，利用 Dijkstra 算法进行链路映射的时间复杂度为 $O(V_v \times V_s^2)$ ，因此，提出的 Pre-node-opt 算法的时间复杂度为 $O(V_v \times V_s^2)$ 。

3.6 本章小结

在本章中，主要给出了基于多层 K 划分的前处理过程的具体处理方式，并通过分析已有的一些图划分方案，说明其局限性，由此给出前处理过程的必要性和其改进点。

首先是对整个前处理过程作出描述，并分析前处理过程所采用的多层 K 划分方法，通过分析多层 K 划分方法的优化目标和大规模虚拟网络映射问题的优化目标，证明多层 K 划分方法可以用来解决大规模虚拟网络映射问题。

然后就对前处理过程的三个主要部分给出详细过程，并说明每一步的优化目标。第一步是粗化处理，将虚拟网络拓扑图压缩成小的拓扑图，给出了压缩方法；第二步是 K 路划分，给出了分割方法，并对比已有的图划分方案的问题，给出了精细的 K 计算公式；第三步是细化处理，将压缩并被分割之后的拓扑图还原为原图，并不断优化划分方案，给出了优化算法。

得到整个前处理过程后，给出了一个带有前处理过程的经典算法 Node-opt 的变种，Pre-node-opt 算法，并进行简单分析。

第 4 章 基于节点契合度的映射过程

通过在上一章给出的前处理过程，结合之前给出的先前处理，再映射的算法整体思路，整个算法的工作流程如图 4-1。首先读入虚拟网络，然后进行前处理过程，根据前处理的结果进行映射过程，如果映射成功则更新物理网络，如果映射失败，更新前处理过程的负载系数 ε ，重新前处理和映射，直至成功或者负载系数 ε 达到最大，失败退出。

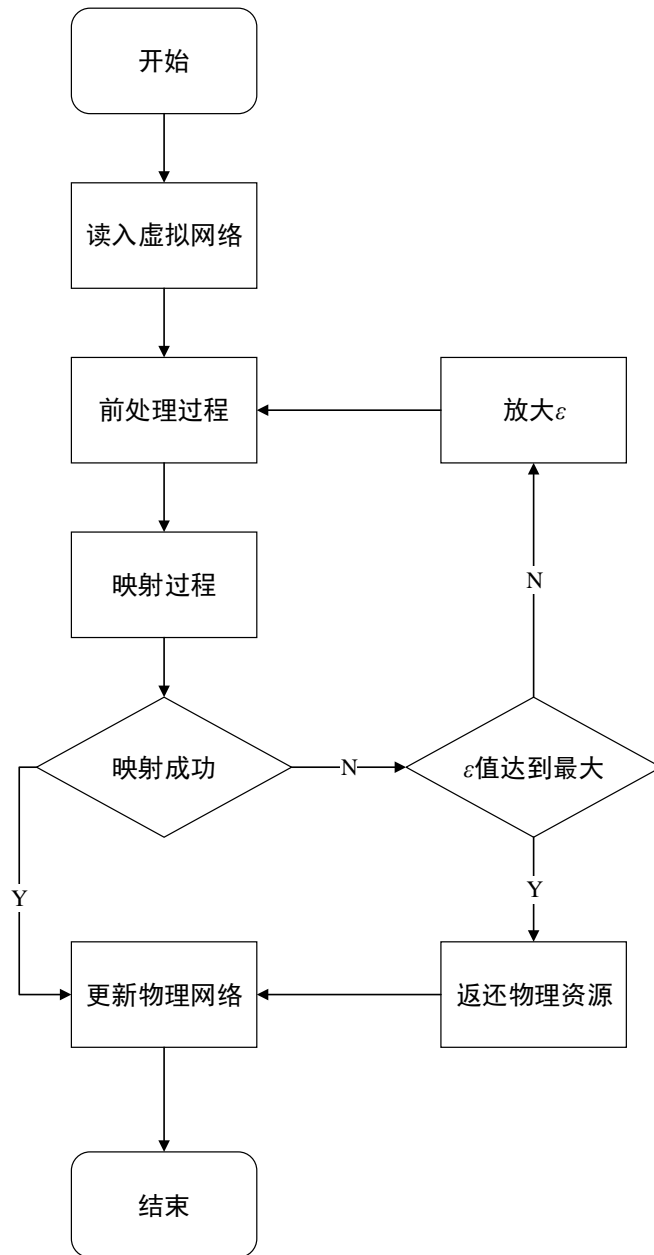


图 4-1 算法流程图

接下来在这一章，本文将给出基于节点契合度的映射过程，首先分析已有的映射算法，给出映射过程的整体优化思路，说明算法应该如何进行优化。然后对映射过程提出两个新的计算公式，即节点排序计算公式和节点集合度计算公式，分析给出新公式的优化点在哪里，并依照公式，给出算法的节点映射过程，同时针对链路映射，也给出一种优化的链路映射方式。最后，给出整个映射过程的算法，并从性能的角度对算法进行分析。

在本章中，在 4.1 节说明映射算法的整体思路。4.2 节给出了节点排序公式和节点契合度计算公式。4.3 节给出了链路映射的过程。4.4 节对算法进行了全面的分析。最后，4.5 节对本章的主要内容作出总结。

4.1 算法思路

首先给出一个典型的映射方案的工作流程，如图 4-2，首先对于虚拟网络拓扑图和物理网络拓扑图进行排序，然后根据排序顺序，把物理节点与虚拟节点建立一一对应关系，查看资源是否匹配，如果匹配成功，则更新物理网络，继续进行匹配，直至全部虚拟节点被映射完毕，进行链路映射，如果链路映射也完成，则接受这个虚拟网络请求，更新物理资源，本次映射成功。如果节点匹配失败或者链路映射失败，那么这个虚拟网络请求就会被拒绝，返还消耗的物理网络资源，本次映射失败退出。

可以发现，映射方案因为在映射时，不断地计算每一个虚拟节点和每一个物理节点的最佳匹配，因此在收益开销比方面，有着显著的优势，但是大部分传统的虚拟网络映射算法，主要有三个缺点。具体如下：第一点，大部分虚拟网络映射算法规模不够大，物理节点和虚拟节点往往为一对一的匹配。第二点，大部分虚拟网络映射算法，对于在节点排序时，往往对链路考虑的比较少，导致链路映射失败率较高。第三点，大部分虚拟网络映射算法的匹配关系，往往只计算物理网络节点能否容纳虚拟网络节点，不够细致。

针对传统的虚拟网络映射算法的缺点，参考映射方案的策略，本文将提出一种新的两阶段映射方案。在本方案中，针对虚拟节点和物理节点的映射，考虑网安试验的特殊性，一个物理节点可以匹配多个虚拟节点，使之适应大规模问题。在节点映射过程中，综合考虑节点和链路的关系进行排序，以此提高链路映射的成功率。然后根据排序关系进行匹配时，对于虚拟节点和物理节点，计算两者之间的契合度，根据契合度的大小，选择最适合的虚拟节点和物理节点进行匹配，而不是直接根据排序关系进行匹配，使得物理节点与虚拟节点的匹配更加精细化，提高收益开销比。

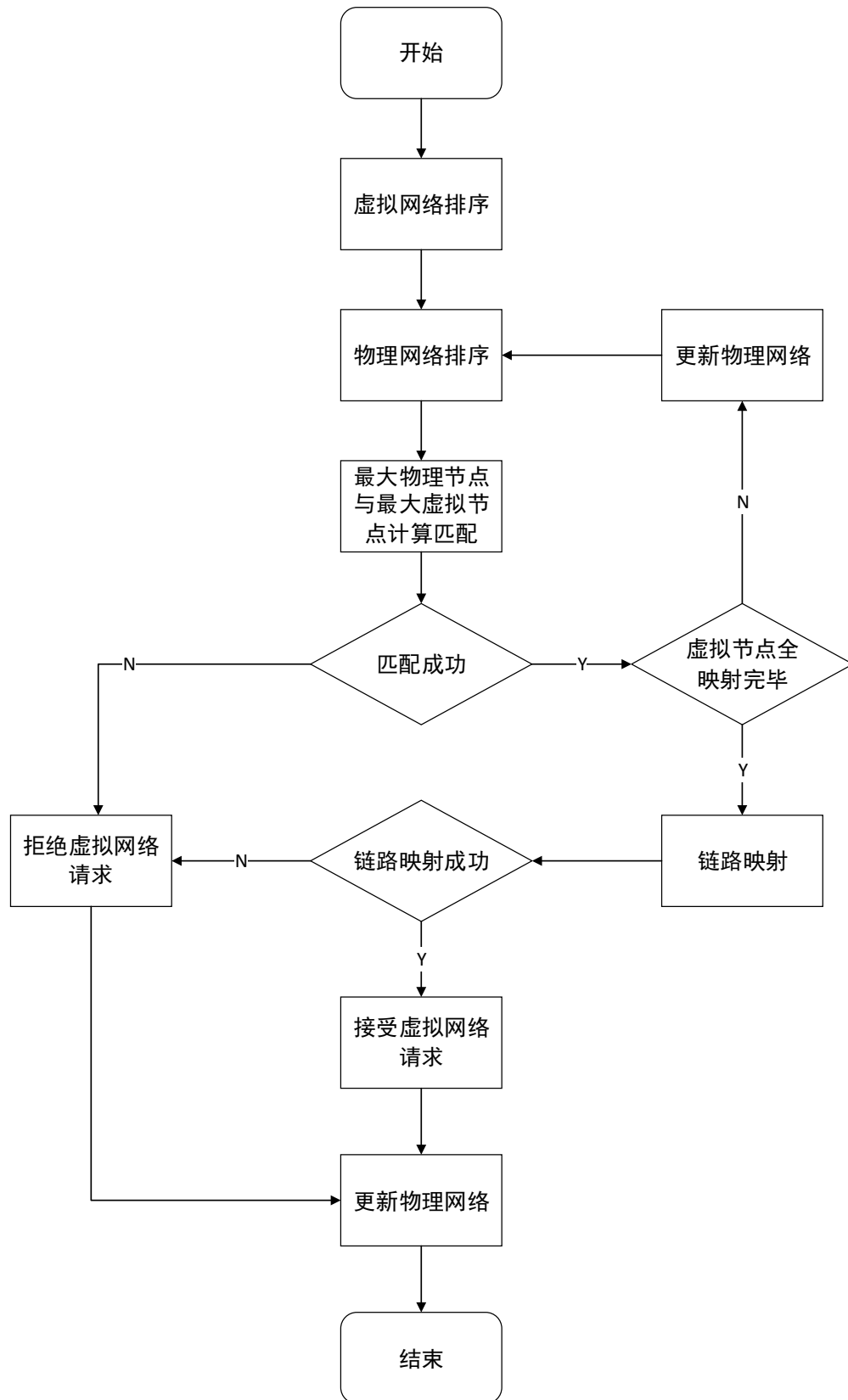


图 4-2 映射流程

4.2 节点映射过程

在这一节，对节点映射过程作出详细描述。如之前所描述的，节点映射过程主要由两部分组成，节点排序和节点匹配。首先把物理节点以及虚拟节点进行整理，然后排序，按照次序，依次把虚拟节点和物理节点之间进行匹配。本文将对这种思路进行改进，提出一个新的节点映射过程。

