

分类号：
密 级：(不填)

单位代码：10389
学 号：(S/B)



福建农林大学

硕士学位论文

面向 5G 网络服务功能链的迁移机制研究

学 科 门 类：工学

一级学科名称：计算机科学与技术

二级学科名称：

研 究 方 向：云计算与大数据

研 究 生 姓 名：冯浩贤

指 导 教 师：舒兆港

完 成 时 间：二〇二三年四月

Research on migration mechanism of 5G Service Function Chaining

By

Haoxian Feng

Supervised by Associate Prof. Zhaogang Shu

A Dissertation Submitted to

**Fujian Agriculture and Forestry University
in Partial Fulfillment of the Requirements
for**

Master Degree of Engineering

**College of Computer and Information Science
Fujian Agriculture and Forestry University
Fujian, P.R. China**

Completion Date (April, 2023)

Commencement Date (April, 2023)

独创性声明

本人声明，所呈交的学位（毕业）论文，是本人在指导教师的指导下独立完成的研究成果，并且是自己撰写的。尽我所知，除了文中作了标注和致谢中已作了答谢的地方外，论文中不包含其他人发表或撰写过的研究成果。与我一同对本研究做出贡献的同志，都在论文中作了明确的说明并表示了谢意，如被查有侵犯他人知识产权的行为，由本人承担应有的责任。

学位（毕业）论文作者亲笔签名：

日期：

论文使用授权的说明

本人完全了解福建农林大学有关保留、使用学位（毕业）论文的规定，即学校有权送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

保密，在 年后解密可适用本授权书。

不保密，本论文属于不保密。

学位（毕业）论文作者亲笔签名：

日期：

指导教师亲笔签名：

日期：

目 录

目 录.....	I
图 录.....	IV
表 录.....	VI
摘 要.....	VII
ABSTRACT.....	IX
1 绪论.....	1
1.1 研究背景与意义.....	1
1.2 研究现状.....	2
1.2.1 面向 5G SFC 资源分配的研究现状.....	2
1.2.2 面向 5G SFC 迁移的研究现状.....	3
1.3 研究内容.....	4
1.4 论文章节安排.....	6
2 技术背景和理论基础.....	7
2.1 网络服务功能链技术.....	7
2.1.1 网络功能虚拟化.....	8
2.1.2 软件定义网络.....	10
2.2 组合多摇臂赌博机模型.....	10
2.2.1 超臂.....	11
2.2.2 奖赏.....	11
2.3 时间序列预测技术.....	12
2.3.1 差分整合移动平均自回归模型.....	12
2.3.2 长短时记忆神经网络.....	13
2.3.3 Prophet.....	14
2.4 本章小结.....	15
3 基于 CMAB 的关键业务流识别策略.....	16
3.1 引言.....	16
3.2 问题陈述.....	17
3.2.1 问题定义.....	18
3.2.2 模型修订.....	19
3.3 问题求解.....	20
3.3.1 贪婪摇臂选取算法.....	20
3.3.2 摇臂选取依据.....	21

3.3.3 复杂度分析.....	22
3.4 仿真与性能分析.....	22
3.4.1 数据集.....	23
3.4.2 对比算法介绍.....	23
3.4.3 评价指标.....	24
3.4.4 仿真结果及分析.....	24
3.5 本章小结.....	30
4 面向关键 SFC 的迁移策略	31
4.1 引言.....	31
4.2 问题陈述.....	33
4.2.1 物理网络模型.....	34
4.2.2 SFC 请求模型	35
4.2.3 问题定义.....	35
4.3 问题求解.....	37
4.3.1 一种保守的迁移策略.....	37
4.3.2 一种激进的迁移策略.....	43
4.4 仿真与性能分析.....	46
4.4.1 实验设置.....	46
4.4.2 后续 SFC 请求的资源分配算法	48
4.4.3 评价指标.....	48
4.4.4 仿真结果及分析.....	49
4.5 本章小结.....	56
5 基于资源预测机制的关键 SFC 迁移优化策略	57
5.1 引言.....	57
5.2 问题定义.....	57
5.3 问题求解.....	58
5.3.1 数据处理.....	58
5.3.2 预测方法.....	59
5.4 仿真与性能分析.....	59
5.4.1 数据集.....	59
5.4.2 评价指标.....	60
5.4.3 仿真结果及分析.....	60
5.5 本章小结.....	66
6 总结与展望.....	67
6.1 主要工作总结.....	67
6.2 创新点.....	67

6.3 不足与展望.....	68
参考文献.....	70
攻读学位期间的学术论文与研究成果.....	75
致 谢.....	76

图 录

图 1-1 由各因素造成物理资源分布的不均衡	5
图 2-1 在物理网络中部署 SFC 请求 ^[16]	8
图 2-2 NFV 架构	9
图 2-3 SDN 架构	10
图 2-4 LSTM 结构.....	13
图 3-1 基于 KDN 的 top- K 流识别	17
图 3-2 动态识别网络中的 top- K 流.....	19
图 3-3 随机选择算法与 mean-greedy 策略在 Geant 网络中的性能对比.....	25
图 3-4 随机选择算法与 mean-greedy 策略在 Abilene 网络中的性能对比.....	25
图 3-5 各摇臂选取策略在 Geant 网络中的性能对比.....	26
图 3-6 各摇臂选取策略在 Abilene 网络中的性能对比.....	26
图 3-7 不同权重值在 Geant 网络中的性能对比.....	27
图 3-8 不同权重值在 Abilene 网络中的性能对比.....	27
图 3-9 不同滑动窗口值在 Geant 网络中的性能对比.....	27
图 3-10 不同滑动窗口值在 Abilene 网络中的性能对比.....	28
图 3-11 各算法在 Geant 网络中的性能对比.....	28
图 3-12 各算法在 Abilene 网络中的性能对比.....	29
图 4-1 SFC 迁移说明图	33
图 4-2 保守 SFC 迁移策略说明图	42
图 4-3 激进 SFC 迁移策略说明图	44
图 4-4 SFC 迁移仿真实验拓扑	46
图 4-5 单个实验实例的物理资源状态	49
图 4-6 20 个实验实例的物理资源状态	50
图 4-7 单个实验实例的迁移开销	50
图 4-8 20 个实验实例的迁移开销	50
图 4-9 单个实验实例中服务队列中 SFC 请求个数的变化	52
图 4-10 单个实验实例中 SFC 请求的接收率的变化	52
图 4-11 单个实验实例的链路资源利用率的变化	52
图 4-12 单个实验实例的节点资源利用率的变化	53
图 4-13 单个实验实例的网络运营商的长期收益的变化	53
图 4-14 20 个实验实例在 $T = 60$ 时的 SFC 请求接收率	54
图 4-15 20 个实验实例在 $T = 60$ 时的链路资源利用率	55

图 4-16 20 个实验实例在 $T = 60$ 时的节点资源利用率	55
图 4-17 20 个实验实例在 $T = 60$ 时的长期收益	55
图 5-1 SFC 请求 i 的流量走势图	58
图 5-2 单个 SFC 请求在 2021 年 8 月 2 日与其他天数之间的皮尔森系数	61
图 5-3 单个 SFC 请求在 2021 年 8 月 2 日与其他天数之间的皮尔森系数（移除 周末的流量数据）	61
图 5-4 数据的 ACF	62
图 5-5 数据的 PACF	62
图 5-6 不同预测方法的预测结果	63
图 5-7 结合资源预测机制后单个实验实例中 SFC 请求的接收率的变化	64
图 5-8 结合资源预测机制后单个实验实例的网络运营商的长期收益的变化	64
图 5-9 结合资源预测机制后 20 个实验实例在 $T = 60$ 时的 SFC 请求的接收率 ..	65
图 5-10 结合资源预测机制后 20 个实验实例在 $T = 60$ 时的长期收益	65

表 录

表 3-1 本文工作与现有 top- K 摇臂识别工作的对比分析.....	17
表 3-2 top- K 流识别问题所使用的符号总结.....	18
表 3-3 本文所提的贪婪摇臂选取算法的复杂度分析	22
表 3-4 top- K 流识别算法仿真拓扑参数.....	23
表 3-5 贪婪摇臂选取算法与其他方法的对比分析	29
表 4-1 本文工作与现有面向 SFC 迁移工作的对比分析	32
表 4-2 SFC 迁移问题所使用的符号总结	34
表 4-3 SFC 迁移仿真实验拓扑参数	47
表 4-4 SFC 迁移仿真实验中各类型 SFC 请求的参数	47
表 4-5 20 个实验实例的平均 std 及迁移开销.....	49
表 4-6 20 个实验实例的平均结果及增幅	56
表 5-1 不同方法在单个 SFC 请求的预测结果的对比分析	63
表 5-2 Prophet 在 6 个 SFC 请求中的预测结果分析.....	64
表 5-3 结合预测机制后 20 个实验实例的平均结果及增幅	66

摘要

随着科技的迅速发展,各种新型的网络垂直业务层出不穷。然而,传统的移动通信网络采取“一刀切”的方式来为各种业务提供服务,因而忽略了业务在垂直方向上的不同需求。网络服务功能链 (Service Function chaining, SFC) 作为 5G 网络的关键技术,不仅可以灵活地应对不同的应用场景,而且能为用户提供定制化的服务。但是,由于 SFC 请求的生命周期及资源调整等因素,物理网络中的可分配资源会变得不均衡,这会对后续 SFC 请求以及运营商带来潜在的负面影响,如降低后续服务请求的接收率、减少物理资源的利用率等。并且网络中超过 80% 的流量是由占比 10% 的 top- K 关键业务流产生的。为提高后续服务请求的接收率和物理资源的利用率,运营商需要针对 5G 网络中关键的 SFC 请求进行迁移。针对以上问题,本文拟先研究如何有效地识别出 5G 网络中的 top- K 关键业务流,再研究关键 SFC 请求的迁移策略。

对于 5G 网络中 top- K 关键业务流的识别问题,本文首先利用组合多摇臂赌博机 (Combinatorial Multi-Armed Bandit, CMAB) 模型来对问题进行建模,然后根据新兴网络技术(如软件定义网络技术,网络功能虚拟化技术等)的特性对 CMAB 模型进行了修订。最后,本文设计了一种基于贪婪思想的摇臂选取算法来动态识别网络中的 top- K 关键业务流。基于真实网络流量数据的仿真实验表明,在 Abilene 拓扑中,本文设计算法在识别性能上比 2 个现有关键摇臂识别策略分别高 65.42%, 32.05%。而在 Geant 网络中,上述数字则变化为 40.48%, 8.43%。

对于关键 SFC 迁移问题,本文在综合考虑 SFC 请求在迁移时的约束条件、物理资源的分布及迁移开销的情况下,利用整数线性规划模型来对问题进行建模。然后,本文提出了两种不同的迁移策略,保守的迁移策略和激进的迁移策略来完成 SFC 的迁移任务。与保守的迁移策略相比,激进的迁移策略侧重于物理资源分布的均衡性,因此会造成更多的迁移开销。最后,仿真实验表明,利用保守的策略进行

SFC 迁移后,使用两种不同的策略对后续 SFC 请求进行资源分配时,请求的接收率分别平均提高了 1.81%和 2.54%,运营商的长期收益分别平均提高了 3.05%和 4.62%。而利用激进策略进行 SFC 迁移后,上述数字则分别变化为 11.75%, 6.84%, 9.49%和 7.14%。

此外,由于在许多现实应用场景中,网络流量的变化会有明显的昼夜现象,即业务在白天的资源需求会明显高于其在夜晚的资源需求。针对这一现象,本文基于先进的时间序列预测技术对 SFC 迁移策略进行了优化。仿真实验表明,在结合资源预测机制后,利用两种不同的策略对后续 SFC 请求进行资源分配时,对于保守的迁移策略而言,请求的接收率分别进一步提高了 16.66%和 11.02%,运营商的长期收益分别进一步提高了 14.33%和 11.37%。而对于激进的迁移策略而言,上述数字则分别变化为 9.28%, 8.54%, 8.52%和 9.57%。

关键词: 组合多摇臂赌博机, top- K 流识别, 网络服务功能链, SFC 迁移, 时间序列预测

ABSTRACT

As technology evolves, a range of new vertical use cases emerges. But traditional mobile networks employ a one-size-fits-all approach to providing services, regardless of the diverging requirements of vertical services. Service function chaining (SFC) is regarded as an important technology for the 5th generation communication networks (5G) not only because it can flexibly tackle diverse usage scenarios, but also can provide users with customized services. However, due to SFC requests' lifecycle and resource adjustment, the distribution of the remaining physical resources may become unbalanced, which brings potential negative effects to subsequent SFC requests as well as network operators, **such as reducing the acceptance ratio of subsequent services, reducing utilization of physical resources, etc.** And more than 80% of network traffic is generated by less than 10% of top- K flows. **To improve the acceptance ratio of subsequent services and the utilization of physical resources**, operators need to migrate key SFC requests in the 5G network. **Aiming at the above problems, this paper intends to first study how to effectively identify the top- K flows in 5G networks, and then study the problem of SFC migration.**

Aiming at the identification of top- K flows in 5G networks, this paper first formulates the problem as a Combinatorial Multi-Armed Bandit (CMAB) model and then revises the CMAB model according to the characteristics of emerging network technologies (such as software defined network, network function virtualization, etc). Finally, this paper designs an effective algorithm based on greedy thoughts to dynamically identify the top- K flows in the network. **Simulation experiments based on real network traffic data show that in the Abilene topology, the identification performance of the algorithm designed in this paper is 65.42%, 32.05% higher than two existing top- K arms identification strategies, respectively. In the Geant network, the above numbers vary to 40.48%, and 8.43%, respectively.**

For the SFC migration problem, this paper uses the integer linear programming model to formulate the problem under the condition of comprehensively considering the constraints of the SFC request during

migration, the distribution of physical resources, and the migration cost. Then, this paper proposes two different migration strategies, a conservative migration strategy and an aggressive migration strategy, to accomplish the migration task of SFC. Compared with the conservative migration strategy, the aggressive migration strategy focuses on the balance of physical resource distribution, thus causing more migration cost. **Finally, simulation experiments show that after SFC migration using a conservative strategy, when two different strategies are used to allocate resources to subsequent SFC requests, the request acceptance ratio increases by an average of 1.81% and 2.54%, respectively, and the long-term profit of operators increases by an average of 3.05% and 4.62%, respectively. After using the aggressive strategy for SFC migration, the above numbers change to 11.75%, 6.84%, 9.49%, and 7.14%, respectively.**

In addition, in many real application scenarios, the change of network traffic will have obvious diurnal phenomenon, that is, the resource requirements of the services during the day will be significantly higher than that at night. To tackle this phenomenon, this paper optimizes the SFC migration strategy based on advanced time series prediction techniques. **The simulation experiments show that when two different strategies are used to allocate resources to subsequent SFC requests in combination with the resource prediction mechanism, the acceptance ratio of requests for the conservative migration strategy is further increased by 16.66% and 11.02% respectively, and the long-term profit of operators is further increased by 14.33% and 11.37% respectively. For the aggressive migration strategy, the above numbers change to 9.28%, 8.54%, 8.52%, and 9.57% respectively.**

KEY WORDS: Combinatorial Multi-Armed Bandit, top- K flows identification, service function chaining, SFC migration, time series prediction

1 绪论

1.1 研究背景与意义

近年来,随着技术的迅速发展,涌现出一系列新型的网络垂直业务^[1],如在线教育,虚拟现实以及智能制造。然而,现有的移动通讯网络架构忽略了业务垂直需求间的差异性并采取“一刀切”的方式来为各种业务提供服务,从而导致了业务的灵活性低,可伸缩性不足以及资源浪费等问题^[2]。此外,一些传统的网络功能服务(如防火墙,负载均衡器,深度包检测等)只能部署在特定的物理设备上。因此,网络运营商需要花费大量的金钱来维护旧的设备以及购买新的设备来满足不断增长的用户需求。为了降低网络运营商的运营费用和提高用户的体验质量,对传统网络架构进行革新是历史发展的必然趋势。

软件定义网络 (Software Defined Network, SDN)^[3] 和网络功能虚拟化 (Network Function Virtualization, NFV)^[4] 被视为解决上述不足的关键技术。SDN 通过分离传统网络的控制平面和转发平面,不仅能灵活高效地对网络进行管理,而且能有效地降低网络的运维成本。NFV 利用虚拟化技术将特定的网络功能从专用设备分离到通用的硬件设备上,因此,虚拟网络功能 (Virtual Network Functions, VNFs)^[5] 可以被灵活地部署在合适的网络位置上,从而为用户提供服务。此外,随着 NFV 技术的快速发展,VNFs 不仅可以在各种流量条件下进行动态的收缩和扩张^[6,7]以保障用户的体验质量 (Quality of Experience, QoE),还可以进行集群部署以克服单点故障^[8]的缺点。

5G 网络根据业务的应用场景以及安全需求把业务分成三种类型,分别是移动增强带宽 (enhanced Mobile Broad-Band, eMBB),低时延高可靠通信 (Ultra-Reliable and Low-Latency Communications, URLLC) 以及大规模物联网通信 (Massive Machine-Type Communications, mMTC)。为满足信息时代下多样化的业务需求,互联网工程任务组 (Internet Engineering Task Force, IETF) 基于 SDN 和 NFV 技术,提出了网络服务功能链 (Service Function Chaining, SFC)^[10] 的概念。通常,一个网络服务功能链请求包含一组 VNFs,并且该 SFC 请求产生的流量需要以特定的顺序来通过这些 VNFs^[11,12]。运营商可以在一个具备 SDN 和 NFV 功能的物理网络中灵活地部署 SFC,以提高物理资源的利用率,降低运维成本以及实现业务间的逻辑独立。当前,SFC 已经成为一个研究热点^[15,16]。

当网络的管理和编排 (Management and Orchestration, MANO) 层接收到一个 SFC 请求时,它需要为该请求进行资源配置。此外,在 SFC 运行的过程中,MANO 还需要根据业务资源需求的波动性来动态调整已分配的资源。最后,当 SFC 请求的生命周期结束后,MANO 要负责对资源进行回收。然而,上述过程

可能会造成剩余物理资源分布的不均衡，从而给后续的 SFC 请求和运营商带来负面影响（如，更低的服务接收率、更低的物理资源利用率，等）。

研究面向 5G 网络的 SFC 迁移机制可以有效地消除潜在的负面影响，从而进一步地保障用户的 QoE，提高物理资源的利用率，加快 SFC 技术的落地应用。因此，本研究具有良好的学术价值和应用前景。

1.2 研究现状

由于 SFC 具有部署灵活，业务定制化，维护成本低等优点，近年来，它已经成为了学术界和工业界的一个研究热点^[13-16]。为了加速 SFC 技术的落地应用，国内外学者围绕 SFC 资源分配及其迁移展开了一系列的研究工作。

1.2.1 面向 5G SFC 资源分配的研究现状

SFC 资源分配主要涉及 VNF 的放置和路由的选择问题，文献[15-18]分别基于不同的理论方法提出了有效的资源分配方案。文献[15]先利用马尔可夫决策模型来定义连续性的 SFC 资源分配问题，然后，为了最小化网络运营商的运营成本以及最大化 SFC 业务吞吐量，该文献的作者利用了深度强化学习模型来对问题进行求解。此外，为了解决离散动作空间过大以及训练速度慢等问题，研究人员还结合了序列化，回溯以及梯度下降等方法来对模型进行了优化。而文献[16]则把 VNF 放置和路由选择问题定义成一个二进制整数规划问题，并提出了两个神经网络来对问题进行求解。其中，第一个神经网络用于选择 VNF 实例，而第二个神经网络用于实现 VNF 实例间的串联。除此之外，研究人员还预定义了 VNF 实例的可行动作集和路径的可行动作集以减少动作空间的规模。实验结果表明，通过两个神经网络的协作，运营商可以有效地降低 SFC 业务的总时延。与结合人工智能方法^[15,16]的资源分配方法不同，部分研究人员^[17,18]通过传统的优化方法来寻求 SFC 的资源分配方案。由于 VNF 的放置需要考虑到多个优化目标，如降低总带宽消耗、最大化链路资源利用率等，文献[17]基于多目标遗传算法和改进的非支配排序遗传算法框架，提出了四种不同的 VNF 放置策略。文献[18]综合考虑了 VNF 放置，资源分配的公平性以及流量优化三个因素，通过排队论来表征 5G 网络中垂直业务的资源分配问题。此外，研究人员基于 Karush-Kuhn-Tucker 条件对问题进行了简化，并结合现有的凸优化求解器来构造出一个高效且快速的求解方法。

在 5G 网络切片场景^[19]中，切片需求可以由 SFC 来表征，因此，部分科研工作者^[20-23]展开了面向 5G 网络切片资源分配的研究。文献[20]提出了基于复杂网络理论的网络切片资源分配方案。研究人员首先基于物理节点的计算资源、邻接

链路的带宽资源、节点的度以及中介中心性信息来表征每个物理节点的重要程度, 然后按降序原则来放置网络切片中的 VNFs, 最后使用弗洛伊德算法把 VNFs 进行串联。实验结果表明, 在无标度网络中, 该解决方案能够不仅有效地降低计算成本, 而且能够提高物理资源的利用率。文献[21]利用混合整数线性规划模型来对网络切片资源分配问题进行建模, 随后, 研究人员提出了有效惩罚连续上界最小化方法以及两个启发式算法来对问题进行求解。同样地, 文献[22]也利用混合整数线性规划模型来对网络切片资源分配问题进行建模, 并提出了一种简洁有效的启发式算法来对问题进行求解。不同的是, 研究人员在分配资源时考虑到了 VNF 的性能衰减问题, 在此前提下, 他们通过性能补偿来保障用户的 QoE。文献[23]提出了一种新型的框架来为网络切片请求进行资源分配, 研究人员所提的框架把网络切分成了无线接入网, 传输网和核心网三个部分, 然后他们设计了不同的算法来为不同类型的网络切片进行资源分配, 以满足各用户的服务需求。

1.2.2 面向 5G SFC 迁移的研究现状

目前, 大部分 SFC 迁移工作的目的都是保障用户的 QoE^[24-31]。在一些垂直的业务场景中, 如, 自动驾驶, 用户终端的位置会随着用户的移动而变化, 文献[24]介绍了跟随云 (Follow-Me Cloud, FMC) 的概念并对其框架进行了阐述。随后, 研究人员利用马尔可夫决策过程来对业务的迁移进行求解。文献[25]在自动驾驶的应用场景中应用了 FMC 解决方案, 并提出在对 SFC 中的 VNFs 进行迁移时, 需要尽可能地减少迁移所需要的时间, 以确保用户的 QoE 不受影响。文献[26,27]展开了面向低时间成本的 VNFs 迁移研究。对于依赖同步的应用程序, 文献[26]的研究人员设计了四种 SFC 迁移模式并对每一个模式都进行了性能评估。此外, 研究人员还设计了一种基于共享文件系统^[27]的方法来减少迁移的时间成本。

尽管不少科研人员都研究了如何对 SFC 进行迁移以满足业务的移动特性, 但是他们最终的优化目标是不同的。在核心云和边缘云的混合环境中, 文献[28]利用整数线性规划模型来对 SFC 迁移问题进行建模, 然后利用 Gurobi 优化器来进行求解。由于研究人员专注于如何降低运营商的运维成本以及节约物理资源, 因此, 他们忽略了迁移成本。此外, 由于 Gurobi 优化器高昂的计算成本, 该解决方案在大规模网络中的可行性较低。文献[29]在雾计算的背景下研究了由终端用户移动性引发的 SFC 迁移问题。同样地, 研究人员也是利用了整数线性规划模型来对问题进行建模, 并提出了两种高效的启发式算法来对问题进行求解。考虑到迁移的开销, 研究人员仅对 SFC 中的部分 VNFs 进行迁移, 并且他们忽略了 SFC 的生命周期管理以及剩余物理资源的分布情况。

部分研究人员^[30,31]结合了人工智能技术来展开 SFC 和网络切片的迁移工作。文献[30]设计了两种基于深度强化学习的 SFC 迁移方案来对带宽资源进行重新

分配。然而，他们提出的模型仅能确定给定 workflow 所需的带宽而缺乏泛化能力，且他们的研究工作并没有考虑迁移成本。文献[31]在核心云环境中探索了由用户需求的波动造成的网络切片资源重分配问题。研究人员利用马尔可夫决策模型来对问题进行建模，然后提出了一个离散的分支决斗 Q 网络来对问题进行求解。尽管他们的方法考虑到了迁移成本，但是他们设计的网络的动作集是提前设定好的，因而并不具备良好的泛化能力。上述基于深度强化学习的解决方案均存在两个不足，一是动作集的设计缺陷，对于不同的应用场景（如不同的拓扑，不同类型的 SFC、网络切片等），研究人员需要重新设计动作集。二是泛化能力弱，训练好的模型在迁移到其他类型的拓扑时需要耗费大量的资源进行重新训练^[32]。

也有少部分研究是基于资源分布不均衡展开的^[33]。为了消除物理节点出现过载情况时对运营商造成的负面影响，文献[33]提出了一种懒惰的迁移策略。然而，为了尽可能地减少迁移开销，该策略仅仅对切片的最后一个 VNF 进行迁移。虽然这种解决方案可以在一定程度上减轻高负荷节点的负担，但是没有从全局的角度来考虑资源分布的均衡性。

1.3 研究内容

本文针对 5G 网络中，由 SFC 请求的资源变化，生命周期等因素所造成物理资源分布不均衡问题，研究面向 5G 网络服务功能链的迁移机制。首先，本文对面向 5G 网络服务功能链的资源分配及迁移机制的相关研究工作进行了分析与调研。然后，考虑到网络中不同业务资源需求的差异性，本文展开网络中关键业务流动态识别的相关研究。接着，本文综合考虑 SFC 请求在迁移时的约束条件、物理资源的均衡性及迁移开销等因素，研究具有高可用性的 SFC 迁移策略。最后，在此基础上，考虑到各业务资源需求的时变性，本文结合时间序列预测的相关技术，对 SFC 迁移策略进行优化。

因此，本文的具体研究内容如下：

(1) 网络中关键业务流的动态识别

尽管物理网络上承载着非常多的业务流，然而网络中的大部分流量仅分布在小部分（约 10%）的关键业务流中^[34]。在对 SFC 进行迁移前，为了有效地减少问题的解空间，动态地识别网络中的关键业务流是非常重要的。若把每一条流看作一个摇臂，每条流在单位采样间隔内的带宽需求看作该摇臂在单个回合内产生的随机奖赏，则可以使用组合多摇臂赌博机 (Combinatorial Multi-Armed Bandit, CMAB) 模型来对网络中关键业务流的动态识别问题进行建模。同时，本文拟综合考虑新兴网络技术的发展（结合 SDN 的灵活性及全局性等特点），对 CMAB 模型进行修订。接着，本文拟基于贪婪思想来设计摇臂选取策略。最后，本文拟搜集两个真实网络 (Abilene, Geant)^[35] 的流量数据来展开仿真实验。

(2) 面向关键 SFC 的迁移策略

本文的第二个研究内容为面向关键 SFC 迁移策略的研究。考虑到 SFC 请求之间的差异性（如，生命周期的不同、资源需求的差异等）及 MANO 对 SFC 请求的重配置（如对已分配资源的扩张或收缩），网络中剩余物理资源的分布可能会出现不均衡的情况，如图 1-1 所示。此时，若 MANO 收到新的 SFC 请求，MANO 很有可能会受到物理网络中某些高负荷节点或链路的影响而拒绝接收新的 SFC 请求，从而给后续的用户，运营商造成负面影响（如，更低的物理资源利用率，更低的长期收益，等）。为了消除或降低这些负面影响，MANO 需要对服务队列中的关键 SFC 请求进行迁移。本文拟在综合考虑 SFC 请求在迁移时的约束条件、迁移开销和物理资源分布的前提下，利用整数线性规划模型 (Integer Linear Programming, ILP) 对 SFC 迁移问题进行建模，并设计 SFC 迁移的求解方法。最后，本文需要通过仿真实验来验证 SFC 迁移的必要性。

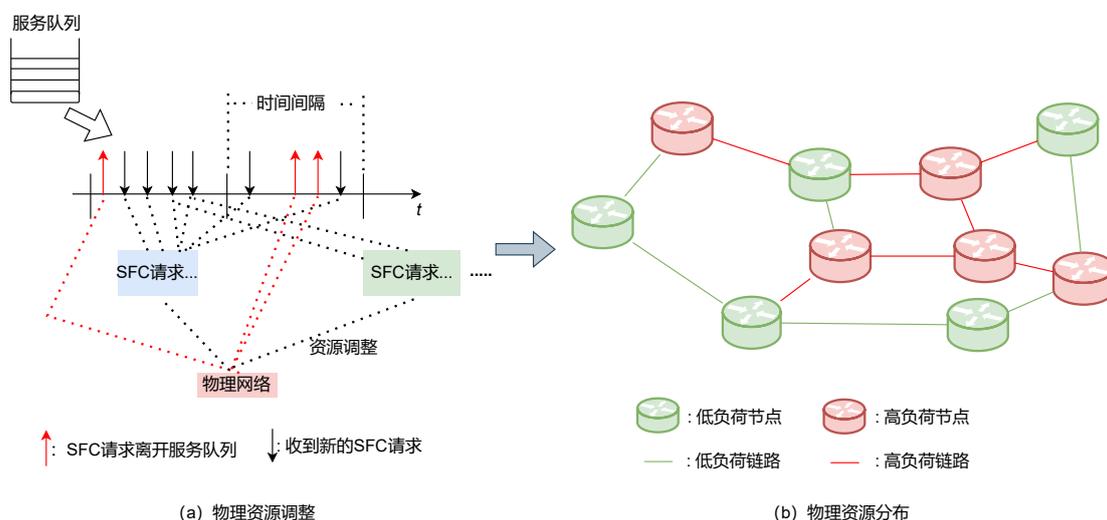


图 1-1 由各因素造成物理资源分布的不均衡

(3) 基于资源预测的关键 SFC 迁移优化策略

考虑到在现实的应用场景中，大部分业务的流量分布具有明显的昼夜现象^[36]，即 SFC 请求在白天的资源需求要高于其在黑夜的资源需求。因此，本文的最后一个研究内容为基于资源预测机制对面向关键 SFC 的迁移策略进行优化。具体为，在资源需求的收缩期，本文拟通过各时间序列预测方法来预测出关键 SFC 请求在后续一段时间内的资源需求，并根据预测的结果来回收部分已分配的资源，以此来进一步地提高运营商的长期收益。为验证预测机制的有效性，本文拟基于 WIDE 项目组^[37]提供的真实网络数据来展开仿真实验。

1.4 论文章节安排

本文共 6 章，每个章节的具体结构安排如下：

第 1 章：绪论。本章首先介绍了面向 5G 网络服务功能链迁移机制的研究背景和意义，以及总结了目前面向 5G 网络服务功能链的资源分配以及迁移的研究现状，接着阐述了本文的主要研究内容，最后介绍了本文的章节安排。

第 2 章：技术背景和理论基础。本章首先介绍了本文涉及的网络服务功能链、网络功能虚拟化以及软件定义网络等技术背景，然后介绍了组合多摇臂赌博机模型和时间序列预测技术等理论基础。

第 3 章：基于 CMAB 的关键业务流识别策略。本章阐述了本文所展开的基于组合多摇臂赌博机模型的关键业务流识别的研究工作，首先给出了问题的背景、定义以及对模型的修订，然后介绍了本文所提出的贪婪摇臂选取算法，最后通过基于真实数据的仿真实验来验证本文所提算法的有效性。

第 4 章：面向关键 SFC 的迁移策略。本章阐述了本文所展开的面向关键 SFC 的迁移策略的研究工作，首先给出了迁移问题的背景和定义，然后详细说明了本文所设计的两种 SFC 迁移策略，最后通过仿真实验来证明本文所设计的迁移策略的有效性；

第 5 章：基于资源预测机制的关键 SFC 迁移优化策略。本章针对现实应用场景中业务资源需求的昼夜现象，对本文第 4 章所提的迁移策略进行优化。本章首先介绍了问题的背景和定义，然后简单介绍了本章所使用的 3 种预测技术，最后通过搜集真实网络流量数据展开仿真实验，以验证资源预测机制的有效性。

第 6 章：总结与展望。本章对全文的主要工作和创新点进行了总结，然后分析出本研究的一些不足之处，最后对后续的研究工作进行展望。

2 技术背景和理论基础

本章介绍本文所涉及的专业知识。首先，本章介绍了网络服务功能链技术及其涉及的重要网络技术，然后介绍了组合多摇臂赌博机理论模型。最后，本章介绍了 3 种时间序列预测技术。

2.1 网络服务功能链技术

软件定义网络 (Software Defined Network, SDN) 和网络功能虚拟化 (Network Function Virtualization, NFV) 等新兴网络技术的快速发展催生了网络服务功能链 (Service Function Chaining, SFC) 技术。SFC 作为 5G 网络中的关键技术，通过 NFV 技术把用户需要的各种网络功能以虚拟机的方式部署在网络中的各个位置。通过这种方式，网络运营商不仅可以有效地降低网络的部署成本和运维成本（无需再购买昂贵的专用物理设备），还能灵活为用户提供各种业务，并且业务的性能、安全性和可管理性都得到了提高^[38,39]。此外，SFC 通过 SDN 技术实现虚拟机之间路径的动态计算，从而满足业务在网络带宽，时延方面的需求。