因为硬件的能力越来越强，一个物理节点往往能够容纳极多的虚拟节点，因此限制虚拟网络映射成功的关键点往往在于虚拟链路的映射，很多虚拟网络的映射中，都会发生节点的映射成功，但是受制于物理资源的剩余带宽，映射失败。因此，本文将在节点映射过程中，就考量链路，综合计量节点的容量与节点的链路来完成节点映射过程。

在节点排序阶段，本文改进排序的策略，提出新的节点排序公式，对节点的排序将不只是考量节点的权重，而是放大节点所链接的链路带来的影响，使得物理节点和虚拟节点的排序更加有效。

在节点匹配阶段，在大规模虚拟网络映射中，一个物理节点是要和多个虚拟节点进行匹配的，这就带来了两个问题，一个虚拟节点如何匹配一个物理节点，以及一次匹配后，如何选择下一个物理节点，传统的方法十分简单，就是不断的进行高次序物理节点匹配高次序虚拟节点，但是这种方法非常粗糙，没有考虑已映射的虚拟节点和未映射的虚拟节点的关系，同时，物理节点间的关系也没有考虑，映射效果较差。因此，本文提出一种节点契合度的计算方法，基于节点的契合度关系，为物理节点和虚拟节点之间的匹配提供指导。

4.2.1 节点排序

在节点排序阶段，主要工作就是分别把物理网络拓扑中的节点和虚拟网络拓扑中的节点排出次序，提供给节点匹配阶段使用，这就涉及到了两个问题，排序的依据是什么，以及排序的方法是什么。

首先是排序的依据，这也是节点排序阶段最重要的问题，对于一个网络拓扑图 $G = (V, V, C, C)$ 的一个节点 v ，目前的经典排序依据如公式 (4-1)：

$$P(v) = C(v) \cdot \sum_{e \in E(v)} W(e) \quad (4-1)$$

其中， $P(v)$ 就是节点 v 的排序计算值，本文称其为节点的性能，其计算方法就是节点 v 的处理能力，也就是权重 $C(v)$ 乘上节点所连接的全部链路的权重之和 $\sum_{e \in E(v)} W(e)$ 。

但是经过前面的分析可以得出，在节点映射的时候，关键点在于链路，因此应

当放大节点的链路属性对于节点性能计算的影响。由此，给出本文所采用的新的节点性能计算公式（4-2），对于节点 v ，其 $P(v)$ 计算方式如下：

$$P(v) = C(v) \cdot (\sum_{e \in E(v)} W(e) + \delta \cdot |E(e)|) \quad (4-2)$$

在公式（4-2）中， $P(v)$ 的值等于节点 v 的权重乘上节点 v 所连接的全部链路的权重之和加上系数 δ 乘以节点 v 所连接的全部链路的数量，其中系数 δ 是一个变量，它的意义是节点 v 所连接的全部链路的权重之和和节点 v 所连接的全部链路的数量处在一个数量级上，因为往往不同条件下对于节点链路的权重设置是不同的， δ 就是一个自适应的量。

分析公式（4-2）和（4-1）可以发现，相较于经典性能公式，在本文所提出的公式除了考虑了节点所连链路的权重之外，还考虑了节点所连接的链路的数量。因此，对于链路的计量，新公式（4-2）相对于老公式（4-1）更加的完善，链路的影响被放大了，符合之前对于映射过程的分析。

然后就是节点排序的方法，这和常规的排序问题并没有什么差别，因此根据性能计算公式算出的值，使用常规的排序方法排序就好。排序方法的研究已有很多了，这里就不针对排序算法展开说明了，在节点排序阶段采用的排序方法为快速排序[38]。

4.2.2 节点契合度匹配

在节点契合度匹配阶段，主要工作就是把物理节点和虚拟节点进行匹配。匹配阶段的两个问题，一个虚拟节点如何匹配一个物理节点，以及一次匹配后，如何选择下一个物理节点将在本节进行解答。

首先是节点之间的匹配问题。传统算法中，就是直接根据节点性能的次序，依次配对物理节点与虚拟节点。这种思路有一个明显的问题，就是没有考虑已经映射的节点和尚未映射的节点之间的链路关系，在这种情况下，对于直接相连的两个虚拟节点，它们可能不被分配在较近的物理节点上，甚至可能分布在很远的两个物理节点上，这就为之后的链路映射带来了麻烦。

因此，提出公式（4-3），作为物理节点和虚拟节点进行匹配的依据。对于物理网络拓扑 $G_s = (V_s, E_s, C_s, W_s)$ 和虚拟网络拓扑 $G_v = (V_v, E_v, C_v, W_v)$ ，当虚拟节点 v_v 被映射到物理节点 v_s ，设已经被映射到 v_s 中的虚拟节点集合为 V_{sv} ，其契合度 $CV(v_v \rightarrow v_s)$ 计算方法为如下：

$$CV(v_v \rightarrow v_s) = \frac{IN(v_v \rightarrow v_s)}{OUT(v_v \rightarrow v_s)} \quad (4-3)$$

其中，上式 $IN(v_v \rightarrow v_s)$ 的计算方法为公式（4-4），如下：

$$IN(v_v \rightarrow v_s) = \sum_{e \in E(v_v) \& e \in E(v_{sv})} W(e) \cdot |E(e)| \quad (4-4)$$

其含义为，把虚拟节点 v_v 映射到物理节点 v_s 后， v_v 和已经被映射到 v_s 中的虚拟节点的链路关系，计算方法为 v_v 和已经被映射到 v_s 中虚拟节点相连接的链路权重之和乘以链路数。

下式子 $OUT(v_v \rightarrow v_s)$ 的计算方法如公式（4-5），如下：

$$OUT(v_v \rightarrow v_s) = \sum_{e \in E(v_v) \& e \notin E(v_{sv})} W(e) \cdot |E(e)| \quad (4-5)$$

其含义为，把虚拟节点 v_v 映射到物理节点 v_s 后， v_v 和没有被映射到 v_s 中的虚拟节点的链路关系，计算方法为 v_v 和没有被映射到 v_s 中虚拟节点相连接的链路权重之和乘以链路数。

所以，整个契合度 $CV(v_v \rightarrow v_s)$ 的含义就是，将一个虚拟节点 v_v 映射到物理节点 v_s 之后，虚拟节点 v_v 对物理节点 v_s 内部产生的连接关系除以对节点外产生的连接关系。这个值越大，代表虚拟节点 v_v 和物理节点 v_s 已有的关联性越大，说明其契合程度越高。

然后说明一次匹配后，如何选择下一个物理节点。经过之前的分析，越多的链路被隐藏在节点内不需要映射，映射的成功率越高，并且映射的结果越好。因此，在物理节点还有剩余能力的情况下，它匹配到的虚拟节点越多，这个节点的利用率就越高，整体的资源利用率也会越高。从链路的角度讲，一个物理节点最大程度的放入虚拟节点，可以最大程度的减少对外的链路。因此，选择下一个物理节点的原则就是，只要这个物理节点还能继续承载虚拟节点，就继续选择这个物理节点，直至物理节点被装满。

综上，整个匹配的过程为，物理节点选择排序第一的，未匹配的虚拟节点全部按顺序遍历，依次计算每个未匹配的虚拟节点和该物理节点的契合度，选择最佳契合度的虚拟节点和物理节点匹配，直至该物理节点无法容纳虚拟节点，按照排序找到下一个物理节点。如果所有虚拟节点均被匹配，节点匹配结束。在这个过程中，如果发生任何一个物理节点都无法容纳剩余虚拟节点的时候，代表节点映射过程失败，拒绝这个虚拟网络。

4.3 链路映射过程

链路映射过程的目的是找到链路间的对应关系。

如之前所说，节点映射完成，对于链路映射而言，相当于链路的起点和终点都已经被确定了，只要计算出两点之间符合带宽要求的最短路径，就完成了链路映射过程。

算法 4-1 BWDijkstra Algorithm

Input: 起点 v_{source} , 终点 $v_{destiny}$, 带宽限制 $bwlimit$

基础物理资源 $SR = \{SN, SC, SW\}$, $SN = (V_s, E_s, C_s, W_s)$

Output: 最短路径 $path[]$

```

1  初始化最短路径数组  $dist[]$ , 全部置为 MAX_VALUE
2  初始化最短路径前驱节点数组  $prev[]$ , 全部置为-1
3  初始化节点访问数组  $select[]$ , 全部置为  $false$ 
4  for  $v_t$  in  $V_s$  do
5      if (  $v_t$  与  $v_{source}$  直接相连 and 连接链路剩余带宽大于  $bwlimit$  )
6           $dist[v_t] = 1$ 
7      end if
8  end for
9   $dist[v_{source}] = 0$ ,  $select[v_{source}] = true$ 
10  $i = 0$ 
11 while (  $i < |V_s| - 1$  )
12      $min = MAX\_VALUE$ 
13     for  $v_t$  in  $|V_s|$  do
14         if (  $select[v_t] = false$  and  $dist[v_t] < min$  )
15              $min = dist[v_t]$ 
16              $v_{min} = v_t$ 
17         end if
18     end for
19      $visited[v_{min}] = true$ 
20     for  $v_t$  in  $V_f - v_{min}$  do
21         if (  $SW( v_{min}, v_t ) > bwlimit$  and  $dist[i] < dist[v_{min}] + 1$  )
22              $dist[i] = dist[v_{min}] + 1$ 
23              $prev[i] = v_{min}$ 
24         end if
25     end for
26      $i = i + 1$ 
27 done
28 根据  $prev$ , 反向递推  $prev[v_{destiny}]$  到  $v_{source}$  得到  $path[]$ 