一般地，一个 SFC 请求 (SFC Request, SFCR) 可以被表征为虚拟节点和虚拟链路的集合，而 SFCR 上的流量需要以预先设定好的顺序依次经过这些虚拟节点。网络运营商在接收到新的 SFCR 时，需要在满足各种约束（如时延约束，带宽约束，资源上限约束，等）的前提下完成虚拟节点和虚拟链路的映射。如图 2-1 所示，在支持 SDN/NFV 技术的物理网络中，一共有 5 种类型的 VNF，且每种类型的 VNF 都有多个实例放置于网络中的不同位置。其中，分别表示 VNF_a 的第 1, 2, 3 个实例。一个 SFCR 的流量从目的 A 节点开始出发，在到达目的节点 J 之前，需要依次经过 $VNF_a, VNF_c, VNF_b, VNF_d$ 。针对该 SFCR，MANO 有多种资源映射方案，如图 2-1 中不同颜色的虚线所示。

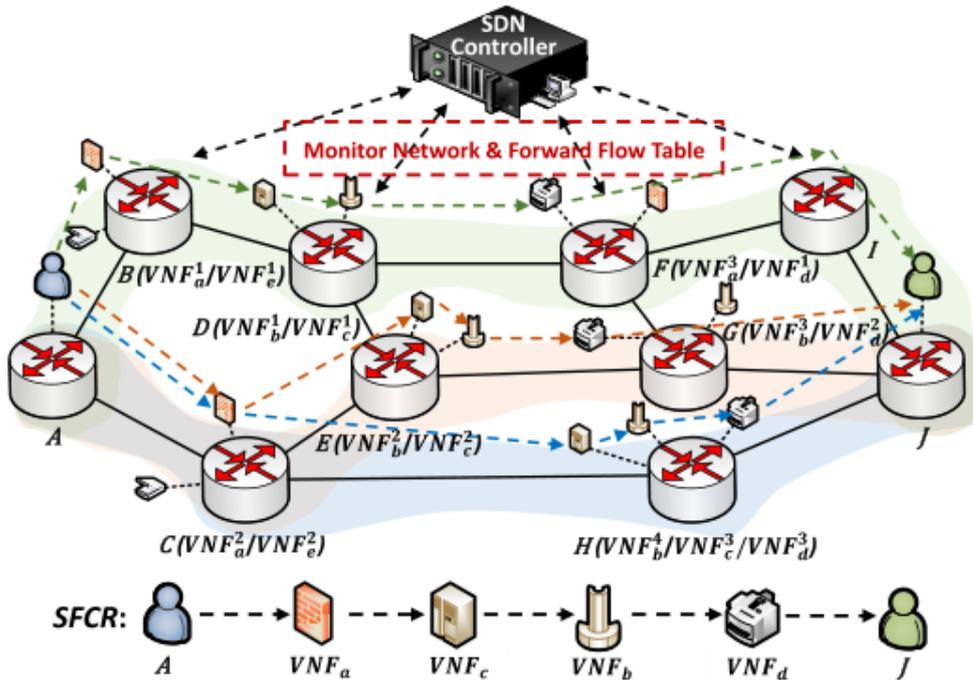


图 2-1 在物理网络中部署 SFC 请求^[16]

下面分别介绍 SFC 技术的两项关键网络技术，网络功能虚拟化和软件定义网络。

2.1.1 网络功能虚拟化

众所周知，由于现有硬件设备的专有性质、为各类中间盒提供空间和能源的成本以及缺乏熟练的专业人员来集成等因素，将新的服务引入现有的网络并对这些服务进行维护变得越来越困难。而网络功能虚拟化 (Network Function Virtualization, NFV) 技术则有望解决上述问题^[40]。

NFV 技术利用成熟的虚拟化技术将软件实例与硬件平台分离，并通过将功能与位置分离来加快网络服务的部署，从而改变网络运营商构建其基础架构的方式。在本质上，NFV 通过软件虚拟化技术来实现网络功能，并在通用的硬件（如行业标准的服务器、存储和交换机，等）上对其进行部署，且这些虚拟的网络功能可以根据业务的具体需求来进行实例化，而不需要购买新的专用物理设备。例如，网络运营商可以在基于通用的 x86 平台的虚拟机上运行开源的虚拟防火墙。对于其他网络功能（如，用于蜂窝核心网的组件功能、用于物理层的信号处理功能，等），他们部署的可行性已经得到了验证^[41,42]。

NFV 技术的框架^[43]如图 2-2 所示，包括编排管理模块、VNF 管理模块、虚拟基础设施管理模块，以及虚拟基础设施。其中，编排管理模块负责管理软件资源以及虚拟化的硬件基础设施，以实现各种网络服务。VNF 管理模块负责 VNF 生命周期中的实例化，资源调整以及信息的更新。虚拟基础设施中的虚拟化层负

责把各种物理资源（如计算资源、网络资源和存储资源）进行抽象，并把 VNFs 锚定到虚拟基础设施上，这使得 VNF 的生命周期对于下层硬件平台而言是独立的。虚拟基础设施管理模块被用于对可配置的计算、网络和存储资源进行虚拟化和和管理，并控制这些资源与 VNFs 的交互过程。

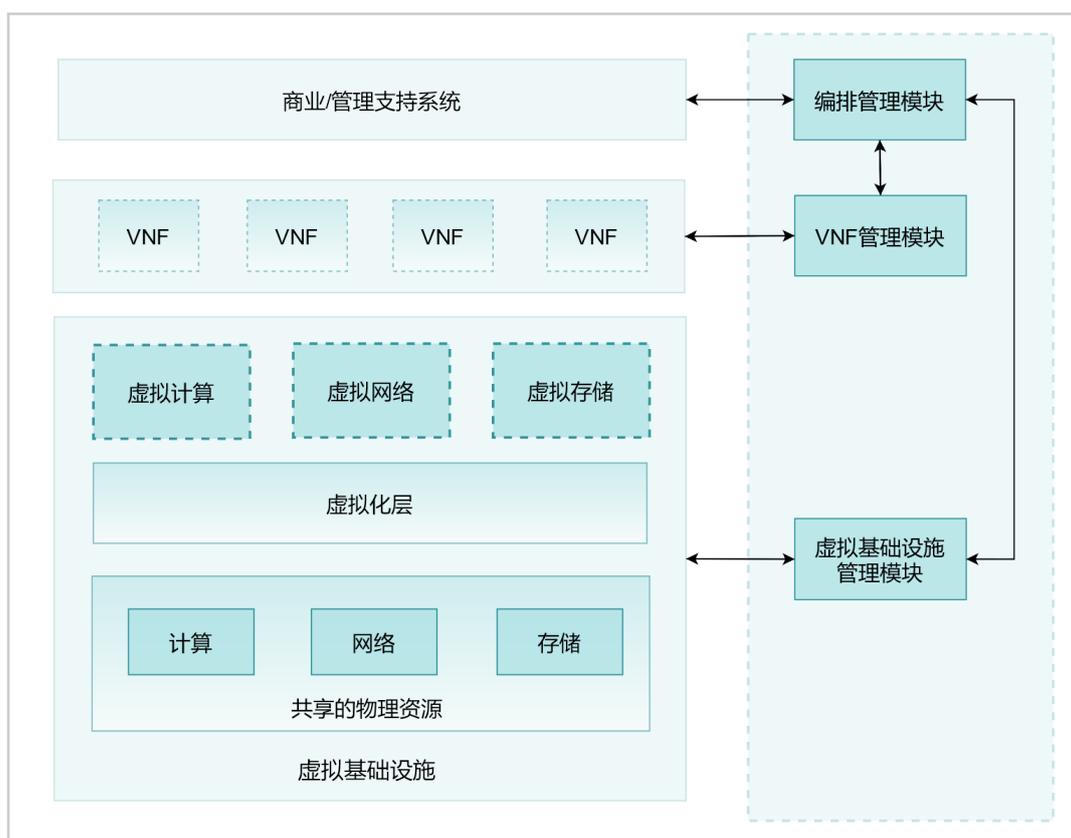


图 2-2 NFV 架构

相比于传统网络，NFV 技术具备以下三个特点。

1. 软硬件分离。这种分离使得软件能够独立于硬件发展，反之亦然。
2. 灵活地部署网络功能。NFV 可以在硬件资源池上自动对网络功能软件进行部署，而这些硬件资源池可能在不同的数据中心的不同时间运行着不同的功能。
3. 动态业务调整。网络运营商可以根据当前用户的需求来动态调整 VNF 的性能，以进一步保障业务的 QoE。

随着 NFV 技术的不断发展，运营商还可以把 VNFs 部署在集群上^[8]。为了克服单点故障和可扩展性差的限制，单个网络功能可以同时有多个虚拟化实例，且这些实例被部署在不同的通用的服务器上。

2.1.2 软件定义网络

软件定义网络 (Software Defined Network, SDN) 是一种新型的网络架构, 它创新性地 把网络的转发平面与控制平面进行解耦合, 以实现 对全局网络的动态管理。SDN 的架构^[44]如图 2-3 所示, 包含基础设施层, 控制层以及应用层。其中, 基础设施层包含数据平面上的转发设备, 这些设备主要负责根据控制器下发的转发规则对收到的数据包进行转发。控制层通过南向接口与基础设施层进行交互, 从而获取到网络拓扑, 网络流的统计数据等信息。此外控制层还通过北向接口与应用层进行交互, 为应用层提供各种网络状态信息。应用层则包含大量的应用以满足客户的各种需求。通过控制层提供的可编程平台, SDN 应用程序 (如动态访问控制、服务器负载均衡, 网络流重路由, 等) 可以访问和控制基础设施层的设备。SDN 的三层架构使得运营商能够以灵活的, 低成本的方式对全局网络进行管理, 并提高网络的整体性能。

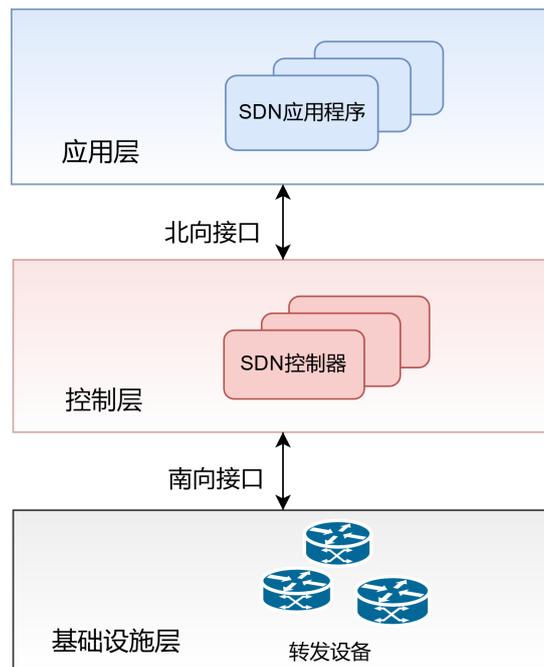


图 2-3 SDN 架构

2.2 组合多摇臂赌博机模型

多摇臂赌博机 (Multi-Armed Bandit, MAB)^[45]是统计学及机器学习领域中的一个研究热点。一般地, MAB 模型可以被表示为在一个具备 m 个摇臂或机器的系统中, 反复地在每个回合内对摇臂或机器进行选择, 以此来使得总的奖赏能够尽可能地接近最佳摇臂的奖励。然而, 每个摇臂或机器的奖赏分布都是未知的。

MAB 算法需要在每个回合内，根据之前所获取到的摇臂的随机奖赏信息来决定拉动哪个摇臂或机器。而衡量 MAB 算法的指标可以表示为累积奖赏或累积遗憾（总是选择最佳摇臂和总数选择按照算法输出的摇臂的累积奖赏的差值）。

然而，许多实际的应用场景并不能简单地定义为 MAB 模型。例如，在线广告投放问题需要决策者在信息缺失的情况下决定把广告投放在特定的 K 个网页上，以最大化用户的点击量。另外的一个例子为社交网络中的营销问题，销售人员需要在网络中反复地选择节点对，以观察并学习整个社交网络中个体之间的关联性，以提高营销效果。在上述例子中，如果把一个网页，一个节点对视为一个摇臂，则玩家在每个回合中都需要同时选中 K 个独立的摇臂，而不是逐一地对摇臂进行选择。通常，这类问题可以被定义为组合多摇臂赌博机 (Combinatorial Multi-Armed Bandit, CMAB)^[46]模型。下面分别介绍 CMAB 模型的两个重要概念，超臂和奖赏。

2.2.1 超臂

一般地，假设一个 CMAB 模型中包含 m 个摇臂，令 $\mathcal{S} \subseteq 2^{[m]}$ ，其中 $2^{[m]}$ 为摇臂所有可能的子集的集合。则在每个回合内，玩家可执行的单个动作 $S \in \mathcal{S}$ 可视为一个超臂。特殊地，若玩家在每个回合内都需要同时选取 K 个相互独立的摇臂，则玩家在单个回合内可选择的超臂的组合数上限为 C_m^K 。

2.2.2 奖赏

在 MAB 模型中，玩家在回合 t 内选择摇臂 i 所获得的随机奖赏可以表征为 $X_{i,t}$ 。一般地，每个摇臂的奖赏数据会服从一个未知的数学分布 u_i ，且摇臂与摇臂之间是相互独立的。则易得，在 CMAB 模型中，当玩家在回合 t 内选择超臂 S 后， $\forall i \in S, X_{i,t}$ 的信息都是已知的。若把超臂 S 在回合 t 内的随机奖赏记为 $R_t(S)$ ，则在部分情况下， $R_t(S) = \sum_{i \in S} X_{i,t}$ 。在少数的应用场景中（如最大化社会影响力，在线广告投放，等），摇臂奖赏的线性相加可能无法精确地反馈决策的奖赏情况，因此 $R_t(S)$ 的计算需要根据具体的应用场景进行调整。

2.3 时间序列预测技术

时间序列预测是预测的一个重要分支,它通过收集和分析过去对同一变量的观察结果,以建立描述潜在关系的模型。然后,该模型被用于推断时间序列数据在未来一段时间的发展趋势,数值区间等^[47]。时间序列预测技术对科学应用有深刻的影响,如社会科学家跟踪人口的时间序列数据来推断未来的出生率和入学率,流行病学家可能会对一段时间内观察到的病例数感兴趣。此外,时间序列预测技术还深刻地影响着人们的日常生活,如天气预测的本质就是时间序列的预测。随着研究工作的不断深入,科研人员设计出了大量优秀的时间序列预测技术,下面介绍3种常见的时间序列预测方法。

2.3.1 差分整合移动平均自回归模型

差分整合移动平均自回归模型 (Autoregressive Integrated Moving Average model, ARIMA) 由 Box-Jenkins 提出^[48],是一个经典的用于对时间序列进行分析和预测的模型。一般地,ARIMA 模型由参数 p, d, q 来确定,本小节将介绍 ARIMA 模型的组成部分及基本参数。

1. 自回归模型 (Autoregressive Model, AR)。AR 是基于目标变量历史数据的组合来对目标变量进行预测,通常,一个 p 阶的自回归模型可以表示为式子 (2-1)。

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2-1)$$

其中, y_t 为目标值, c 为系数, ϕ 为权重值, ε_t 为白噪声。AR 在处理拥有复杂特征的时间序列上非常灵活,通常权重值 ϕ 的变化会使得时间序列拥有不同的特征,而误差项 ε_t 的方差只会改变序列的数值范围而不会改变它的特征。

2. 移动平均模型 (Moving Average Model, MA)。与使用预测变量的历史值进行回归不同,MA 通过历史预测误差来建立回归模型,通常,一个 q 阶的移动平均模型可以表示为式子 (2-2)。

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (2-2)$$

其中, c 为系数, θ 为权重值, ε_t 为白噪声。值得注意的是, y_t 的每一个值都可以被认为是一个历史预测误差的加权移动平均值。

3. 差分。差分是一种通过计算相邻观测值之间的差值的方法,这是一种常见的使非平稳序列变平稳的方法。一般地,一阶差分表示为 $y_t' = y_t - y_{t-1}$,而二阶差分则表示为 $y_t'' = y_t - 2y_{t-1} + y_{t-2}$ 。在现实的应用中,通常没有必

要进行二阶以上的差分^[49]。

当把差分、AR、MA 结合起来时，就可以得到 ARIMA 模型，一般地，ARIMA 模型可以被表示为式子 (2-3)。

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2-3)$$

其中 y'_t 是差分序列（可能经过多次差分），而右边则包括了预测变量 y_t 的延迟值和延迟误差。

2.3.2 长短时记忆神经网络

长短时记忆神经网络 (Long-short term memory, LSTM)^[50] 是一种特殊的神经网络^[51]。在传统的神经网络的结构中，信号流从输入层到输出层依次流过，同一层的神经元之间的信号是不会相互传递的。这样的设计会导致输出信号仅与输入信号相关，而与输入信号的先后顺序无关，且神经元并不具备“记忆”能力。为了使得网络模型能更好地提取输入信号的时间性信息，研究人员提出了循环神经网络模型 (Recurrent Neural Network, RNN)^[52]。RNN 利用神经网络的内部循环来保留时间序列的上下文信息，随着输入数据的不断增加，RNN 的自我循环结构会把上一个时间刻的状态传递给当前的单元来一起作为新的输入数据进行训练。然而，随着时间推移，这种设计模式很容易会出现梯度爆炸或梯度消失的问题，因而无法获取长距离数据的信息。

LSTM 通过对 RNN 进行结构上的改造来规避长期依赖问题，其结构如图 2-4 所示。具体地，LSTM 包含三个组成部分，即遗忘门，输入门和输出门，对应图 2-4 的三个 σ ，下面本文对三个组成部分进行详细介绍。

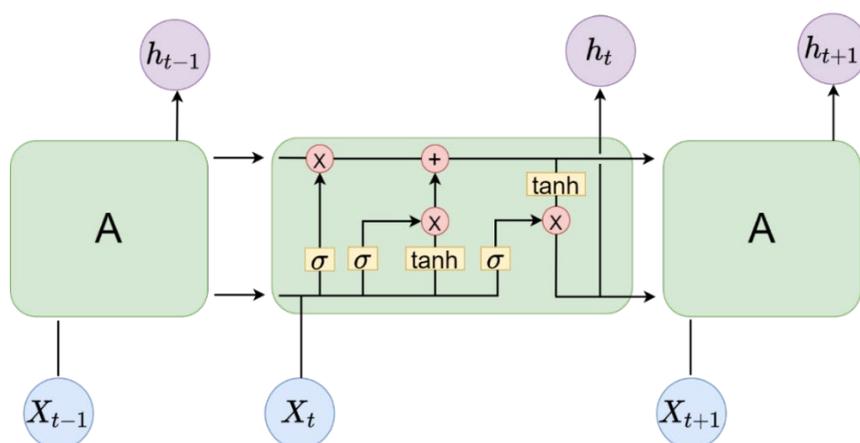


图 2-4 LSTM 结构

1. 遗忘门。遗忘门是 LSTM 模型的关键组成部分，它通过读取 h_{t-1} 和当前时刻的输入 x_t ，并利用权重矩阵 w_f 和偏置项 b_f 来对输入的数据进行处理，

然后通过 sigmoid 函数把处理后的数值替换成 0 到 1 之间的数值，从而决定应该保留哪部分信息。上述操作可以表示为式子 (2-4)，其中 sigmoid 函数可以表示为 $\sigma(x) = 1/(1 + e^{-x})$ 。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2-4)$$

2. 输入门。输入门负责确定什么样的信息应该被存放到单元当中，这包含两个部分，首先输入门需要对当前时刻的输入 x_t 进行有选择性的记忆，其主要通过 sigmoid 函数及权重矩阵 W_i ，偏置项 b_i 来处理输入信息，然后得到输入门的门控状态。上述处理过程见式子 (2-5)。然后，输入门还需要结合遗忘门保留的上一时刻的单元状态信息来计算当前时刻的单元状态信息，具体过程见式子 (2-6)。由于遗忘门和输入门的共同作用，LSTM 可以保存很久之前的信息，同时也可以避免存储太多的无用信息。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2-5)$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2-6)$$

3. 输出门。输出门被用于决定当前单元应该将哪些信息作为当前时刻的输出。同样地，输出门通过 sigmoid 函数以及权重矩阵 W_o ，偏置项 b_o 来得到输出门的门控状态，然后确定经过 tanh 激活函数变化后，哪部分信息应该被输出，上述处理过程见式子 (2-7)。

$$h_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) * \tanh(c_t) \quad (2-7)$$

2.3.3 Prophet

Prophet^[53]是一个开源的时间序列预测模型，它具有灵活性高，自动补全缺失值，拟合速度快，可解释性高等一系列优点。Prophet 模型通过将时间序列数据分解成趋势项 g_t ，周期项 s_t ，节假日项 h_t 以及残差项 ε_t ，如式子 (2-8) 所示，下面分别对其进行介绍。

$$y_t = g_t + s_t + h_t + \varepsilon_t \quad (2-8)$$

1. 趋势项 g_t 。一般地， g_t 包含基于逻辑回归的饱和增长模型和分段线性模型。基于逻辑回归的模型会随着时间推移而达到饱和，其基本形式如式子 (2-9) 所示，其中 C 为增长饱和值， k 为增长速度， m 为偏移参数。

$$g_t = \frac{C}{1 + \exp(-k(t - m))} \quad (2-9)$$

而线性增长模型则适用于预测没有饱和增长趋势的数据，其基本形式如式子 (2-10) 所示，其中 k 为增长速度， $a(t)$ 为突变点的集合， δ 为增长率的变化值向量， m 为偏移参数， γ 为使得趋势函数连续的向量。

$$g_t = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \quad (2-10)$$

2. 周期项 s_t 。Prophet 使用傅里叶级数来建立周期模型，以拟合时间序列数据的周期性，其基本形式如式子 (2-11) 所示，其中 P 为周期。

$$s_t = \sum_{n=1}^N (a_n \cos(\frac{2\pi nt}{P}) + b_n \sin(\frac{2\pi nt}{P})) \quad (2-11)$$

3. 节假日项 h_t 。由于节假日和特殊时间对时间序列会有较大的影响，且一般这种影响不具备周期性。因此，为了消除这类影响，Prophet 认为节假日的影响是独立的，且对与每个节假日，令 D_i 为该节假日附近几天的日期，然后设定参数 k 来表示该节假日会带来影响。为实现上述的理论，Prophet 生成一个回归矩阵 $Z_t = [1(t \in D_1), \dots, 1(t \in D_L)]$ ，则节假日项 h_t 可以表示为式子 (2-12)。

$$h_t = Z_t k \quad (2-12)$$

4. 残差项 ε_t 。残差项 ε_t 代表 Prophet 模型中不能适应的任何特殊变化，并假设这些变化符合正态分布。

2.4 本章小结

本章首先介绍了组合多摇臂赌博机模型的相关概念，接着阐述了 5G 网络服务功能链的相关技术，并重点介绍了与其相关的两个关键网络技术，网络功能虚拟化和软件定义网络。最后对时间序列预测的三个常用方法差分整合移动平均自回归模型、长短时记忆神经网络以及 Prophet 进行了详细介绍。

3 基于 CMAB 的关键业务流识别策略

3.1 引言

在现实网络中,网络流量的分布是不均衡的^[34],即 80%以上的流量大小都是小于 10KB,并且网络中大部分的数据包都是由少于 10%的大流量产生的,这意味着网络中的少数流量有非常大的带宽需求。在各种各样的应用场景中,如网络流重路由^[54]、异常检测^[55]、网络切片资源调整^[56,57]、时延敏感网络^[58,59],识别或提前预测出具有大带宽需求的网络流(top- K 流)是非常有必要的。

由于在 SDN 中,网络管理人员可以及时并高效地搜集网络流的统计数据^[60,61],一些研究人员基于 SDN 以及网络分析(Network Analytics, NA)^[62]技术提出了知识定义网络(Knowledge-Defined Networking, KDN)^[63]的概念。如图 3-1 (a)所示,KDN 在网络中设置了一个人工大脑来获取网络当中的知识(如,如何动态识别 top- K 流)以及如何应用这些知识来设计各类网络管理策略。然而,动态识别网络中的 top- K 流是一项非常具有挑战性的任务。一方面,由于组合爆炸,问题的求解空间会是网络规模的指数级。如,在 M 条流中识别出 top- K 流,这个问题的解一共有 C_M^K 种组合,本文简单地令 $K=10, M=100$,则解空间会有多达 17 万亿^[32]种可能的组合。另一方面,不同类型的业务的带宽需求是动态变化的^[64,65],尽管把 K 简单地设置为 1, top- K 流的预测任务仍然是十分困难的。如图 3-1 (b)所示,从时间段 t_0 到 t_1 ,流 1 的带宽需求是最高的,因此流 1 是目标流。然而,各个流的带宽需求会随着时间推移而波动,因此目标流也会随着时间而变化。从 t_1 到 t_3 ,目标流为流 2,而 t_3 之后,目标流则变为流 3。

尽管一些研究人员探索了如何在特定的应用场景中有效地识别出 top- K 大象流^[66]或识别出 top- K 关键流^[32],他们的研究主要集中在单个时间窗内识别 top- K 流,且他们忽略了网络流带宽需求的时变性。如图 3-1 所示,为了解决各种流带宽需求的不确定性,设计一种高效的 top- K 流识别机制是非常有必要的。

本章节研究了如何在 SDN 中动态地识别出不断变化的 top- K 流。本文首先把 top- K 流识别问题定义成随机多摇臂赌博机组合模型(Combinatorial Multi-Armed Bandit, CMAB)^[46],与现有的工作不同,本文增加了一些必要的假设与约束来适应现实的应用场景,具体为, a) 摇臂奖赏分布的时变性。b). 弱化探索与利用的均衡性。此外,在本文中,关键业务流的数目还会随着网络的规模而发生变化,上述对比的总结如表 3-1 所示。接着,本文基于贪婪思想设计了一个通用的摇臂选择方法来对问题进行求解,并采集了真实的网络流量数据来生成摇臂的奖赏值。最后通过仿真实验进行算法性能的验证。

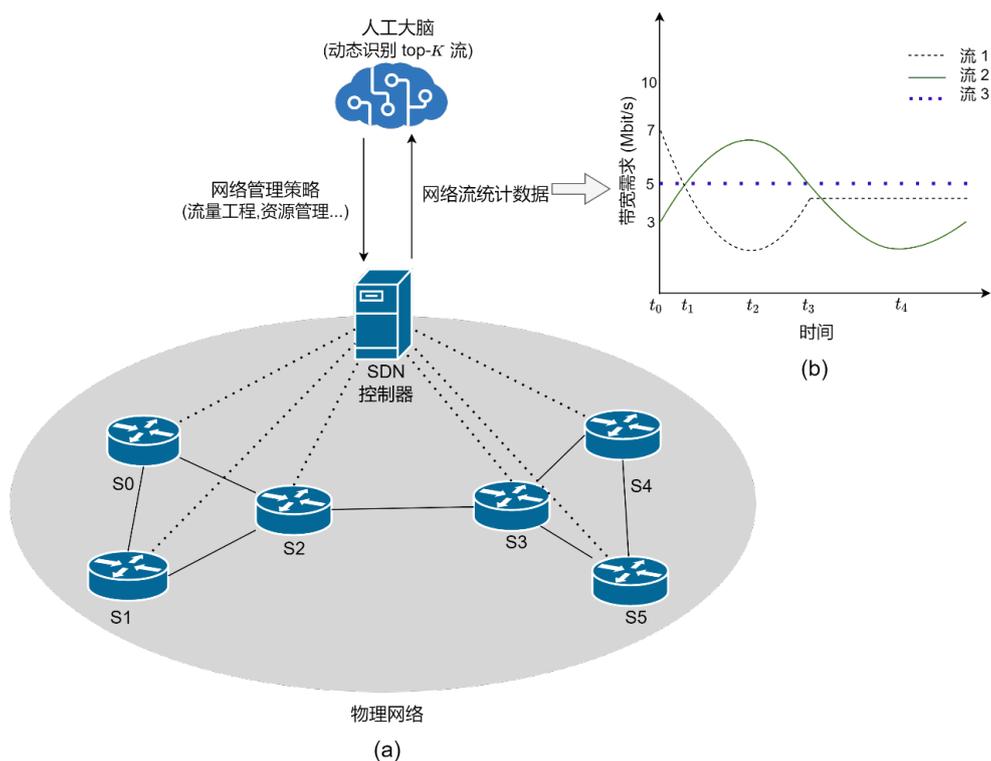


图 3-1 基于 KDN 的 top- K 流识别

表 3-1 本文工作与现有 top- K 摇臂识别工作的对比分析

文献作者	top- K 摇臂的数量	摇臂的奖赏分布	是否考虑探索与利用的均衡
Audibert <i>et al.</i> ^[67]	1	服从伯努利分布	是
Bubeck <i>et al.</i> ^[68]	随摇臂总数而变化	服从伯努利分布	是
Zhang <i>et al.</i> ^[69]	不超过 5	服从高斯分布或指数分布	是
Zhuang <i>et al.</i> ^[70]	不超过 3	服从伯努利分布	是
本文所提方法	随网络规模发生变化	由真实网络流量数据生成, 具有时变性	否

3.2 问题陈述

本节将系统地描述 top- K 流的识别问题, 为了方便参考, 表 3-2 总结了本章所使用的符号。

表 3-2 top- K 流识别问题所使用的符号总结

符号	定义
N	网络中节点的总数
n	回合总数
M	摇臂总数
K	top- K 摇臂总数
y_i	第 i 个摇臂的奖赏分布
$X_{i,t}$	第 i 个摇臂在第 t 个回合所获得的随机奖赏
$\mathcal{X}_{i,t}$	截止至第 t 个回合, 第 i 个摇臂所获得的所有奖赏信息的集合
$X_{i,t}^E$	第 i 个摇臂在第 t 个回合所获得的预期奖赏
S_t	在第 t 个回合所选中的超臂
S_t^*	在第 t 个回合的最优超臂
r	累积遗憾
T	采样间隔
w	权重
d	滑动窗口

3.2.1 问题定义

在 SDN 中, 每隔一个采样间隔 (如, 每 5 分钟, 每 15 分钟, 等), 网络管理人员都需要根据网络数据流的历史信息来识别出下一个时间段中有高带宽资源需求的 top- K 流, 在本文的工作中, 一条网络流被定义成一个源节点到一个目的节点对^[32]。与相关研究工作^[71,72]类似, 动态预测过程如图 3-2 所示, 其中, D_{T_n} 表示从 t_{n-1} 到 t_n 收集到的统计数据。

该动态识别问题可以转化为 CMAB 模型。假设网络拓扑中一共有 N 个节点, 则在 CMAB 模型中, 一共有 M 个摇臂, 每过指定的采样间隔 T , 玩家都需要从 M 个摇臂中选出 K 摇臂, 其中 $M = N*(N-1), K = 10\%M$ ^[32]。如果把网络当中的一条流视为一个摇臂, 并且把该流在 $t-1$ 到 t 时间段内的带宽需求 $X_{i,t}$ 视为摇臂在第 t 个回合产生的随机奖赏, 则动态识别问题的目标可以被表示为式子 (3-1), 其中 n 代表回合总数。

$$\text{maximize } \sum_t^n \sum_i^K X_{i,t} \quad (3-1)$$

此外, 玩家在回合 t 中选中的摇臂集合 S_t 可以定义为一个超臂^[46], 显然, 在单回合内超臂的组合数为 C_M^K 。本文假设在回合 t 内可以最大化玩家收益的超臂定义为最佳超臂 S_t^* , 则由于最大化累积奖赏在本质上等价于最小化 S_t^* 与 S_t 之间的

差异^[73]，因此式子 (3-1) 可以被进一步地表示成式子 (3-2)。在本文中，超臂的奖赏被定义成超臂中所有元素的随机奖赏的线性叠加，即本文并不考虑更为复杂的超臂奖赏情况。

$$\text{minimize } r = \sum_t (\sum_{i \in S_t} X_{i,t} - \sum_{j \in S_t} X_{j,t}) \quad (3-2)$$