```

在这里。本文通过对 Dijkstra 算法^[39]进行优化，形成链路映射算法。直接利用 Dijkstra 算法计算最短路径会产生一个问题，Dijkstra 算法在计算的时候只考虑最短路径，因此计算出的最短路径需要检查是否符合带宽约束，如果不符合带宽约束，最短路径还要重新计算，这就导致链路映射的开销增大。因此，本文对 Dijkstra 算法进行优化，在寻找最短路径的过程中就考虑带宽限制，如果一条链路的距离满足约束，但是带宽约束没有符合条件，则直接舍弃该链路，不纳入最短路径的计算。

如算法 4-1，BWDijkstra 算法就是优化后的 Dijkstra 算法。算法的整体过程和 Dijkstra 算法相同，维护最短路径数组，前驱节点数组，和访问数组。然后开始一轮一轮的迭代计算最短路径，但更新最短路径的条件有所变化，只有当更新路径的带宽大于最小带宽限制的时候，最短路径才会更新，算法的时间复杂度为 $O(V_s^2)$ 。

根据 BWDijkstra 算法，如果找到了路径，就找到了满足带宽限制的最短路径，映射成功。如果找不到路径，代表不存在满足带宽限制的路径，映射失败，这个虚拟网络就会被拒绝。

4.4 映射算法整体分析

在链路映射过程完成后，整个映射过程就完成了，由此就得到了基于节点契合度的映射过程的整个处理流程。

映射过程首先进行节点映射过程，基于公式（4-2）进行节点的排序，然后依据节点排序的顺序，基于公式（4-3）依次进行节点契合度匹配，匹配完成后，节点映射过程结束。接下来进行链路映射过程，依照算法 4-1，完成链路的映射，至此，整个的映射过程结束，如果之前的任何一步失败，那么这个虚拟网络请求映射失败，否则虚拟网络映射成功。

根据上述过程描述，给出整个映射算法如算法 4-2，本文将该映射算法称为 OPTMap。

算法的输入部分就是两部分，物理网络资源 SR 和虚拟网络请求图 G_v 。开始后首先对虚拟网络 G_v 进行上一章给出的前处理过程，再利用之前给出的性能计算公式（4-2），分别计算物理网络图和虚拟网络请求图中的节点性能，并进行由大到小的排序，得到两个数组 PS 和 PV 。接下来找到 PS 数组中第一个满足 $PV[1]$ 约束的节点 v_{st} ，然后对于这个节点 v_{st} ，在 PV 中寻找可满足约束且与已经映射到 v_{st} 中虚拟节点相连的节点，计算 CV 值，然后把 CV 值最高，即契合度最好的节点嵌入 v_{st} 中，如此则一个节点映射成功，然后往复进行映射，节点映射完成后，利用 BWDijkstra 算法计算链路。

算法 4-2 OPTMap Algorithm

Input: 基础物理资源 $SR = \{SN, SC, SW\}$, 其中 SN 为 $G_s = (V_s, E_s, C_s, W_s)$

虚拟网络拓扑 $G_v = (V_v, E_v, C_v, W_v)$

Output: 节点映射集合 $R(V)$, 链路映射集合 $R(E)$

```

1  对于  $SR$ , 计算  $P(V_s)$ , 对节点由大到小排序得到  $PS[]$ 
2  对于  $G_v$ , 前处理  $G_v$ , 计算  $P(V_v)$ , 对节点由大到小排序得到  $PV[]$ 
3  寻找  $PS$  中第一个满足  $PV[0]$  约束的节点  $v_{st}$ 
4  for  $i = 0$  to  $|V_v|$  do
5      for  $v_{vt}$  in  $PV[]$  do
6          if ( $R(V)$  不包含  $v_{vt}$  and  $v_{vt}$  和  $v_{st}$  已映射虚拟节点相连接 )
7              if ( $v_{st}$  满足  $v_{vt}$  约束 )
8                  计算  $CV(v_{vt} \rightarrow v_{st})$ 
9              end if
10             end if
11         end for
12         if ( 存在  $max(CV)$  )
13              $R(V) : max(CV)$  的映射
14             清空  $CV[]$ 
15         else
16             删除  $PS[v_{st}]$ 
17         end if
18         if ( $v_{st} = NULL$  and 找到  $PS$  中第一个满足  $PV[0]$  约束的节点  $v_{st}$  )
19             continue
20         else
21             拒绝  $G_v$  并退出
22         end if
23     end for
24     if ( 使用 BWDijkstra 算法计算链路映射成功 )
25         链路映射结果保存在  $R(E)$  中
26     else
27         拒绝  $G_v$  并退出
28     end if

```

分析 OPTMap 算法，在节点排序的时候，放大了链路的影响，然后在节点匹配的时候，计算匹配度，并且优先把虚拟节点放入一个物理节点，这些优化的目的，都是减小链路映射的压力。根据之前的分析，映射成功率和映射代价的主要影响因素都是链路，因此，这些优化方式有助于提高整体的映射成功率和收益开销比，本文将在下一章，利用实验证明这一点。

算法进行映射时，前处理过程的时间复杂度为 $O(|E_v|)$ ，前处理之后节点为 K ，排序部分的时间复杂度 $O(K^2)$ ，匹配部分是一个基于虚拟网络节点数的两重循环，时间复杂度为 $O(K^2)$ ，进行链路映射时，映射一条链路的时间复杂度为 $O(|V_s|^2)$ ，一共需要 $O(|V_v| \times |V_s|^2)$ ，因此，OPTMap 算法的时间复杂度为 $O(|V_v| \times |V_s|^2)$ 。

4.5 本章小结

在本章中，主要给出了基于节点契合度的映射过程，并结合前文分析已有映射方案的分析，说明本文的所给出的算法的改进点在哪里。

首先是对映射过程给出了整体的描述，映射过程主要由节点映射过程和链路映射过程组成。然后分析了映射过程的关键点在于链路映射，因此，应该在节点映射的时候，就考虑链路关系，使得两个过程之间的关联性更强，映射效果更好。由此，本文给出了新的节点排序公式和节点契合度公式，在节点映射时，就考虑链路关系的影响，并针对链路映射过程，提出了优化的链路映射算法 BWDijkstra。

最后，给出映射过程整体的算法 OPTMap，并对算法作出分析，对照算法的细节说明算法的优化点。

第 5 章 虚拟网络映射算法的实验研究

5.1 虚拟网络映射系统

5.1.1 模块划分

纵观以往的大部分虚拟网络映射算法论文，大多采用模拟实验来验证算法的效果，虚拟网络请求都是人工生成的。为了验证本文所提出的算法的效果，结合一些拓扑生成算法的分析^[40-44]，本文将采用模拟实验和真实拓扑实验相结合的办法，从两方面来展示算法的优越性。

首先给出本文搭建的虚拟网络映射系统，如图 5-1。虚拟网络映射系统由虚拟网络控制器和映射控制器两部分组成，接下来详细说明其功能。

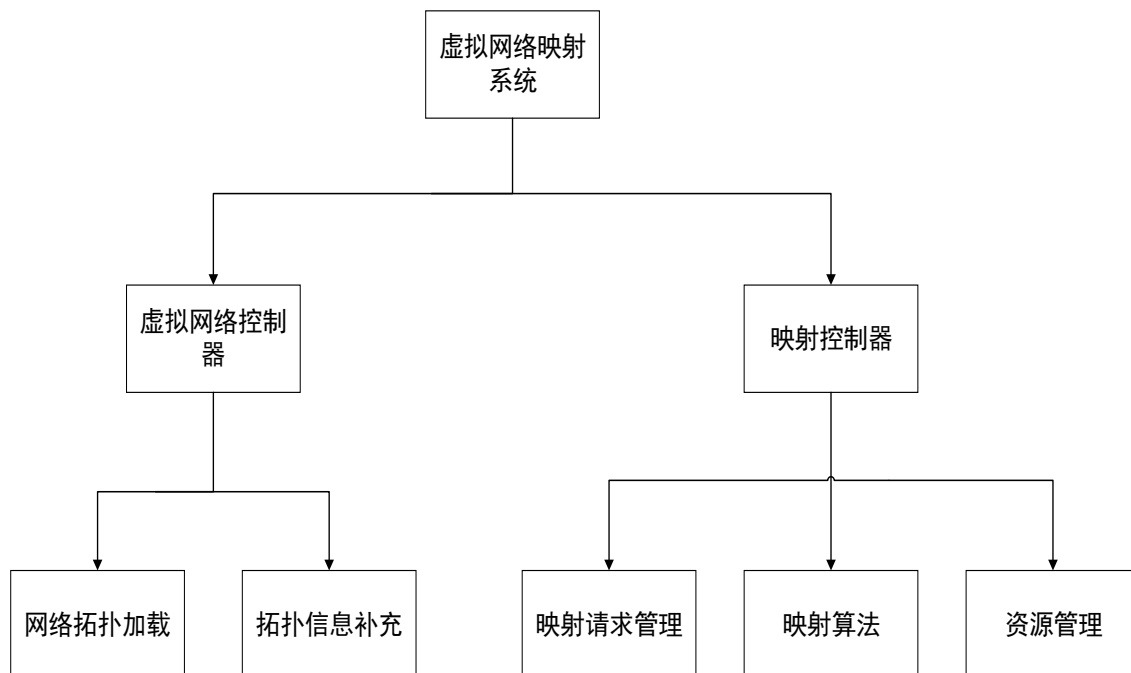


图 5-1 虚拟网络映射系统

首先是虚拟网络控制器，虚拟网络控制器由两部分组成，网络拓扑加载和拓扑信息补充。

（1）网络拓扑加载：网络拓扑加载模块的主要功能就是根据需要加载网络拓扑信息。首先是物理网络，本文采用网络生成工具 GT-ITM^[40]生成一个 1000 个节点，4000 条链路的物理网络，物理网络在程序开始时一次加载即可。然后是虚拟网络，虚拟网络拓扑信息主要分为两个部分，人工模拟拓扑和真实网络拓扑。人工模拟拓扑是采用波士顿大学的 BRITE 拓扑生成工具^[41]来生成，它可以根据需要，