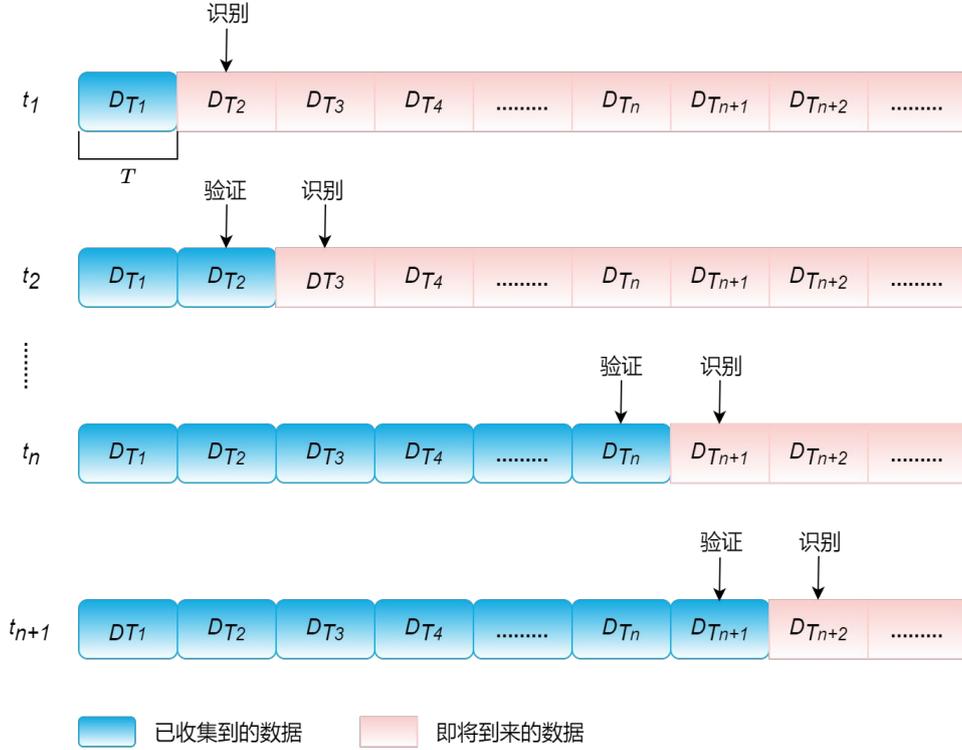


图 3-2 动态识别网络中的 top-K 流

3.2.2 模型修订

为了使 CMAB 模型更加契合于实际的应用场景，本文增加了以下的限制以及假设。

1. 奖赏的时变性。在 CMAB 或 MAB 模型中，每一个摇臂都与一个特定的奖赏分布相关联，这个奖赏分布通常是固定的。一般地，第 i 个摇臂的奖赏分布可以被表示为 y_i ，而玩家是不会提前知道 y_i 的相关信息的。在 SDN 中，奖赏分布不仅是不可见的，而且还会因为业务需求的波动而具有时变性^[64,65]。因此，本文把 y_i 重定义为 $y_i(t)$ ，即摇臂的奖赏分布是一个与时间相关的函数。
2. 奖赏信息共享。如何平衡探索与利用的关系一直是 CMAB 模型中的研究热点。但是，由于 SDN 的全局管理能力及灵活性，网络运营商可以及时地收集到全局网络流的统计信息，如流量矩阵信息。因此，本文并不考虑

探索与利用之间的均衡问题。在每一回合结束后，玩家可以获得全部摇臂在上一回合的随机奖赏信息，即在每一回合后， $\forall i \in M, X_{i,t}$ 都是可见的。

尽管本文不需要考虑探索与利用之间的均衡问题，但是由于摇臂奖赏的时变性，如何有效地利用历史奖赏信息来计算摇臂 i 在 $t+1$ 回合的期望奖赏 $X_{i,t+1}^E$ 是困难的^[74]。简而言之，一个摇臂保留的历史奖赏信息越少，对摇臂与平均奖赏估计相关的随机误差就会越大，而同时使用过多的历史奖赏数据来预测一个摇臂的期望奖赏则会增加这些数据存在偏差的风险。

3.3 问题求解

本节先介绍一种基于贪婪思想的摇臂选取方法，然后再介绍了多种不同的选取依据。

3.3.1 贪婪摇臂选取算法

1. 动机。贪婪是一个非常重要的思想^[75]。尽管贪婪思想不能提供理论保证，但是它不仅简单易行，且能够很好地解决许多实际问题。在 MAB 模型中，贪婪思想的核心是在每一个回合 t ，玩家总是会执行期望奖赏最大的动作 A_t ，其中 $A_t = \operatorname{argmax}_{i \in M} X_{i,t}^E$ ，而 $X_{i,t}^E$ 可以基于第 i 个摇臂的所有历史奖赏数据 $\mathcal{X}_{i,t} = \{X_{i,1}, X_{i,2}, \dots, X_{i,t-1}\}$ 计算得出。本文所设计的算法在每一回合内都会贪婪地选取期望奖赏排在前 K 的摇臂，从而最大化总期望奖赏。
2. 实现。通常，贪婪思想可以通过以下步骤来实现。(1) 对每个摇臂 $i \in M$ ，根据其历史随机奖赏数据 $\mathcal{X}_{i,t}$ 计算它在下一回合的期望奖赏。(2) 根据期望奖赏对所有摇臂进行排序。(3) 选取前 K 个摇臂作为下一个回合需要执行的超臂。注意，如图 3-2 所示，为了避免摇臂奖赏初始化的问题，本文所述的 CMAB 模型从第二个回合开始启动。

上述算法的过程的伪代码如算法 3-1 所示，其中，第 8 到第 9 行表明，在每个回合结束后，算法需要对一些必要的信息进行更新维护，如历史奖赏信息。

算法 3-1 贪婪摇臂选取算法

输入： 摇臂总数 M ，top- K 摇臂总数 K ，总回合数 n ，所有摇臂的初始历史随机奖赏信息 $\mathcal{X}_{i,t}, \forall i \in M$ 。

输出： 累积遗憾 r 。

- 1: 初始化，令 $r = 0, t = 1$;
- 2: **while** $n > 0$ **do**
- 3: **for** 每个摇臂 $i \in M$ **do**
- 4: 根据不同的计算标准（式子 (3-3), (3-4), (3-5)）以及每个摇臂的历史随机奖赏信息 $\mathcal{X}_{i,t}$ 来计算摇臂在下一回合的期望奖赏 $X_{i,t+1}^E$;
- 5: **end for**
- 6: 根据期望奖赏对所有摇臂进行降序排序;
- 7: 令 $S_{t+1} = \{i_1, i_2, \dots, i_K\}$; // 选取超臂
- 8: 令 $t = t + 1, n = n - 1$; // 当前回合结束
- 9: For $i \in M$, let $\mathcal{X}_{i,t+1} = \mathcal{X}_{i,t} + X_{i,t+1}$; // 维护历史奖赏信息
- 10: $r = r + (\sum_{i \in S_{t+1}} X_{i,t+1} - \sum_{j \in S_{t+1}} X_{j,t+1})$; // 维护累积遗憾
- 11: **end while**
- 12: **return** r ;

3.3.2 摇臂选取依据

如前所述，在动态识别 top- K 摇臂问题中，如何计算每个摇臂的期望奖赏是一个困难的问题。本节介绍了本文所提出的不同的期望奖赏计算方法，从而为摇臂的选取提供依据。

1. **mean-greedy:** 计算摇臂 i 的历史随机奖赏数据的平均值作为其期望奖赏是一个朴素的计算方法^[76,77]，本文称这种方法为 **mean-greedy**。其中， $X_{i,t+1}^E$ 可以通过式子 (3-3) 来计算。

$$X_{i,t+1}^E = \frac{X_{i,1} + X_{i,2} + \dots + X_{i,t}}{t} \quad (3-3)$$

2. **weighted-greedy:** 由于 **mean-greedy** 策略仅仅适用于平稳的摇臂模型，在一些非平稳的摇臂模型中，更加关注摇臂近期所获得的随机奖赏信息是一个明智的决策^[78]，本文把这种策略称为 **weighted-greedy**。在这种策略中， $X_{i,t+1}^E$ 可以通过式子 (3-4) 来进行计算，其中， $w \in (0,1]$ 是权重参数。

$$\begin{aligned} X_{i,t+1}^E &= wX_{i,t} + (1-w)^2 wX_{i,t-2} + \dots + (1-w)^{t-1} wX_{i,1} \\ &= \sum_{j=1}^t (1-w)^{t-j} wX_{i,j} \end{aligned} \quad (3-4)$$

3. sliding window-greedy: 与 weighted-greedy 策略类似, sliding window-greedy 也是注重于摇臂的近期随机奖赏信息。不同的是, sliding window-greedy 只关注摇臂最近的 d 个随机奖赏信息, 而完全忽略其他的随机奖赏信息, 因此 $X_{i,t+1}^E$ 可以通过式子 (3-5) 来计算, 其中 $d \geq 1$ 为滑动窗口。

$$X_{i,t+1}^E = \frac{\sum_{j=t-d}^t X_{i,j}}{d} \quad (3-5)$$

3.3.3 复杂度分析

本节将分析本文所提的贪婪选取算法在单个回合内 (算法 3-1 的第 3 到第 8 行) 的复杂度。首先, 对第一个 **for** 循环, 算法需要计算所有摇臂的期望奖赏值 $X_{i,t+1}^E$ 。注意, 算法并不需要使用全部的历史奖赏信息。对于 mean-greedy 而言, $X_{i,t+1}^E = \frac{(t-1) \cdot X_{i,t-1}^E + X_{i,t}}{t}$, 对 weighed-greedy 而言, $X_{i,t+1}^E = (1-w) \cdot X_{i,t-1}^E + w \cdot X_{i,t}$ 。因此, 对于上述两种策略, 仅需要维护上一回合的期望奖赏信息就已经足够了。然而, 对于 sliding window-greedy 策略, 摇臂期望奖赏的计算必须依赖于近期的 d 个随机奖赏信息, 对于这个情况, 每个摇臂需要维护 d 个信息。综合考虑排序的时间复杂度 $O(M \log_2 M)$, 每种策略的时间复杂度, 空间复杂度的分析结果如表 3-3 所示。

表 3-3 本文所提的贪婪摇臂选取算法的复杂度分析

方法	时间复杂度	空间复杂度
mean-greedy	$O(M + M \log_2 M)$	$O(M)$
weighted-greedy	$O(M + M \log_2 M)$	$O(M)$
sliding window-greedy	$O(M + M \log_2 M)$	$O(dM)$

3.4 仿真与性能分析

本节评估了本文所提出的贪婪摇臂选取算法的性能。首先介绍了本文所使用的数据集, 接着介绍了先进的 top- K 摇臂的识别算法。最后, 本节进行了基于真实网络流量数据的仿真实验来对比本文所提算法与目前先进的 top- K 摇臂选取算法间的性能差异。

本章所提算法的运行环境如下：操作平台: Windows 11; 硬件平台: Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz; 编程语言: Python 3.6; 具体算法实现如算法 3-1 所示。此外，本章所采用的数据集、对比算法和评价指标的具体内容如章节 3.4.1, 3.4.2 和 3.4.3 所示。

3.4.1 数据集

本章使用了两个被广泛应用于计算机网络领域研究^[32,79]的真实网络拓扑 (Abilene, Geant) 的数据来验证不同的摇臂选取算法的性能。表 3-4 汇总了所用拓扑的属性信息，如节点数，采样间隔，top- K 流的总数。对于 Abilene 网络，本文收集了 6 天的带宽需求数据（从 2004 年 4 月 2 日开始），一共有 1728 个矩阵信息。对于 Geant 网络，本文收集了一周的带宽需求数据（从 2005 年 5 月 5 日开始），一共有 672 个矩阵信息。

表 3-4 top- K 流识别算法仿真拓扑参数

网络拓扑	N	M	K	T	n
Abilene	12	132	13	5 分钟	1728
Geant	22	462	46	15 分钟	672

3.4.2 对比算法介绍

本文选择了两个先进的 top- K 摇臂选取算法 SAR^[68], Q-SAR^[69]作为对比算法，并将随机选择算法作为最弱的基线算法。如果某个摇臂选择算法的性能弱于随机选择算法，则该算法可以视为无效算法。SAR 和 Q-SAR 算法是 MAB 模型中识别 top- K 摇臂的优秀算法，本文先对这些算法进行简要的介绍。

1. Random: 随机选择算法完全忽略每个摇臂的历史奖赏信息，在每一个回合中，该算法随机选择 K 个不同的摇臂。
2. SAR^[68]: SAR 算法专注于在固定预算的前提下识别出多摇臂赌博机游戏中的 top- K 个摇臂。该算法首先把所有的预算切分成 $M-1$ 个阶段，然后在每个阶段，SAR 以相同的频率来拉动每个摇臂。在每个阶段结束后，SAR 要么接收平均奖赏最高的摇臂，要么移除平均奖赏最低的摇臂。
3. Q-SAR^[69]: Q-SAR 是 SAR 的改进版本。与 SAR 类似，Q-SAR 同样先把预算切分成 $M-1$ 个阶段，并在每个阶段中以相同的频率来拉到每个摇臂。与 SAR 不同的是，Q-SAR 利用分位数作为评价指标，然后根据最优指标与最低指标的差距来决定是否评定一个摇臂为最优摇臂。

SAR 与 Q-SAR 的最终目标都是在利用完固定的预算后，输出一个摇臂的集

合 $\{i_1, i_2, \dots, i_K\}$ ，该集合对应于期望奖赏最高的 K 个摇臂。在原始的相关研究工作中，每个摇臂都有一个固定的奖赏分布，由于这些不会随着时间的推移而发生变化，这类问题可以被定义为离线问题。本文利用前 70% 的带宽需求数据作为训练数据（在 Abilene 网络中，训练数据开始于 2004 年 4 月 4 日 00:00，结束于 2004 年 4 月 6 日 06:10，在 Geant 网络中，训练数据开始于 2005 年 5 月 5 日 00:00，结束于 2005 年 5 月 9 日 21:30），并利用训练数据来为 SAR 及 Q-SAR 算法来随机生成摇臂的奖赏数据。然后，本文利用剩余 30% 的数据作为测试数据来验证所有算法的性能。

3.4.3 评价指标

为了更加直观地反映每个算法的性能，除了式子 (3-2) 中定义的累积遗憾外，本文还定义了 $R_{S^*,S}(t)$ 来衡量每个算法输出的超臂与理论最优超臂的累计差异，其中 $R_{S^*,S}(t)$ 可以通过式子 (3-6) 来计算。

$$R_{S^*,S}(t) = \frac{\sum_t \sum_{i \in S_t} X_{i,t}}{\sum_t \sum_{i \in S_t^*} X_{i,t}} \quad (3-6)$$

3.4.4 仿真结果及分析

本文先验证贪婪思想的有效性，然后对比随机选择算法与朴素的 mean-greedy 策略间的差异。如图 3-3，图 3-4 所示，在一段时间内，mean-greedy 策略可以在每个回合内有效地输出理论上的最优超臂，且性能远远超过随机选择算法。然而，在一段时间后（图 3-3 的 500 回合后，图 3-4 的 1150 回合后），随着每个摇臂的奖赏分布发生变化，mean-greedy 策略的性能开始出现衰减。