生成指定节点数和链路数的拓扑图。真实网络拓扑则采取真实的网络拓扑数据，其来源是 DNS 网络拓扑采集项目，需要提前下载好。虚拟网络拓扑在运行中不断地加载。

(2) 拓扑信息补充：拓扑信息补充模块的功能是补充网络拓扑加载模块所生成的拓扑图的信息，因为实验部分会进行多种规模的实验，不同规模的实验网络拓扑权重的设置也有所不同。本模块主要包含节点权重的补充和链路权重的补充两部分，具体的数值设置在实验中给出。

然后是映射控制器，映射控制器由三部分组成，映射请求管理，映射算法控制和资源管理。

(1) 映射请求管理：映射请求管理模块的功能是管理进入的虚拟网络请求。虚拟网络控制器生成虚拟网络请求后，映射请求管理模块负责实际控制虚拟网络请求的映射，模拟泊松分布控制虚拟网络请求的到达时间，并控制虚拟网络请求的物理网络留存时间。在映射时间内，到来多少个虚拟网络请求在实验中具体给出。

(2) 映射算法控制：映射算法控制模块的功能就是实际选择算法进行映射。为了对比实验，本系统提供三种算法，经典的 Node-opt 算法，带有前处理过程的 Pre-node-opt 算法以及本文提出的算法 OPTMap。

(3) 资源管理：资源管理模块负责实时更新物理网络资源的使用状况。映射算法控制模块在映射时，资源的改变都通过资源管理模块完成。同时，一个虚拟网络请求退出时，也通过资源管理模块返还消耗的物理资源。

5.1.2 工作流程

了解整个虚拟网络映射系统的模块划分之后，在本节给出虚拟网络映射系统的工作流程。

如图 5-2，虚拟网络映射系统开始工作后，首先载入物理网络资源，读入虚拟网络请求。每当进入一个虚拟网络，首先判断是否达到了物理网络资源容量的设置阈值，如果达到，则退出，否则进行虚拟网络的映射。然后根据映射的结果，如果这个虚拟网络映射失败，直接读入下一个虚拟网络，如果映射成功，模拟泊松分布进行映射控制，根据具体设置，控制虚拟网络的到来时间与退出时间。

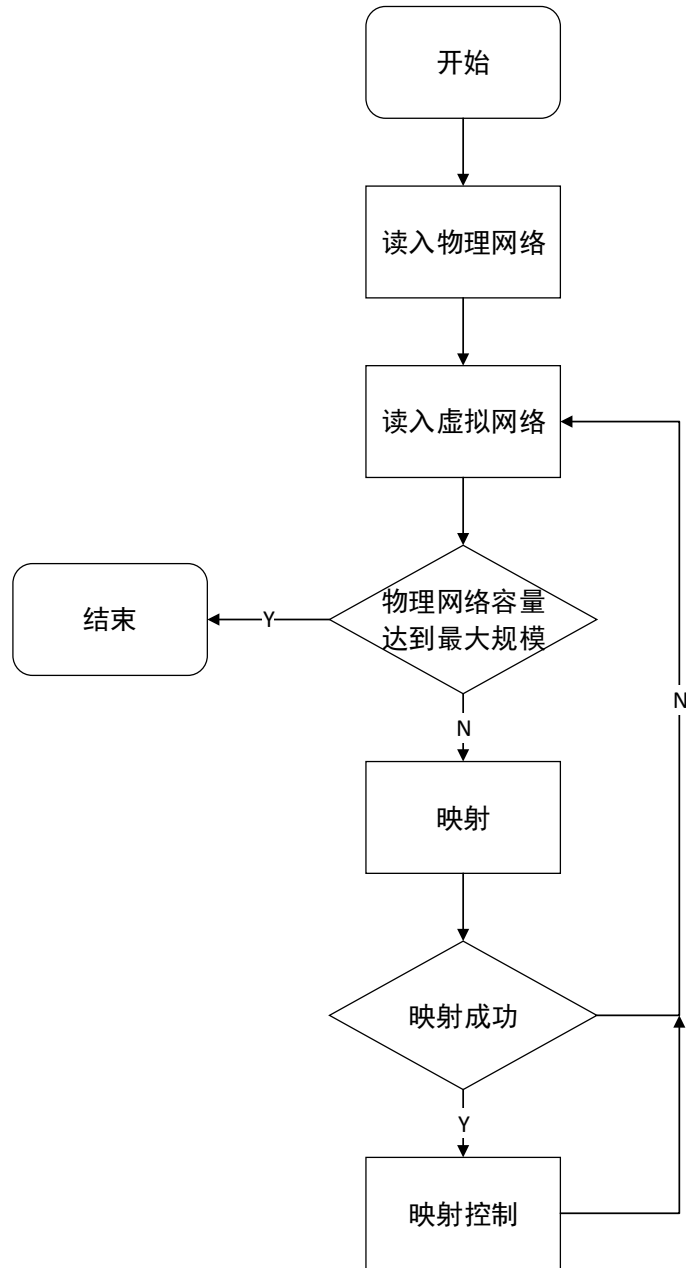


图 5-2 系统流程图

5.2 映射对比实验

在这一节，本文进行映射算法的对比实验，所对比算法为经典的 Node-opt 算法，带有本文前处理过程的 Pre-node-opt 算法，以及本文提出的 OPTMap 算法。对比实验分为两种，首先进行模拟网络映射实验，然后再进行真实拓扑映射实验。所有实验均为 5 次实验取平均值，减小偶然因素的影响。

模拟网络实验分为 4 个规模，按照虚拟网络请求节点数量，分为 50 个节点到

100 个节点的小规模实验，500 个节点到 1000 个节点的中等规模实验，5000 个节点到 10000 个节点的大规模实验，50000 个节点到 100000 个节点的超大规模实验，每个节点随机生成 2 到 4 条边。真实拓扑实验则采用真实的数据。物理网络规模在实验中保持不变，物理网络节点数为 1000，物理链路数为 4000。

5.2.1 小规模实验

小规模实验的数值设置如下。

首先是物理网络，物理网络节点权重设置为 1，有 10% 的波动，服从高斯分布。物理链路权重设置为 10，有 10% 的波动，服从高斯分布。物理网络中不存在孤点。

然后是虚拟网络，虚拟网络节点数为 50 变化到 100，每次递增 10，节点权重设置为 0.1，有 10% 的波动，服从高斯分布。虚拟链路权重设置为 1，有 10% 的波动，服从高斯分布。虚拟网络中不存在孤点。虚拟请求每组 20 个，到达服从泊松分布。

因为小规模实验的规模过小，前处理过程没有意义，反而会降低映射效果，因此不采用前处理过程进行处理。

由实验得出的虚拟网络请求接受率如表 5-1。

表 5-1 小规模实验接受率

节点数	Node-opt	OPTMap
50	100%	100%
60	95%	100%
70	80%	100%
80	50%	95%
90	40%	75%
100	30%	80%

根据表 5-1，绘制虚拟网络请求接受率的折线图，使结果更加简单明了，具体如图 5-3。

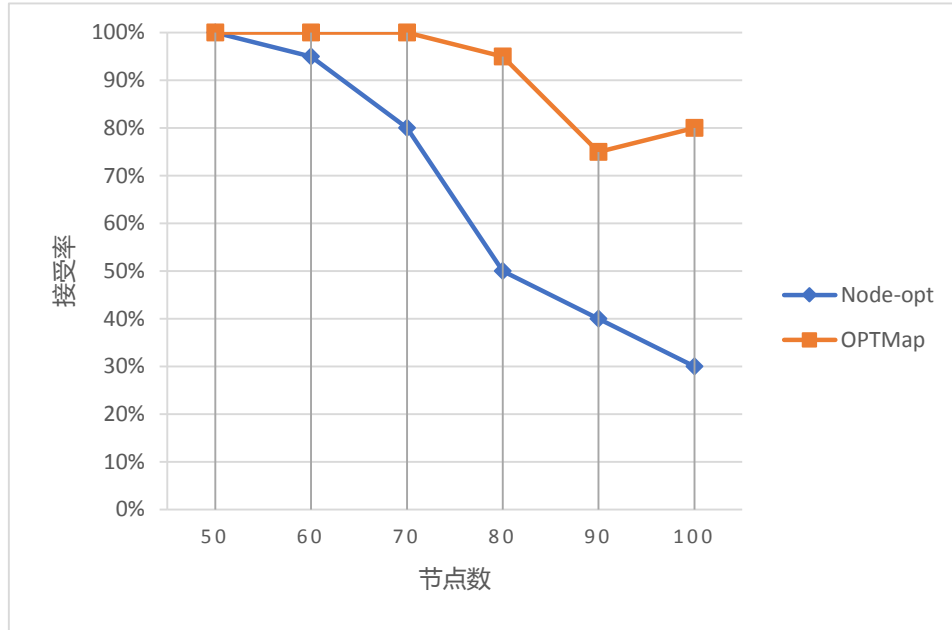


图 5-3 小规模实验接受率

从图中可以发现，当虚拟网络请求的规模较小时，Node-opt 算法与 OPTMap 算法的成功率差别不大，但当虚拟网络请求的规模逐渐增大时，OPTMap 算法开始明显的优于 Node-opt 算法，或者说物理网络的压力越大时，两者的差距越大。

根据评价函数计算具体收益值 R 与开销值 C ，结果如表 5-2。具体的收益开销比如表 5-3。

表 5-2 小规模实验收益与开销(R/C)

节点数	Node-opt(R/C)	OPTMap(R/C)
50	3014/5454	3014/2949
60	3614/6375	3806/4066
70	3445/6567	4313/6295
80	2422/5366	4609/7813
90	2196/4738	4117/6687
100	1863/3586	4943/6986

表 5-3 小规模实验收益开销比

节点数	Node-opt	OPTMap
50	0.553	1.022
60	0.567	0.936
70	0.525	0.685
80	0.451	0.590
90	0.463	0.616
100	0.520	0.708

根据表 5-3，绘制收益开销比的折线图，如图 5-4。

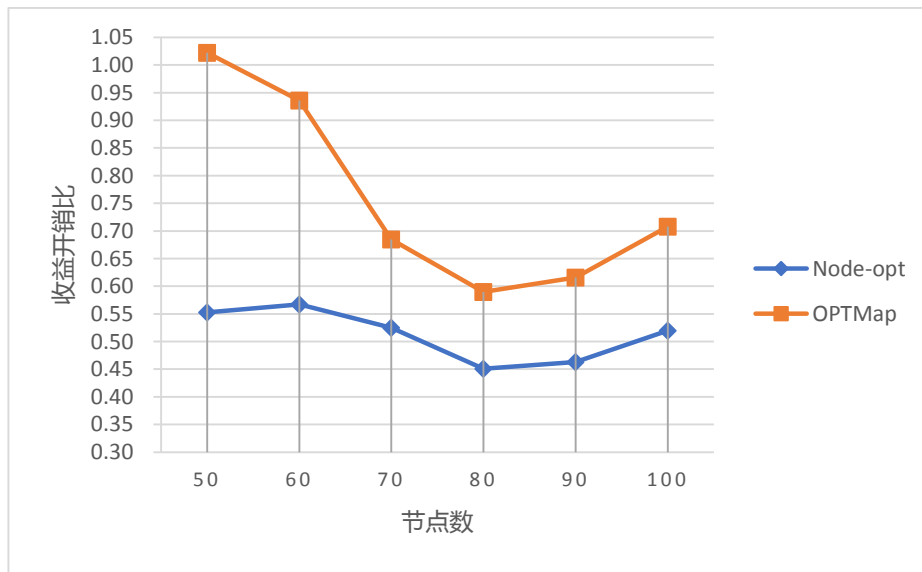


图 5-4 小规模实验收益开销比

从图中可以看出。在收益开销比方面，**OPTMap** 算法明显的优于 **Node-opt** 算法。结合之前的接受率数据，可以得出结论，在小规模实验中，**OPTMap** 算法优于 **Node-opt** 算法。

5.2.2 中等规模实验

中等规模实验的数值设置如下。物理网络节点权重设置为 1，有 10%的波动，服从高斯分布。物理链路权重设置为 10，有 10%的波动，服从高斯分布。物理网

络中不存在孤点。虚拟网络节点数为 500 变化到 1000，每次递增 100，节点权重设置为 0.01，有 10% 的波动，服从高斯分布。虚拟链路权重设置为 0.1，有 10% 的波动，服从高斯分布。虚拟网络中不存在孤点。虚拟请求每组 20 个，到达服从泊松分布。

如之前分析，当节点数增多，链路数也会随之增多，因此采用前处理过程，对比的算法包括 Node-opt 算法，带前处理过程的 Pre-node-opt 算法，以及本文提出的 OPTMap 算法。虚拟网络请求接受率如表 5-4。

表 5-4 中等规模实验接受率

节点数	Node-opt	Pre-node-opt	OPTMap
500	100%	100%	100%
600	90%	100%	100%
700	65%	100%	100%
800	45%	100%	100%
900	30%	95%	100%
1000	30%	95%	100%

根据表 5-4，绘制接受率折线图，如图 5-5。

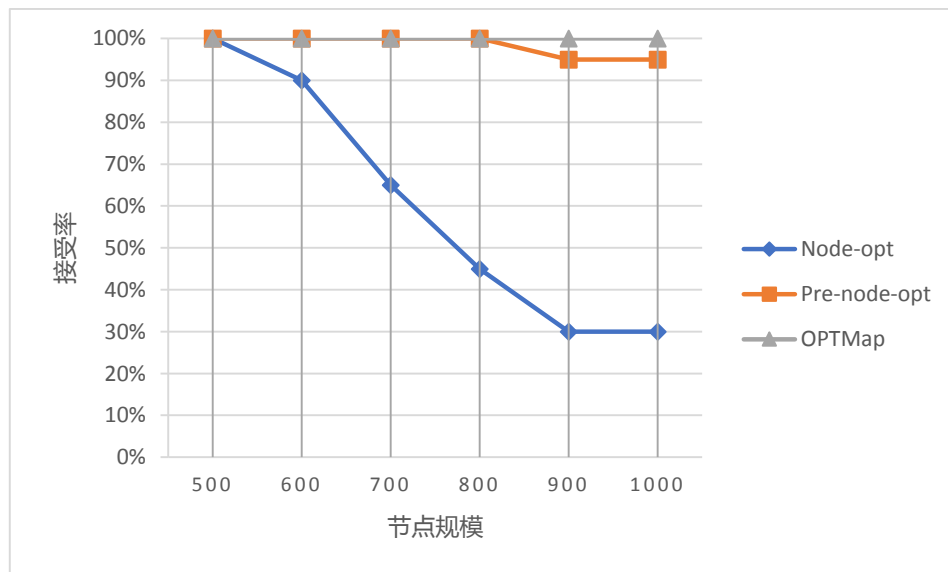


图 5-5 中等规模实验接受率

可以发现,当节点规模变大时,Node-opt 算法失败率变得非常高,因为几百个节点,对于传统的算法已经算是大规模了。在加入了前处理过程后,Pre-node-opt 算法接受率被提高了,但本文的 OPTMap 算法更高一点。

接下来给出中等规模实验收益与开销的数据,表 5-5 展示的是收益 R 与开销 C,表 5-6 是计算出的收益开销比。

表 5-5 中等规模实验收益与开销(R/C)

节点数	Node-opt(R/C)	Pre-node-opt(R/C)	OPTMap(R/C)
500	4701/8538	4701/5570	4701/5448
600	4866/10049	5403/7329	5403/7094
700	4160/8734	6411/9571	6411/9029
800	3258/7136	7241/11932	7241/10920
900	2436/5385	7714/13439	8120/12855
1000	2694/5989	8531/14993	8980/14332

表 5-6 中等规模实验收益开销比

节点数	Node-opt	Pre-node-opt	OPTMap
500	0.551	0.844	0.862
600	0.484	0.737	0.762
700	0.476	0.670	0.710
800	0.457	0.607	0.663
900	0.452	0.574	0.632
1000	0.450	0.569	0.627

根据表 5-6,给出收益开销比的折线图,如图 5-6。

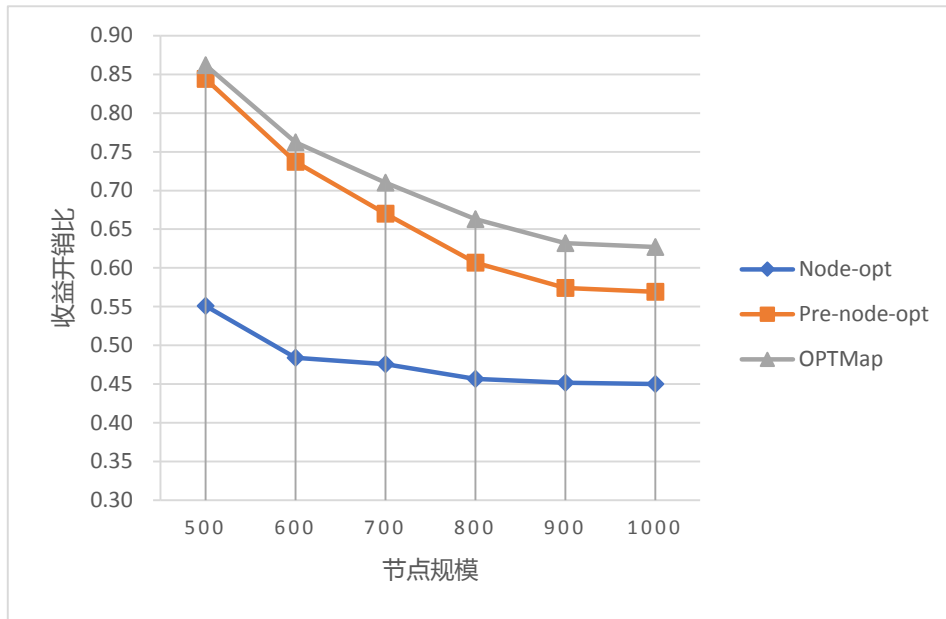


图 5-6 中等规模实验收益开销比

从图中可以看出，在中等规模实验时，加入前处理过程的 Pre-node-opt 算法的收益开销比远胜于 Node-opt 算法，但是本文提出的 OPTMap 算法收益开销比较之 Pre-node-opt 算法更优。随着节点数增多，压力的增大，三个算法的收益开销比都有所降低，但 OPTMap 算法一直保持领先。

5.2.3 大规模实验

接下进行大规模实验，具体的数值设置如下。

首先是物理网络，物理网络节点权重设置为 5，有 10% 的波动，服从高斯分布。物理链路权重设置为 50，有 10% 的波动，服从高斯分布。物理网络中不存在孤点。

然后是虚拟网络，虚拟网络节点数为 5000 变化到 10000，每次递增 1000，节点权重设置为 0.01，有 10% 的波动，服从高斯分布。虚拟链路权重设置为 0.1，有 10% 的波动，服从高斯分布。虚拟网络中不存在孤点。虚拟请求每组 20 个，到达服从泊松分布。

如前文所说，本文提出的算法就是要解决大规模问题，因此重点关注大规模实验的结果。同时也在这里说明，5000 节点到 10000 一万节点这种规模的实验，已经不在传统算法的范围内。

大规模实验对比的算法包括 Node-opt 算法，带前处理过程的 Pre-node-opt 算法，以及本文提出的 OPTMap 算法。

结果如下，虚拟网络请求接受率如表 5-7。

表 5-7 大规模实验接受率

节点数	Node-opt	Pre-node-opt	OPTMap
5000	55%	100%	100%
6000	45%	100%	100%
7000	35%	100%	100%
8000	20%	100%	100%
9000	25%	90%	95%
10000	20%	75%	90%

根据表 5-7，绘制接受率折线图，如图 5-7.