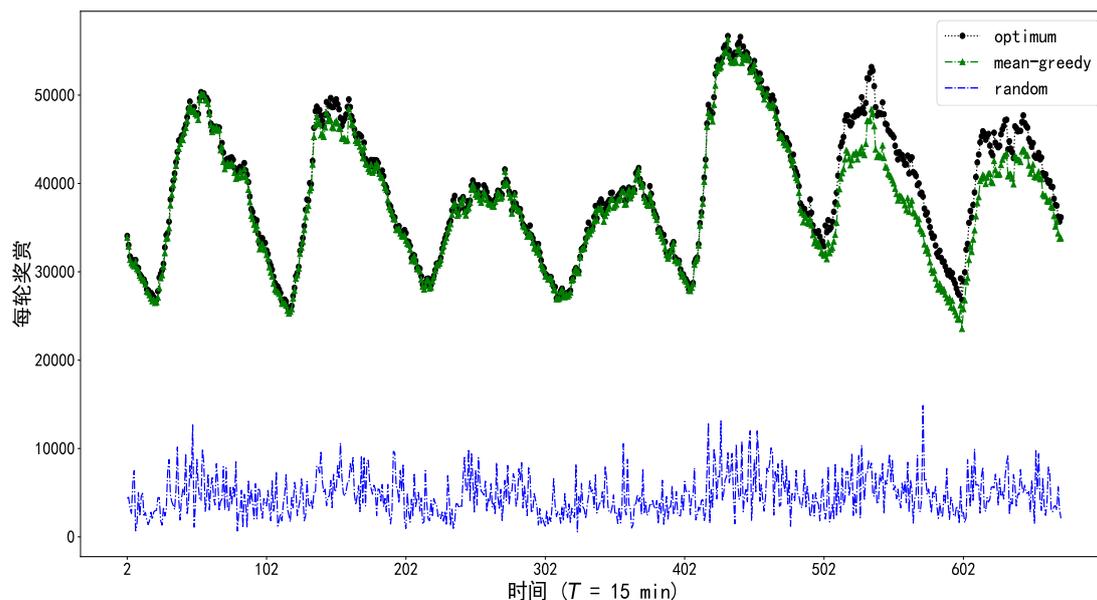


图 3-3 随机选择算法与 mean-greedy 策略在 Geant 网络中的性能对比

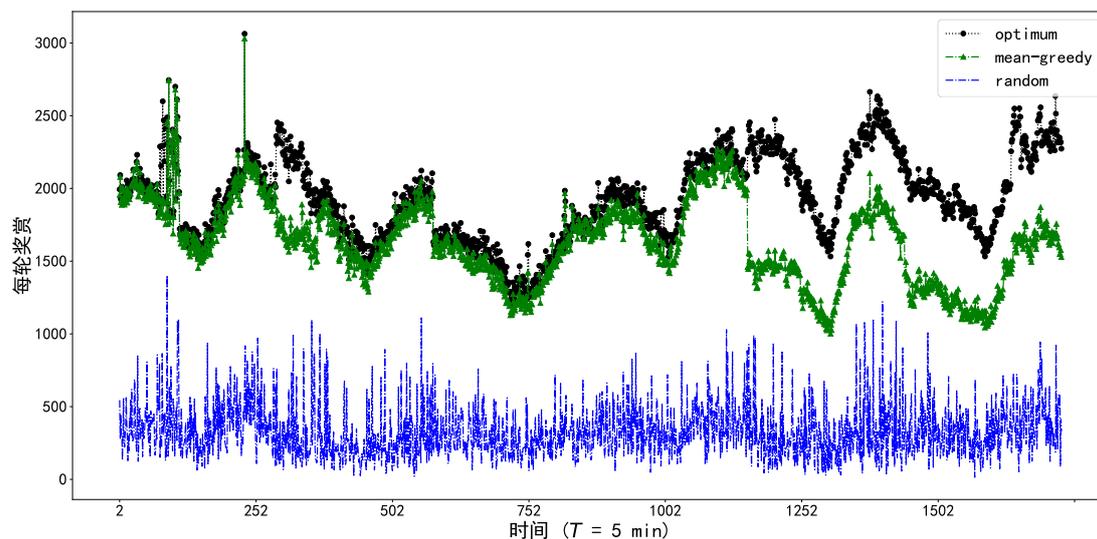


图 3-4 随机选择算法与 mean-greedy 策略在 Abilene 网络中的性能对比

接着，本文对比了所设计的贪婪摇臂选取算法中不同摇臂选取根据的性能。最初，本文设置 $w = 0.01, d = 20$ 。如图 3-5，图 3-6 所示，由于 weighted-greedy 和 sliding window-greedy 策略更多地关注最近的历史随机奖赏信息，因此，这两种策略能更好地适应摇臂奖赏分布的变化（图 3-5 的 500 回合后，图 3-6 的 1150 回合后）。

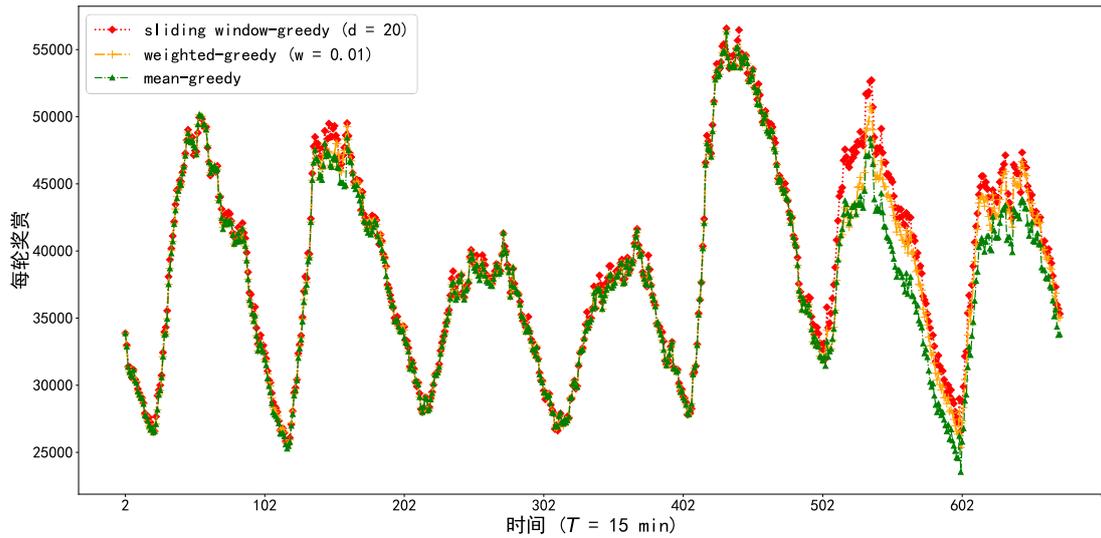


图 3-5 各摇臂选取策略在 Geant 网络中的性能对比

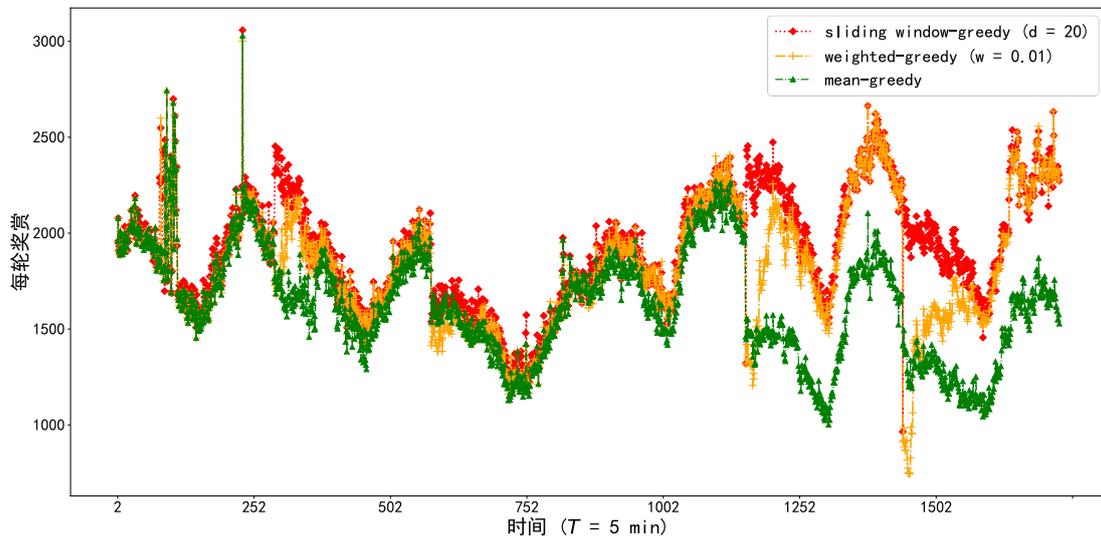


图 3-6 各摇臂选取策略在 Abilene 网络中的性能对比

然而，本文同时也注意到，在某些时候（如图 3-6 所示，在 Abilene 网络中的第 1445 个回合），某些策略的性能会大幅度地下降，因此，本文还探索了权重参数和滑动窗口与算法性能之间的关系。

本文尝试逐渐增大 w 和减少 d 以达到更加注重近期历史随机奖赏数据的目的，并观察各算法性能的变化。有趣的是，至少在本文展开的仿真实验中，当越来越关注近期获取的随机奖赏数据，算法性能就会越来越好，如图 3-7，图 3-8，图 3-9 和图 3-10 所示。对 weight-greedy 策略及 $\text{sliding window-greedy}$ 策略而言，当 w 与 d 变化到一定程度时，算法的性能就会达到最优。最后，在接下来的对比实验中，本文把 w 设置为 0.9，把 d 设置成 1。

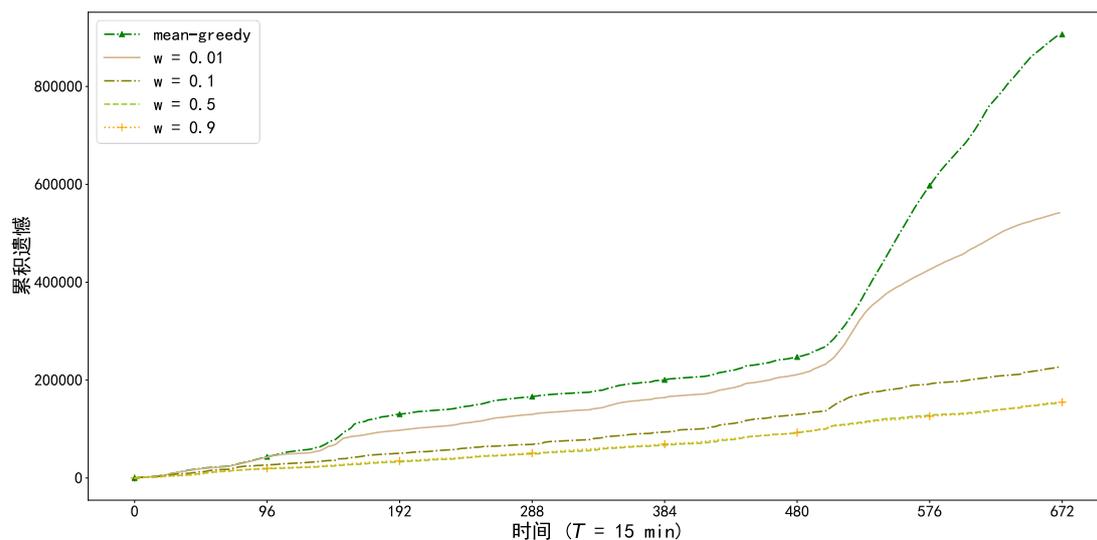


图 3-7 不同权重值在 Geant 网络中的性能对比

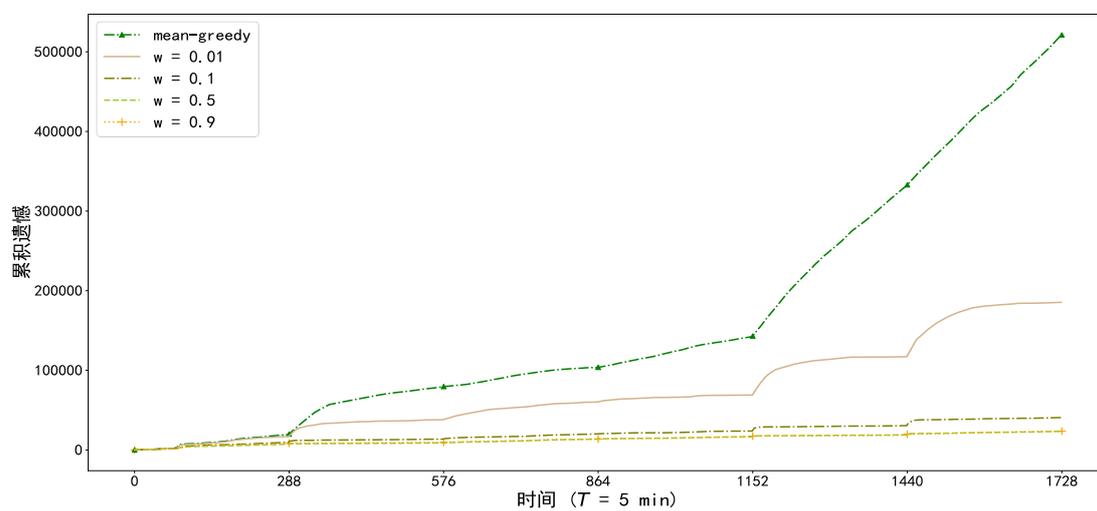


图 3-8 不同权重值在 Abilene 网络中的性能对比

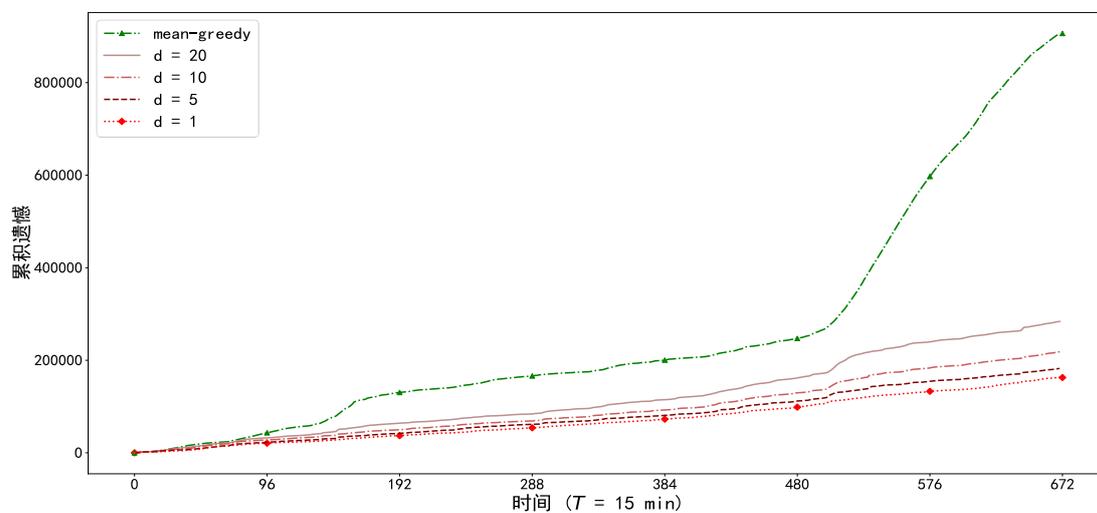


图 3-9 不同滑动窗口值在 Geant 网络中的性能对比

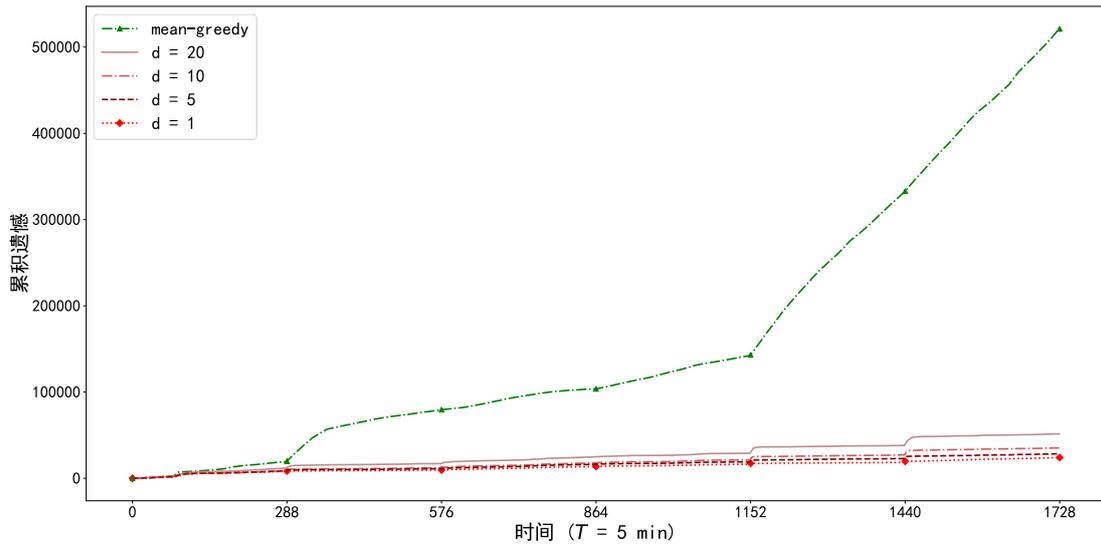


图 3-10 不同滑动窗口值在 Abilene 网络中的性能对比

为了保证公平，对于 SAR 和 Q-SAR 算法，本文把探索开销设计得足够大（在本文的工作中，固定开销设置为摇臂总数的一百倍），以确保两个算法都可以充分地获取各个摇臂的随机奖赏信息。然后，本文在验证数据集上对比了不同算法的性能。如图 3-11，图 3-12 所示，SAR, Q-SAR 算法的性能均不是很理想，其中 SAR 算法的累计遗憾非常大。尽管 Q-SAR 的性能远好于 SAR，该算法的累计遗憾仍高于本文所设计的贪婪摇臂选取算法。同时，本文也注意到，sliding window-greedy 策略的性能要远远优于其他算法。