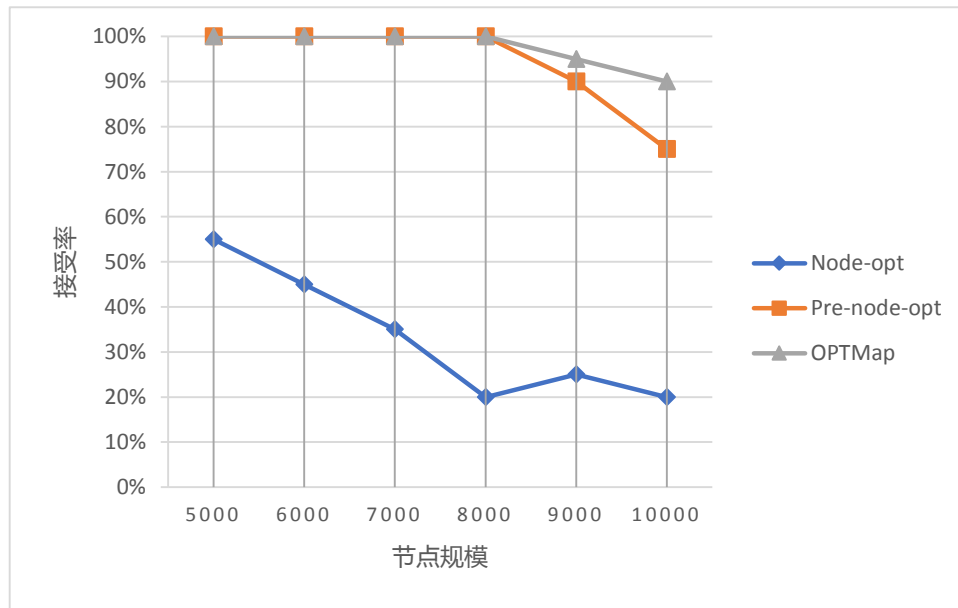


图 5-7 大规模实验接受率

可以发现，在大规模实验时，Node-opt 算法失败率变得非常的高，因为这种实验规模已经不是传统的算法能够处理的范围了。在加入了前处理过程后，Pre-node-opt 算法的接受率大幅度提高了，证明了前处理过程的有效性，但本文的 OPTMap 算法更好。

接下来给出大规模实验收益 R 与开销 C 的数据，表 5-8 是收益与开销，表 5-9 是计算出的收益开销比。

表 5-8 大规模实验收益与开销(R/C)

节点数	Node-opt(R/C)	Pre-node-opt(R/C)	OPTMap(R/C)
5000	24167/59742	43940/72151	43940/67535
6000	23562/61183	52360/93002	52360/84660
7000	21301/54646	60860/116590	60860/103794
8000	13838/34242	69240/133447	69240/121684
9000	19409/51066	69948/136885	73834/132709
10000	17232/44806	64620/127238	77544/143246

表 5-9 大规模实验收益开销比

节点数	Node-opt	Pre-node-opt	OPTMap
5000	0.405	0.609	0.651
6000	0.385	0.563	0.618
7000	0.390	0.522	0.586
8000	0.404	0.519	0.569
9000	0.380	0.511	0.556
10000	0.385	0.508	0.541

根据表 5-9，给出收益开销比的折线图，如图 5-8。从图中可以看出，在大规模实验时，Node-opt 算法的效果十分一般。前处理过程的 Pre-node-opt 算法的收益开销比较之 Node-opt 算法有了很大提高，但是本文提出的 OPTMap 算法收益开销比明显更优。

随着节点数增多，压力逐渐增大，Pre-node-opt 算法，Node-opt 算法和 OPTMap 算法的收益开销比差距变小，这是因为前两者的成功率比 OPTMap 算法要低，因此物理网络的压力比 OPTMap 算法小，所以差距就变小了。

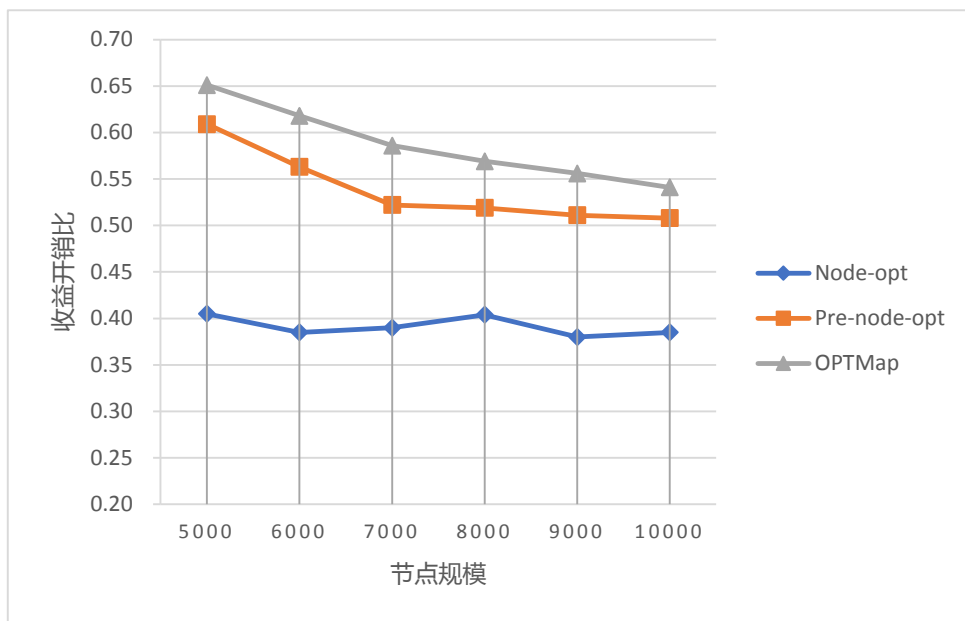


图 5-8 大规模实验收益开销比

5.2.4 超大规模实验

在超大规模实验中，传统的 Node-opt 算法已经无法成功映射任何一个虚拟网络请求，因此重点关注 Pre-node-opt 算法和 OPTMap 算法的效果。实验具体参数设置如下。

物理网络节点权重设置为 25，有 10% 的波动，服从高斯分布。物理链路权重设置为 250，有 10% 的波动，服从高斯分布。物理网络中不存在孤点。

虚拟网络节点数从 50000 个节点变化到 100000 个节点，每次递增 10000。节点权重设置为 0.01，有 10% 的波动，服从高斯分布。虚拟链路权重设置为 0.1，有 10% 的波动，服从高斯分布。虚拟网络中不存在孤点。虚拟请求每组 20 个，到达服从泊松分布。

首先给出虚拟网络请求接受率，如表 5-10，然后根据表中数据，绘制接受率的折线图 5-9。

从实验结果可以看出，在超大规模实验的情况下，OPTMap 算法与 Pre-node-opt 算法的接受率是非常接近的，Pre-node-opt 算法已经出现了优于 OPTMap 算法的情况。但是总体上来讲，OPTMap 算法还是略微强于 Pre-node-opt 算法。原因来自于启发式算法的特性，在真正规模特别大的情况下，启发式算法的效果往往差得不多。

表 5-10 超大规模实验接受率

节点数	Node-opt	Pre-node-opt	OPTMap
50000	0	95%	100%
60000	0	75%	80%
70000	0	60%	60%
80000	0	55%	50%
90000	0	40%	40%
100000	0	35%	35%

因为 Node-opt 算法无法成功，因此只对 Pre-node-opt 算法和 OPTMap 算法画出折线图，如下图 5-9。可以发现，OPTMap 算法的接受率略微强于 Pre-node-opt 算法。

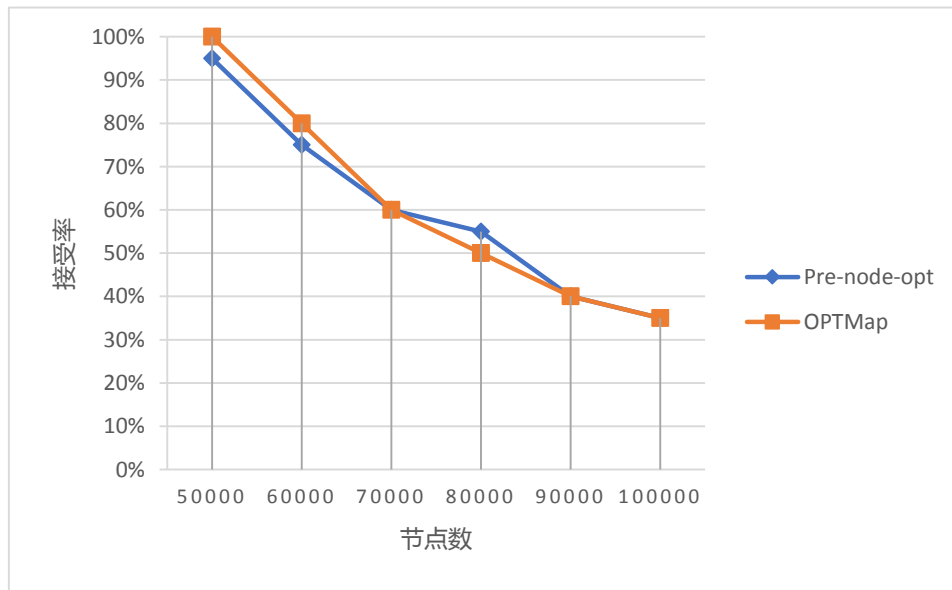


图 5-9 超大规模实验接受率

接下来给出超大规模实验的收益与开销数据，表 5-11 展示的是超大规模实验得出的收益值与开销值，表 5-12 是根据收益值与开销值计算出的收益开销比。具体如下表。