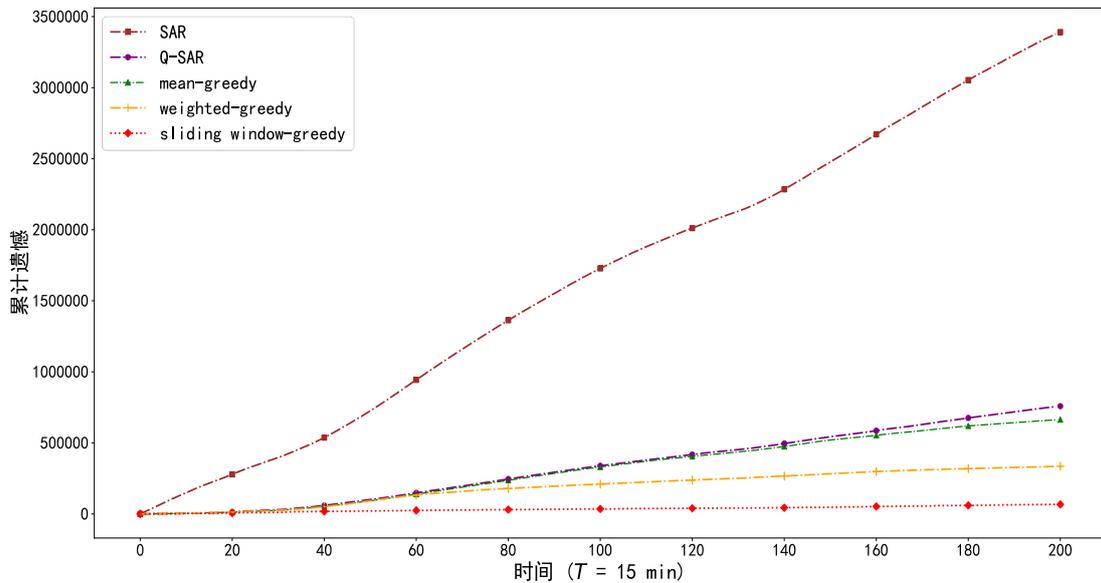


图 3-11 各算法在 Geant 网络中的性能对比

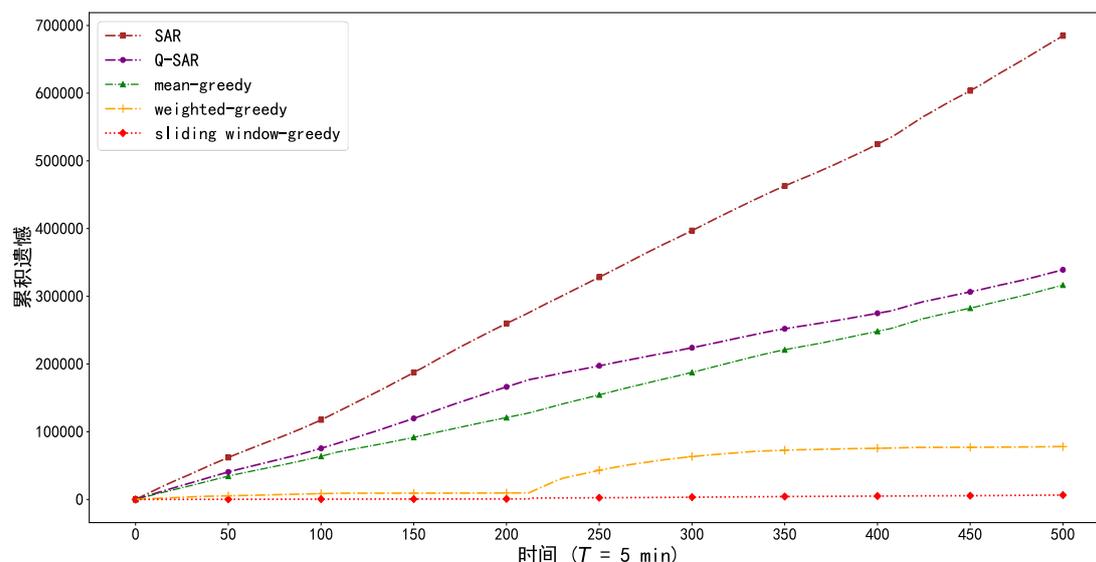


图 3-12 各算法在 Abilene 网络中的性能对比

此外, 本文利用 $R_{S^*, S}$ (在 Geant 网络中, $t = 201$, 在 Abilene 网络中, $t = 501$) 指标来量化分析各个算法的性能。本文发现, sliding window-greedy 策略所选出的超臂与理论最优超臂有超过 99% 的相似度。在 Abilene 拓扑中, sliding window-greedy 的性能要比 SAR 和 Q-SAR 分别高 65.42% 和 32.05%。在 Geant 拓扑中, 上述数字则变化为 40.48% 和 8.43%, 上述结果见表 3-5 所示。

表 3-5 贪婪摇臂选取算法与其他方法的对比分析

网络拓扑	方法	$R_{S^*, S}$
Abilene	SAR	33.96%
	Q-SAR	67.33%
	mean-greedy	69.44%
	weighted-greedy	92.47%
	sliding window-greedy	99.38%
Geant	SAR	58.70%
	Q-SAR	90.75%
	mean-greedy	91.92%
	weighted-greedy	95.92%
	sliding window-greedy	99.18%

值得注意的是, 对于 SAR 和 Q-SAR 两种算法, 在探索结束后, 它们不会更新每个摇臂的评价, 这意味着它们不会带来额外的成本。因此, SAR 和 Q-SAR 都假设各摇臂的奖励分布不具备时变性。然而, 在许多情况下, 上述假设并不成立。对于本文提出的贪婪摇臂选取算法, 为了适应奖励的时间变化, 该算法在每一轮之后更新每个摇臂的评价。这必然会带来额外的开销, 但如表 3-3 所示, 一般而言, 额外的开销是可以接受的。

3.5 本章小结

本章节研究了 SDN 中 top- K 关键流的动态识别问题。本章利用 CMAB 模型对问题进行定义，并结合 SDN 网络的全局管理能力对模型进行了修订（分别增加了一个假设与一个限制条件）。此外，本章基于贪婪思想提出了一个贪婪的摇臂选取算法并设计了三个摇臂选取依据，最后还对所提算法的在单个回合内的时间复杂度和空间复杂度进行了分析。

为验证所提算法的有效性，本章不仅搜集了两个真实网络拓扑的带宽需求数据作为仿真实验的数据支撑，而且将所提算法与当前先进的 top- K 摇臂识别算法进行了性能的分析比较。基于真实网络流量数据的实验结果证明了贪婪思想的有效性。

4 面向关键 SFC 的迁移策略

4.1 引言

由第 3 章的分析可知, 服务队列中的极少数 SFC 请求会占据大量的物理资源(带宽资源, 物理节点的计算资源等), 且这部分 SFC 是可预知的。然而, 随着服务队列中 SFC 的资源调整及生命周期的演变, 在一段时间后, 物理网络中一些具备网络功能虚拟化功能的节点和物理链路可能会出现过负载的风险, 这会对网络运营商的长期收益及后续的 SFC 请求造成潜在的负面影响(如, 更低的服务接收率, 更低的物理资源利用率, 等)。为消除这些潜在的负面影响, 运营商需要对服务队列中的 SFC 进行迁移, 图 4-1 阐述了 SFC 迁移的过程。在初始状态下, 物理网络中具备 VNF 能力的节点和链路出现了过负载的情况。此时服务队列中一共有三个 SFC 请求, 且 SFC 1 和 SFC 2 的资源需求均小于 SFC 3 的资源需求。对运营商而言, 资源需求小的业务更容易找到对应的资源映射方案, 因此在这个场景中, 本文考虑先对 SFC 1 或 SFC 2 进行迁移, 本文以对 SFC 2 进行资源迁移为例进行说明。如图 4-1 所示, SFC 2 的传输路径可以由 $S1 \rightarrow V1 \rightarrow V2 \rightarrow V4 \rightarrow S5$ 迁移至 $S1 \rightarrow V1 \rightarrow V3 \rightarrow S4 \rightarrow V4 \rightarrow S5$, 且 SFC 2 的 VNF 1, VNF 2 可以由 V2, V4 分别迁移至 V1, V3。尽管对 SFC 进行迁移会带来一定的开销, 但是迁移后可以使得物理资源分布变得更加均衡, 以此对运营商及后续的 SFC 请求带来积极的反馈。

目前, 部分研究人员也展开了面向 SFC 或网络切片的迁移工作, 然而, 大部分工作主要是基于保障终端业务的移动性展开的^[28-31], 仅有少部分研究有考虑到物理节点或链路的过负载问题^[33], 且他们没有综合考虑业务的服务周期、迁移开销以及计算成本等问题。本章主要探索了在核心网中由物理资源分布不均衡而引起的 SFC 迁移问题, 与现有工作的对比, 本文的研究动机及所用方法理论等如表 4-1 所示。

本文首先利用标准差来表示物理网络中的资源分布情况, 然后在综合考虑迁移开销以及一些迁移约束(如, 资源上限约束, 传输时延约束, 等)的前提下把 SFC 迁移问题定义成整数线性规划模型。受到相关研究工作^[29,33]的启发, 本文首先设计了一个保守的 SFC 迁移策略来弱化物理资源分布的不均衡。然而, 所提的相关工作过于关注迁移开销^[29,33], 因此仅有部分 VNFs(如, SFC 中的最后一个 VNF)会被迁移, 这会导致迁移后, 物理网络中仍会存在一些过负荷的节点和链路。为了克服上述不足, 本文在保守 SFC 迁移策略的基础上提出了一个激进的 SFC 迁移策略, 该策略会对整个 SFC 进行迁移, 以最大程度地消除物理资源分布的不均衡。在完成迁移后, 本文还使用目前先进的 SFC 资源分配算法,

BestFit^[80,81]和 CN^[20]来为后续的 SFC 请求进行资源分配,并利用物理资源利用率,运营商的长期收益,请求接收率等指标来量化分析迁移策略的性能。

表 4-1 本文工作与现有面向 SFC 迁移工作的对比分析

文献作者	应用场景	迁移动机	迁移目标	方法理论	是否考虑迁移开销	计算开销	迁移规模
Mada <i>et al.</i> ^[28]	边缘云与核心云的混合环境	终端用户的移动性	节约已分配的物理资源	ILP 建模, 使用 Gurobi 优化器求解	否	巨大	未说明
Zhao <i>et al.</i> ^[29]	雾计算环境	终端用户的移动性	保证用户的 QoE	ILP 建模, 启发式算法求解	是	可接受	SFC 的部分 VNFs
Addad <i>et al.</i> ^[30]	边缘云与核心云的混合环境	终端用户的移动性	节约已分配的物理资源	深度强化学习方法求解	否	巨大 (训练开销)	未说明
Wei <i>et al.</i> ^[31]	核心云环境	业务资源需求的时变性	保证用户的 QoE	马尔可夫决策模型建模, 深度强化学习方法求解	是	巨大 (训练开销)	未说明
Zhang <i>et al.</i> ^[33]	边缘云与核心云的混合环境	剩余物理资源分布不均衡	释放高负荷节点的资源	ILP 建模, 启发式算法求解	是	可接受	SFC 的最后一个 VNF
本文所提方法	核心云环境	剩余物理资源分布不均衡	使剩余物理资源分布更均衡	ILP 建模, 启发式算法求解	是	可接受	SFC 的全部 VNFs

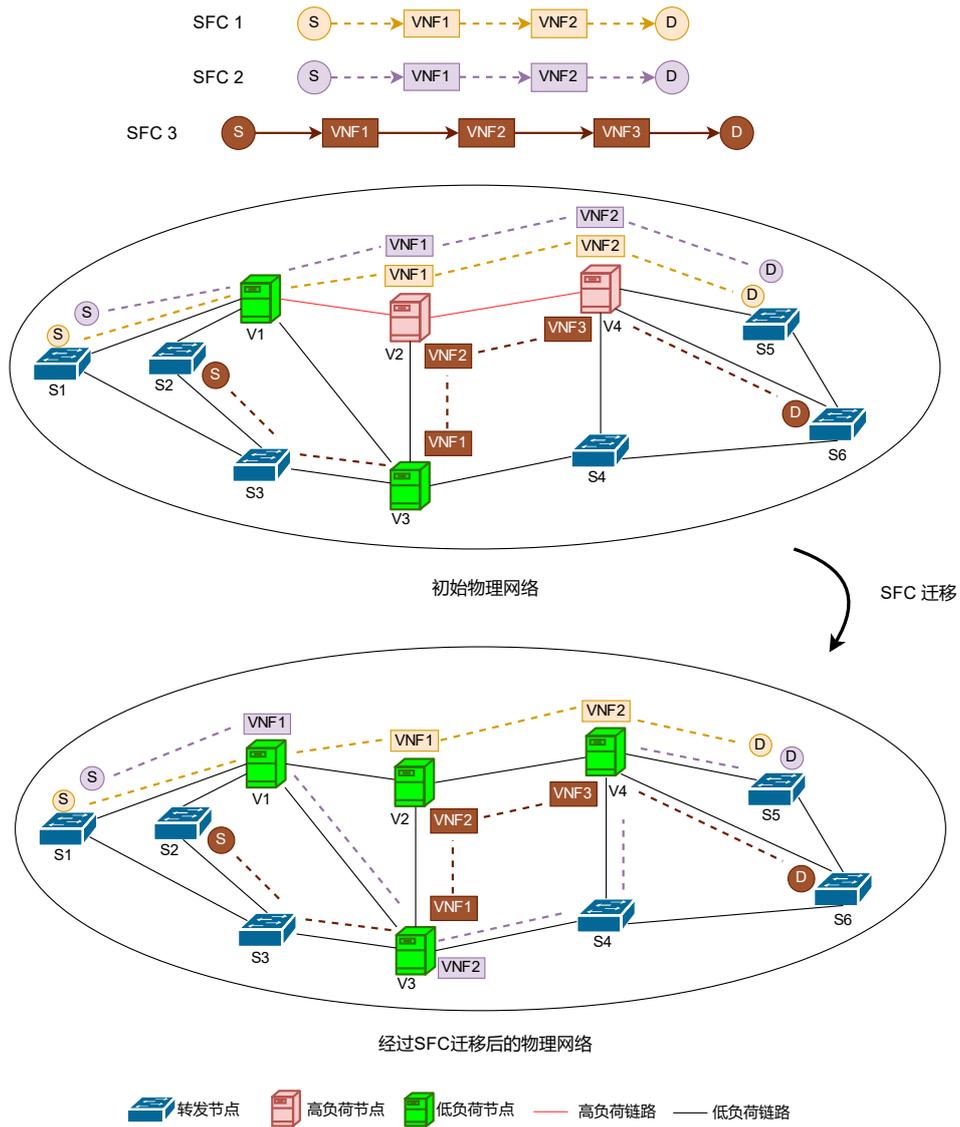


图 4-1 SFC 迁移说明图

4.2 问题陈述

本文考虑了以下的应用场景。在初始时刻，服务队列中有部分关键 SFC 请求，这些 SFC 请求的生命周期是永久的，它们不仅会占据大量的物理资源，且这些请求的初始资源映射方案是随机的。MANO 需要在遵守约束（如，时延约束，带宽资源约束，等）的前提下来迁移这些 SFC 请求以确保物理资源分布的均衡。

由于 SFC 迁移问题的本质是为服务队列中的 SFC 请求计算新的资源映射方案，因此本文先阐述物理网络模型和 SFC 请求模型，然后再对迁移问题进行数学建模。为了方便参考，表 4-2 总结了本章所使用到的符号。

表 4-2 SFC 迁移问题所使用的符号总结

符号	定义
N, E	物理节点（包括转发节点 N_F 和具备 VNF 功能的节点 N_V ）和物理链路
$\tau_{i,j}, B_{i,j}$	链路 (i, j) 的时延以及带宽资源
C_i	具备 VNF 功能的节点 i 的计算资源
s_r, d_r, τ_r	SFC 请求 r 的源节点，目的节点以及时延需求
c_r, b_r	单个 VNF 的 CPU 资源需求，SFC 请求 r 的带宽资源需求
SQ	服务队列
\mathcal{F}_r	SFC 请求 r 的 VNFs 的集合
π_m^r	SFC 请求 r 的第 m 个 VNF
P_r	SFC 请求 r 的传输路径
M_r	\mathcal{F}_r 的 VNFs 放置方案
(r, π_m^r)	虚拟节点 π_m^r, π_{m+1}^r 之间的虚拟链路
$h_i(\pi)$	二进制变量，表征节点 i 能否放置 VNF π
$X_{i,r}(\pi_m^r)$	二进制变量，表征 VNF π_m^r 是否放置在物理节点 i 上
$Z_{i,j}(r, \pi_m^r)$	二进制变量，表征虚拟链路 (r, π_m^r) 是否映射在物理链路 (i, j) 上
β	每单位计算资源的带宽资源消耗
W_{cpu}, W_{bw}	每单位计算资源和带宽资源的收益

4.2.1 物理网络模型

与一些相关工作类似，本文把物理网络定义成一个加权有向图 $G = (N, E)$ ，其中， N 和 E 分别表示物理节点的集合和物理链路的集合。一般地，一个物理网络包含一系列普通的转发节点 N_F 和一系列具备网络功能虚拟化的节点 N_V ，因此物理节点可以被进一步地定义为 $N = N_V \cup N_F$ 。具备网络功能虚拟化的节点可以提供一系列的虚拟网络功能，如防火墙，地址转换，而转发节点仅可用于转发数据。应该注意的是，数据转发可以被视为一个特殊的网络功能^[31]，且该网络功能的资源开销相对较小，因此本文忽略数据转发功能带来的资源开销。