表 5-11 超大规模实验收益与开销(R/C)

节点数	Node-opt	Pre-node-opt(R/C)	OPTMap(R/C)
50000	0/0	383673/ 787889	403867/828344
60000	0/0	364141/ 745887	388416/ 794786
70000	0/0	341968/690578	341968/686990
80000	0/0	357309/734138	324827/655835
90000	0/0	292587/583409	292587/584904
100000	0/0	284769/573112	284769/571100

表 5-12 超大规模实验收益开销比

节点数	Node-opt	Pre-node-opt	OPTMap
50000	0	0.487	0.488
60000	0	0.488	0.489
70000	0	0.495	0.498
80000	0	0.487	0.495
90000	0	0.502	0.500
100000	0	0.497	0.499

收益开销比的实验结果与接受率类似,Node-opt 因为规模过大无法成功映射,OPTMap 算法与 Pre-node-opt 算法的收益开销比是非常的接近,但是总体上来观察,OPTMap 算法还是略微强于 Pre-node-opt 算法。和之前的分析相同,当实验的规模很大时,启发式算法的效果都差不太多。

根据表 5-12,给出 OPTMap 算法与 Pre-node-opt 算法收益开销比的折线图。图中可以看出,两个算法十分的接近,但 OPTMap 算法总体更好。

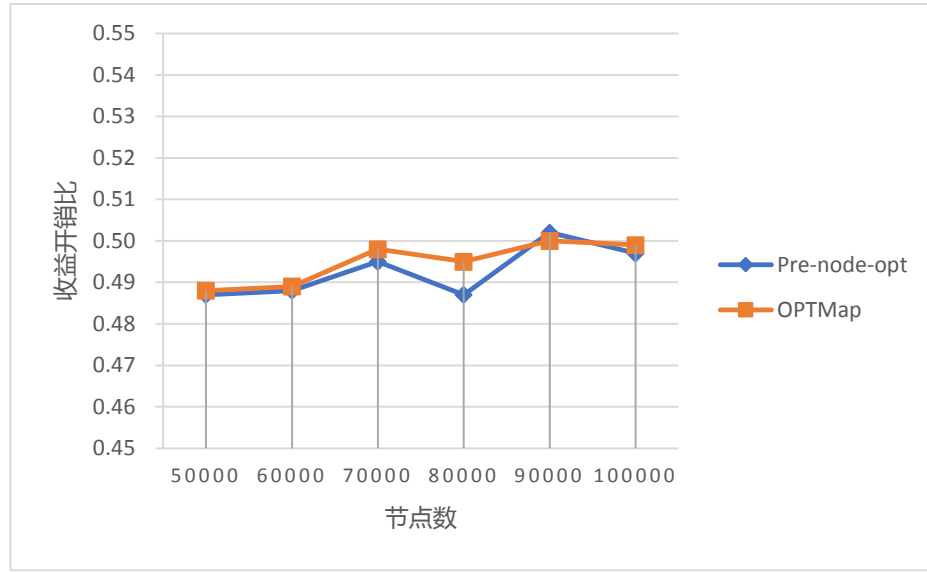


图 5-10 超大规模实验收益开销比

5.2.5 真实拓扑映射实验

在真实拓扑映射实验中，选取两个 DNS 拓扑网络测量的真实拓扑进行映射实验，进行定量分析。

物理网络节点权重设置为 1，有 10% 的波动，服从高斯分布。物理链路权重设置为 10，有 10% 的波动，服从高斯分布。物理网络中不存在孤点。

拓扑 1 节点数为 49531，节点权重设置为 0.01，有 10% 的波动，服从高斯分布。虚拟链路数为 60736，链路权重设置为 0.1，有 10% 的波动，服从高斯分布。拓扑 2 节点数为 49373，节点权重设置为 0.01，有 10% 的波动，服从高斯分布。虚拟链路数为 60136，链路权重设置为 0.1，有 10% 的波动，服从高斯分布。拓扑网络中不存在孤点。

因为 Node-opt 算法无法映射这种规模的网络，因此采用 Pre-node-opt 算法和 OPTMap 算法作对比。虚拟网络请求包含拓扑 1 和拓扑 2 两个网络。映射结果如表 5-13:

表 5-13 映射结果

拓扑网络	Pre-node-opt	OPTMap
拓扑 1	映射成功	映射成功
拓扑 2	映射失败	映射成功

然后给出具体的映射收益与映射开销。

表 5-14 映射收益

拓扑网络	Pre-node-opt	OPTMap
拓扑 1	10967	10967
拓扑 2	0	10344
合计	10967	21311

表 5-15 映射开销

拓扑网络	Pre-node-opt	OPTMap
拓扑 1	14514	14504
拓扑 2	0	11247
合计	14514	25751

收益开销比如表 5-16。

表 5-16 收益开销比

拓扑网络	Pre-node-opt	OPTMap
拓扑 1	0.756	0.756
拓扑 2	0	0.920
合计	0.756	0.828

从上面三个表中，可以显然看出，OPTMap 算法的效果比较好。拓扑 1 和拓扑 2 一起被映射时，Pre-node-opt 算法仅成功映射了拓扑 1。对于收益与开销 OPTMap 算法比 Pre-node-opt 算法映射的开销更小。因此可以证明，相比经典算法，OPTMap 算法的资源利用率更高，可以应对负载高的情况。

5.2.6 实验结果分析

分析上述的实验结果可以得出, 对于模拟实验, 经典的 Node-opt 算法随着虚拟网络请求的规模增大, 接受率下降的非常快, 收益开销比也十分一般, 当规模大到一定程度, Node-opt 算法就无法映射成功了。通过把 Node-opt 算法加上本文提出的前处理过程, 因为前处理过程把图初步聚合, 因此减小了图的规模, 使得链路映射的成功率提高, 所以 Pre-node-opt 算法的映射效果变好, 前处理过程是十分有效的。但是总体上因为 OPTMap 算法采用了新的节点排序方法和节点契合度计算方法, 增强了节点匹配的效果, 提高了节点映射与链路映射的联系, 所以 OPTMap 算法在接受率和收益开销比方面都是最优的。

对于真实拓扑, 选取两个真实网络的拓扑数据, Node-opt 算法无法成功, 因此采用 Pre-node-opt 算法, 但明显发现, OPTMap 算法的开销更小, 成功率更高。综上, 可以得出结论, 前处理过程的初步节点聚合对于映射过程是十分有效的, 新的节点排序公式和节点契合度公式的效果也很好, 本文的 OPTMap 算法的效果是十分优秀的。

当然, 也要看到, 在真实的系统映射时, 将不仅仅只是映射收益的问题, 还有一些其他问题。其一, 物理网络中的物理主机可能不是全部一样的, 因此会存在异构性的问题。其二, 在实验中, 某些关键节点如中心路由等, 是十分重要的, 如何评估节点的重要性, 并为之提供适当的容错能力也是需要考虑的。因此, 本文提出的 OPTMap 算法还是有进步的空间, 需要未来进一步研究。

5.3 本章小结

在本章中, 主要说明了如何进行虚拟网络映射的实验以及实验结果。首先介绍了虚拟网络映射系统, 详细说明了各个模块的功能, 并给出了虚拟网络映射系统的工作流程。

然后, 进行了不同规模的模拟实验, 说明在不同的规模下, 本文的 OPTMap 算法都是由于经典算法的。并补充了真实拓扑的实验, 证明 OPTMap 算法的有效性。

最后对实验结果进行统筹分析, 证明在各个情况下, 前处理过程是有效的, 并且 OPTMap 算法的表现都是强于经典算法的。

结 论

随着网络空间对抗形势的日趋严峻，网络对抗已成为各个国家新一轮的对抗形式。因此利用网络靶场进行网安试验，模拟网络对抗已成为一种常态。进行网安试验就需要网络虚拟化，虚拟网络映射问题作为网络虚拟化的核心问题，在网安试验的情况下，和传统情况有很大的区别，研究解决网安试验下的虚拟网络映射问题成为本文的目的。本文主要做了以下几个方面的内容。

(1) 分析网安试验下的虚拟网络映射问题和传统虚拟网络映射问题的差异，提出了面向网安试验的大规模虚拟网络映射问题，说明其模型和形式化定义，并给出了相应的评价函数。

(2) 分析传统的虚拟网络映射算法，虚拟网络映射分为节点映射和链路映射两个过程。针对网安试验，发现传统的虚拟网络映射算法的局限性在于可解的虚拟网络请求规模太小和链路映射失败率太高。

(3) 提出了一种面向网安试验的解决大规模虚拟网络映射问题的算法。该算法首先是基于多层 K 划分的前处理过程，再进行基于节点契合度的映射过程。针对前处理过程，给出具体流程和采用的算法，并提出了精细化的 K 的计算公式。针对映射过程，给出映射的具体方案，并提出了不同以往的节点排序公式和节点契合度计算公式，针对链路映射也做出了相应的优化。

(4) 搭建了虚拟网络映射系统，将本文提出的算法 **OPTMap** 与经典虚拟网络映射算法作对比，并在传统论文模拟实验的基础之上，进行了真实拓扑实验，验证了本文提出算法的优越性。

在本文研究的基础之上，还有一些研究需要进行，主要包括：

(1) 针对网络的拓扑特点，应当更加深入的进行研究，考虑物理设备的异构性，使得网络虚拟化技术更加完善。

(2) 深入分析网安试验的特点，针对节点重要性评估，节点容错能力等问题进行研究。

参考文献

- [1] Andersen D G. Theoretical approaches to node assignment[J]. Computer Science Department, 2002: 86.
- [2] Keller E, Szefer J, Rexford J, et al. NoHype: virtualized cloud infrastructure without the virtualization[C]//ACM SIGARCH Computer Architecture News. ACM, 2010, 38(3): 350-361.
- [3] 曹勇, 吕光宏, 周飞. 基于 SDN 的网络虚拟化平台研究[J]. 通信技术, 2017, 50(9):1987-1993.
- [4] Morita I, Yoshikane N, Tsuritani T, et al. Heterogeneous Multi-domain Network Virtualization with End-to-end Differentiated Service Provisioning and Virtual Network Organization[C]// Optical Fiber Communications Conference and Exhibition. IEEE, 2015:1-3.
- [5] Kitindi E J, Fu S, Jia Y, et al. Wireless Network Virtualization with SDN and C-RAN for 5G Networks: Requirements, Opportunities, and Challenges[J]. IEEE Access, 2017, PP(99):1-1.
- [6] Sardara M, Muscariello L, Augé J, et al. Virtualized ICN (vICN): towards a unified network virtualization framework for ICN experimentation[C]// ACM Conference on Information-Centric NETWORKING. ACM, 2017:109-115.
- [7] Liu W, Xiang Y, Ma S, et al. Completing virtual network embedding all in one mathematical programming[C]// International Conference on Electronics, Communications and Control. IEEE, 2011:183-185.
- [8] Trinh T, Esaki H, Aswakul C. Quality of service using careful overbooking for optimal virtual network resource allocation[C]// International Conference on Electrical Engineering/electronics, Computer, Telecommunications and Information Technology. IEEE, 2011:296-299.
- [9] Pages A, Perello J, Spadaro S, et al. Strategies for Virtual Optical Network Allocation[J]. IEEE Communications Letters, 2012, 16(2):268-271.
- [10] Nguyen D L, Byun H H, Kim N, et al. Towards Efficient Dynamic Virtual Network Embedding Strategy for Cloud IoT Networks[J]. International Journal of Distributed Sensor Networks, 2018, 14(3).
- [11] Razzaq A, Rathore M S. An Approach towards Resource Efficient Virtual Network Embedding[C]// Second International Conference on Evolving Internet. IEEE,

2010:68-73.

- [12] Razzaq A, Sjödin P, Hidell M. Minimizing bottleneck nodes of a substrate in virtual network embedding[C]// Network of the Future. IEEE, 2011:35-40.
- [13] Nogueira J, Melo M, Carapinha J, et al. Virtual network mapping into heterogeneous substrate networks[C]// Computers and Communications. IEEE, 2011:438-444.
- [14] Zhang S, Qian Z, Wu J, et al. Virtual Network Embedding with Opportunistic Resource Sharing[J]. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(3):816-827.
- [15] Beck M T, Fischer A, Botero J F, et al. Distributed and scalable embedding of virtual networks[J]. Journal of Network & Computer Applications, 2015, 56(C):124-136.
- [16] Leivadeas A, Papagianni C, Papavassiliou S. Efficient Resource Mapping Framework over Networked Clouds via Iterated Local Search-Based Request Partitioning[J]. IEEE Transactions on Parallel & Distributed Systems, 2013, 24(6):1077-1086.
- [17] Yuan Y, Wang C, Wang C, et al. Fault Tolerant Virtual Network Embedding Algorithm Based on Redundant Backup Resource[C]// International Conference on Instrumentation. IEEE, 2014:354-357.
- [18] Shahin A A. A Collective Neurodynamic Approach to Survivable Virtual Network Embedding[J]. International Journal of Advanced Computer Science & Applications, 2018, 9(3).
- [19] Yun D, Yi Y. Virtual network embedding in wireless multihop networks[C]// International Conference on Future Internet Technologies. ACM, 2011:30-33.
- [20] Zhu Y, Ammar M H. Algorithms for assigning substrate network resources to virtual network components[C]// INFOCOM. 2006, 1200(2006): 1-12.
- [21] Yu M, Yi Y, Rexford J, et al. Rethinking virtual network embedding: substrate support for path splitting and migration[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 17-29.
- [22] Chowdhury M, Rahman M R, Boutaba R. ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping[J]. IEEE/ACM Transactions on Networking, 2012, 20(1):206-219.
- [23] Ding J, Huang T, Liu J, et al. Virtual network embedding based on real-time topological attributes[J]. Frontiers of Information Technology & Electronic Engineering, 2015, 16(2):109-118.

- [24] 刘新波, 王布宏, 杨智显,等. 一种基于拓扑势的虚拟网络映射算法[J]. 电子与信息学报, 2018.
- [25] H. Yu , Anand V, Anand V, Qiao C, et al. A Cost Efficient Design of Virtual Infrastructures with Joint Node and Link Mapping[J]. Journal of Network & Systems Management, 2012, 20(1):97-115.
- [26] 黄敏桓, 张尧学, 唐洪,等. 大规模仿真网络拓扑映射的子网分割技术[J]. 北京邮电大学学报, 2011, 34(s1):104-108.
- [27] 刘百川, 郭燕慧. 基于拓扑图划分的仿真节点映射策略研究[C]// 中国信息通信研究新进展论文集. 2014.
- [28] 彭利民. 基于图的邻接分割的虚拟网络映射算法[J]. 华南理工大学学报 (自然科学版), 2015(1):66-71.
- [29] Razzaq A, Rathore M S. An approach towards resource efficient virtual network embedding[C]//Evolving Internet (INTERNET), 2010 Second International Conference on. IEEE, 2010: 68-73.
- [30] Xu Z, Liang W, Xia Q. Efficient Embedding of Virtual Networks to Distributed Clouds Via Exploring Periodic Resource Demands[J]. IEEE Transactions on Cloud Computing, 2016, PP(99):1-1.
- [31] Haeri S, Trajković L. Virtual Network Embedding via Monte Carlo Tree Search[J]. IEEE Transactions on Cybernetics, 2017, PP(99):1-12.
- [32] Rost M, Schmid S. Virtual Network Embedding Approximations: Leveraging Randomized Rounding[J]. 2018.
- [33] 李秋香, 郝文江, 李翠翠,等. 国外网络靶场技术现状及启示[J]. 信息安全, 2014(9):63-68.
- [34] 郭艳来, 崔益民, 唐洪,等. 大规模网络的拓扑分割技术[J]. 计算机工程与设计, 2015(11):2921-2925.
- [35] Karypis G, Kumar V. A Parallel Algorithm for Multilevel Graph Partitioning and Sparse Matrix Ordering[J]. Journal of Parallel & Distributed Computing, 1998, 48(1):71-95.
- [36] George Karypis, Vipin Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs[J]. Siam Journal on Scientific Computing, 1998, 20(1):359--392.
- [37] Karypis G, Kumar V. Analysis of multilevel graph partitioning[C]// ACM/IEEE Conference on Supercomputing. ACM, 1995:29.

- [38] 霍红卫, 许进. 快速排序算法研究[J]. 微电子学与计算机, 2002, 19(6):6-9.
- [39] Dijkstra, E. W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1):269-271.
- [40] Zegura E W, Calvert K L, Bhattacharjee S. How to model an internetwork[C]//INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE. IEEE, 1996, 2: 594-602.
- [41] Medina A, Lakhina A, Matta I, et al. BRITE: An approach to universal topology generation[C]//Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on. IEEE, 2001: 346-353.
- [42] 吴元立, 司光亚, 罗批. 多约束条件下互联网骨干网路由器级拓扑生成方法[J]. 国防科技大学学报, 2016, 38(3):167-173.
- [43] 李周, 许红升, 叶彬,等. 电力通信网结构优化及拓扑生成算法[J]. 电气自动化, 2017, 39(5):20-23.
- [44] 韩青, 兰巨龙, 江逸茗. 一种保证 NFV 可靠性的最优备份拓扑生成方法[J]. 计算机应用研究, 2017, 34(11):3384-3387.

攻读硕士学位期间发表的论文及其它成果

（一）发表的学术论文

- [1] Ao Sun, Weizhe Zhang. Performance Evaluation of Communication Intensive Application in Virtual Computing Environment. International Symposium on Information Processing, 2018. (录用待发表)

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《面向网安试验的大规模虚拟网络映射算法研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：孙奥

日期：2018年6月24日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1) 学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2) 学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3) 研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：孙奥

日期：2018年6月24日

导师签名：朱伟

日期：2018年6月24日

致 谢

眨眼间在哈尔滨工业大学这片知识沃土上的学习生活即将走到终点，这六年时间，在身边各种各样的亲朋好友的帮助下我完成了一个认识自己，改变自己以及提升自己的过程。感谢两年间帮助过我的人们，在这里对你们表示衷心的感谢，并送上诚挚的祝福。

首先感谢网安实验室为我提供的平台，网安实验室良好的学习氛围使得我对学习有了更浓厚的兴趣。在实验室的两年我获益匪浅，接触到了很多厉害的老师，和优秀的同学，在巨人的肩膀上我看到了更广阔的世界。

在这里衷心感谢我的导师张伟哲教授，张老师无时无刻都在用严谨的治学态度教育着我们，使我时时刻刻以老师为榜样。在研究生学习中，张老师不厌其烦地对我进行悉心的指导，给予了我莫大的帮助和支持，无数次在我迷茫彷徨的时候，张老师一次次给了我很多宝贵的指导意见，为我指明了方向。生活中张老师是一位和蔼的长辈，对每一个同学都十分关怀。感谢张老师两年来对我的栽培，让我对知识有了更深层次的理解和感悟，是我一辈子宝贵的财富。

感谢张宏莉老师、何慧老师、翟健宏老师、余翔湛老师、张宇老师以及实验室中的其他各位老师，正因为有老师们的指点与帮助，在实验室的两年才更加充实，更加丰富多彩。

感谢王德胜师兄、郝萌师兄、白恩慈师兄、鲁刚钊师兄、李雄师兄，在科研上给了我许多帮助。感谢我的室友刘国强、张晔、冯俊龙，以及实验室的胡尧、郝科委、李星晨同学，两年来一起度过的学习生活十分充实，研究生期间和这样优秀的同学们一起度过给我留下了一段美好的回忆，我会永远怀念这样大家一起学习一起进步的日子。祝愿大家在今后的学习和生活中能一切顺利。

最后感谢我的父母，感谢他们对我一直以来的包容和无条件的爱，他们的鼓励和不放弃让我能够努力坚持走下去，感谢我的女朋友，谢谢你的每一份帮助。