此外，本文定义具备网络功能虚拟化的物理节点 $i \in N_V$ 的计算资源上限为 C_i ，而其他的资源（如内存资源，存储资源）是相对充足的。每条物理链路 $(i, j) \in E$ 的带宽资源上限为 $B_{i,j}$ 且时延为 $\tau_{i,j}$ 。

4.2.2 SFC 请求模型

在本文中，一个 SFC 请求 r 被定义为一条线性链，且 r 可以被表示为 $r = (s_r, d_r, \mathcal{F}_r, \tau_r, c_r, b_r)$ 。其中， s_r, d_r 为 SFC 请求 r 的源节点和目的节点， \mathcal{F}_r 表示 r 的 VNF 需求， τ_r, c_r, b_r 则分别表示 r 的时延需求，每个 VNF 计算资源的需求以及带宽需求。如图 4-1 所示，SFC 3 可以被表示为 $r_3 = (S_2, S_6, \{VNF1, VNF2, VNF3\}, \tau_3, c_3, b_3)$ 。

4.2.3 问题定义

接下来，本文利用数学模型来定义 SFC 迁移问题。本文先介绍 MANO 在为服务队列中的 SFC 请求进行资源重分配时需要遵守的约束。

首先是 VNFs 的放置约束，为了方便描述，本文定义了一些二进制变量。 $h_i(\pi)$ 表征物理节点 i 能否提供 VNF π 所需要的服务， $X_{i,r}(\pi_m^r)$ 表征 VNF π_m^r 是否放置在物理节点 i 上，其中 π_m^r 表示 SFC 请求 r 的第 m_{th} 个 VNF。为了确保 VNFs 都被放置在具备网络功能虚拟化能力的物理节点上，本文设置了约束 (4-1)。

$$X_{i,r}(\pi_m^r) \leq h_i(\pi_m^r), \forall r \in SQ, \forall \pi_m^r \in \mathcal{F}_r \quad (4-1)$$

本文参考了部分先进的相关工作^[22,29]，为 VNFs 增加了不可切分的约束，即 SFC 请求的每个 VNF 仅仅能映射在一个具备网络功能虚拟化的物理节点上。因此，本文设置了约束 (4-2)。

$$\sum_i X_{i,r}(\pi_m^r) = 1, \forall r \in SQ, \pi_m^r \in \mathcal{F}_r \quad (4-2)$$

此外，本文还假设每个具备网络功能虚拟化的物理节点仅为每个 SFC 请求 r 提供一次服务，该约束可以被表示为约束 (4-3)：

$$\sum_{\pi_m^r \in \mathcal{F}_r} X_{i,r}(\pi_m^r) \leq 1, \forall i \in N_V, r \in SQ \quad (4-3)$$

然后，本文需要考虑虚拟链路的放置约束。为了避免协调的开销，本文并不允许路径分离。与相关工作^[29]类似，本文定义 P_r 为 SFC 请求 r 的传输路径， M_r 为 \mathcal{F}_r 中 VNFs 的放置方案。每条传输路径都需要遵循约束 (4-4), (4-5)。

$$\tau_r \geq \sum_{(i,j) \in P_r} \tau(i,j), \forall r \in SQ \quad (4-4)$$

$$M_r \in P_r, \forall r \in SQ \quad (4-5)$$

其中，式子 (4-4) 为时延约束，该约束表示传输路径的总时延需要小于 SFC 请求所能容忍的最大时延。式子 (4-5) 则表示，传输路径所经过的物理节点必须能提供对应的 VNF 服务。

最后，本文需要考虑资源上限的约束。类似地，本文定义二进制变量 $Z_{i,j}(r, \pi_m^r)$ 来表征虚拟路径 (r, π_m^r) 是否映射在物理链路 (i, j) 上，其中 (r, π_m^r) 表示虚拟节点 π_m^r, π_{m+1}^r 之间的虚拟路径。特别地， (r, π_0^r) 表示从 s_r 到第一个 VNF 之间的虚拟路径，而 $(r, \pi_{L_r}^r)$ 则表示从最后一个 VNF 到 d_r 之间的虚拟路径。如前所述，本文忽略了转发功能所带来的计算开销，因此，一个具备网络功能虚拟化能力的物理节点 i 的计算资源消耗和一段物理链路 (i, j) 的带宽资源消耗可以表示为式子 (4-6), (4-7):

$$R_C(i) = \sum_r \sum_{\pi_m^r} X_{i,r}(\pi_m^r) c_r \quad (4-6)$$

$$R_B(i, j) = \sum_r \sum_{\pi_m^r} Z_{i,j}(r, \pi_m^r) b_r \quad (4-7)$$

其中， c_r 表示 SFC 请求 r 单个 VNF 所需要的计算资源需求， b_r 则表示 SFC 请求 r 所需要的带宽需求。

与相关工作^[16,82]类似，本文假设在同一个 SFC 请求 r 内，VNFs 的计算资源需求都是一致的。因此物理节点计算资源和连理带宽资源的上限约束可以分别被表示为式子 (4-8), (4-9):

$$R_C(i) \leq C_i, \forall i \in N_V \quad (4-8)$$

$$R_B(i, j) \leq B_{i,j}, \forall (i, j) \in E \quad (4-9)$$

此外，本文使用标准差 σ_G 来衡量物理资源分布的均衡性，其中 σ_G 可以通过式子 (4-10) 来计算。

$$\sigma_G = \sqrt{\frac{1}{NUM(N_V)} \sum_{i \in N_V} (R_C(i) - \overline{R_C})^2} + \sqrt{\frac{1}{NUM(E)} \sum_{(i,j) \in E} (R_B(i,j) - \overline{R_B})^2} \quad (4-10)$$

其中， $NUM(N_V), NUM(E)$ 分别表示具备网络功能虚拟化能力的节点的总数以及物理链路的总数， $\overline{R_C}, \overline{R_B}$ 分别表示 N_V 以及 E 的资源消耗的平均值。如图 4-1 所示，在经过 SFC 迁移后，由于物理网络中不再存在高负荷的节点和链路，因此 σ_G 可以被减小。

在 SFC 迁移的过程中，本文还需要考虑迁移开销。参考部分相关工作^[29,33]，本文对迁移开销 o_r 作如下定义。 o_r 表示一个 SFC 请求 r 在迁移过程中需要重新处理的报文的总和，本文使用 t_r 来表示 SFC 请求 r 在迁移开始时的时刻， t_r' 表示其迁移结束时的时刻，且 $u_r(t)$ 表示该请求在时刻 t 时的发包速率。则，在整个迁移过程中，SFC 请求 r 所产生的需要 r 重处理的报文的总和可以表示为 $\int_{t_r}^{t_r'} u_r(t)$ 。参考相关的研究工作^[33]，本文同样利用 SFC 请求 r 的带宽资源需求 b_r 来表示 $u_r(t)$ 。由于在 SFC 技术中，请求 r 的流量需要按照预先设定的顺序依次通过一系列的 VNFs，因此，拟迁移的 VNF 在 SFC 链中的位置越靠前，对整条 SFC 链

的影响就会越大^[33]。本文使用 $Len(P_r^o, P_r^m)$ 来表示经过迁移后，SFC 请求 r 产生的需要重新处理的报文需要经过多少条虚拟链路，其中 P_r^o 表示 r 原始的传输路径，而 P_r^m 则表示经过迁移后， r 的传输路径。例如，在图 4-1 中，SFC 2 的 P_2^o 为 $(S1 \rightarrow V1 \rightarrow V2 \rightarrow V4 \rightarrow S5)$ ，且 P_2^m 为 $(S1 \rightarrow V1 \rightarrow V3 \rightarrow S4 \rightarrow V4 \rightarrow S5)$ 。在这种情况下， $Len(P_2^o, P_2^m) = 4$ ，因为在经过 V1 后，接下来的链路全都发生了改变。因此，本文用式子 (4-11) 来定义 o_r 。

$$o_r = \int_{t_r}^{t_r'} Len(P_r^o, P_r^m) b_r \quad (4-11)$$

此外，为了计算总的迁移开销 \mathcal{O}_G ，本文需要把服务队列中所有 SFC 请求的迁移开销进行叠加，因此总的迁移开销可以被表示式子 (4-12) 为：

$$\mathcal{O}_G = \sum_{r \in SQ} o_r \quad (4-12)$$

最后，综合考虑网络资源分布的均衡性以及总的迁移开销，本文把 SFC 迁移问题定义为式子 (4-13)。

$$\begin{aligned} & \min \quad \sigma_G + \mathcal{O}_G \\ & \text{s.t.} \quad (4-1, 4-2, 4-3, 4-4, 4-5), (4-9, 4-10) \\ & h_i(\pi_m^r) \in \{0, 1\}, \forall i \in N, r \in SQ, \pi_m^r \in \mathcal{F}_r \\ & X_{i,r}(\pi_m^r) \in \{0, 1\}, \forall i \in N, r \in SQ, \pi_m^r \in \mathcal{F}_r \\ & Z_{i,j}(r, \pi_m^r) \in \{0, 1\}, \forall (i, j) \in E, r \in SQ, \pi_m^r \in \mathcal{F}_r \end{aligned} \quad (4-13)$$

4.3 问题求解

本节将分别介绍受到相关工作^[29,33]启发而设计出的保守迁移策略以及本文所提出的激进的迁移策略。

4.3.1 一种保守的迁移策略

为了尽可能地减少 SFC 迁移对用户的服务质量造成的影响，一些研究人员^[29,33]倾向于仅迁移 SFC 链的最后一个 VNF。受到这个想法的启发，本文设计了一个保守的迁移策略，该策略通过以下步骤执行。

1. 获取资源需求低的 SFC 请求。由于一个 SFC 请求所需要的资源越小，为其找到可行的资源映射方案的可能性就越大，因此本文优先对资源需求低的 SFC 请求进行迁移。在本文中，由于单位计算资源的利润更高，因此，一个 SFC 请求的资源需求 \mathcal{Y}_r 可以通过式子 (4-14) 来计算。在进行迁移之前，需要先把服务队列中的 SFC 请求按照 \mathcal{Y}_r 从小到大进行排列，且优先选择 \mathcal{Y}_r 最小的请求进行迁移。

$$\mathcal{Y}_r = \sum_{\pi \in r} c_r \quad (4-14)$$

2. 资源回收。在进行 SFC 迁移前, MANO 需要先回收已分配给请求 r 的各种资源, 包括节点的计算资源, 链路的带宽资源。
3. 确定原资源分配方案中稳定的部分。由于保守的迁移策略仅迁移 SFC 链的最后一个 VNF, 因此 MANO 需要先确定原资源分配方案中稳定的部分。本文使用 m_r^s 来表示 M_r^o 中的稳定部分, 其中 M_r^o 表示 \mathcal{F}_r 的原始放置方案, 而 p_r^s 则表示 P_r^o 中的稳定部分。然后, 本文使用 L_r 来表示 \mathcal{F}_r 的长度, $m_{L_r}^r$ 表示 M_r^o 中的最后一个元素, m_{pen}^r 表示 M_r^o 中的倒数第二个元素。显然, $m_r^s = M_r - m_{L_r}^r$, $p_r^s = (s_r \rightarrow \dots \rightarrow m_{pen}^r)$ 。
4. 重计算迁移部分的资源分配方案。该步骤需要解决以下两个问题。(a) \mathcal{F}_r 中最后一个 VNF 需要被迁移至哪个物理节点上。(b) 如何选择 m_{pen}^r 到 d_r 的路径。本文使用 $\pi_{L_r}^r$ 来表示 \mathcal{F}_r 中的最后一个 VNF, m_{pen}^r 表示对 VNF $\pi_{L_r}^r$ 的新的资源映射方案, p_r^{re} 表示从 m_{pen}^r 到 d_r 的路径的资源映射方案。为了避免环路的情况, 首先需要从物理图 G 中移除 p_r^s 的元素 (为了计算的需要, m_{pen}^r 不会被移除), 并得到一个经过剪枝的物理图 G_{P_1} 。然后, 需要在遵循约束 (4-1), (4-3) 的前提下计算可放置 $\pi_{L_r}^r$ 的物理节点集合 \mathcal{N}_r 。一般地, $\mathcal{N}_r = N_V^i - (N_V^i \cap m_r^s)$, 其中 N_V^i 表示在 G_{P_1} 中具备网络功能虚拟化能力的物理节点的集合。随后, 本文选择把 $\pi_{L_r}^r$ 放置在一个能为 $\pi_{L_r}^r$ 提供服务且剩余计算资源最多的物理节点 i_{most} 上。接下来, 本文移除了 G_{P_1} 中不能满足 SFC 请求 r 的带宽资源需求的链路, 从而得到另一个剪枝图 G_{P_2} , 且利用深度优先遍历算法 (Depth First Search, DFS)^[83] 来计算出从 m_{pen}^r 到 d_r 的所有可行路径。然后需要在遵守约束 (4-4), (4-5) 的前提下在所有的可行路径中构造出 p_r^{re} 的可行解的集合 \mathcal{P}_r , 并把从 m_{pen}^r 到 d_r 的虚拟链路映射在 \mathcal{P}_r 中剩余带宽资源标准差最小的路径 p_{least} 上。经过上述步骤的处理, 可以得到 $m_r^{re} = i_{most} \cdot P_r^{re} = p_{least}$ 。
5. 资源重映射。当 MANO 得到 $p_r^s, p_r^{re}, m_r^s, m_r^{re}$ 后, 它就能确定 SFC 请求 r 的新资源分配方案。然后, MANO 需要根据 P_r^m, M_r^m 来为 r 分配资源。

算法 4-1 描述了保守迁移方案的处理过程, 算法 4-2 被用于执行步骤 2, 而算法 4-3 则用于执行步骤 4, 算法 4-1 的第 8 到 13 行用于执行步骤 5。

在图 4-1 中, 由于 SFC 1, 2 的资源需求比较小, 因此 MANO 会优先对他们进行迁移, 本文使用 SFC 2 的迁移过程作为例子来对保守的迁移策略进行阐述。

首先, MANO 需要回收已分配给 SFC 2 的物理资源, 如图 4-2 (a) 所示。随后, MANO 需要确定 SFC 2 原始资源分配方案中的稳定部分, 如图 4-2 (b) 所示, 其中 m_2^s 为(S1→V1→V2), 而 p_2^s 为{V2}。紧接着, MANO 需要确定 \mathcal{F}_2 中最后一个 VNF 应该被迁移至哪里, 且从 m_{pen}^2 到 d_2 的虚拟链路应该被放置在哪里。为了解决上述问题, MONO 首先移除物理图中的部分节点来避免环路, 如图 4-2 (c) 所示。由于 V3 剩余的计算资源最多且能满足约束 (4-1), 因此 MANO 把 VNF2 迁移至 V3, 如图 4-2 (d) 所示。此后, MANO 需要在剪枝图中计算 m_2^{re} 到 V3 的可行路径。在遵守约束(4-5),(4-6) 的前提下, 由于路径(V2→V3→S4→S6→S5)剩余资源的标准差最小, MANO 会把 p_2^{re} 映射在其上。最后, MANO 会根据 $m_2^s \cup m_2^{re}, p_2^s \cup p_2^{re}$ 来为 SFC 2 重新分配资源, 如图 4-2 (e) 所示。

接下来, 本文需要对保守迁移策略的计算复杂度进行分析。首先, 对所有的请求进行排序(算法 4-1 的第 2 行)的计算开销为 $O(|SQ| \log_2 |SQ|)$ 。随后, 第一个 **while** 循环需要迭代 $|SQ|$ 次。对算法 4-2 的处理过程, 它的计算开销为 $O(|E| + |N_V|)$, 对算法 4-3 的处理过程, 它的第一个 **for** 循环需要迭代 $|p_r^s| - 1$ 次, 第二个 **for** 循环需要迭代 $|P_r^c|$ 次。此外, 对于 DFS 算法, 一条简单路径的计算开销为 $O(|E| + |N|)$, 则其总的开销为 $O(|P|(|E| + |N|))$, 其中 P, N, E 分别表述从节点 s 到 d 的可行路径的集合, 图中链路的集合, 和图中节点的集合。本文令 E_r, N_r 分别表示图 G_{p_2} 中的链路集合和节点集合, 且令 $E_{con} = \max_{r \in [1, |SQ|]} |E_r|, N_{con} = \max_{r \in [1, |SQ|]} |N_r|, A_{con} = \max_{r \in [1, |SQ|]} (|p_r^s| - 1), B_{con} = \max_{r \in [1, |SQ|]} |P_r^c|$ 。因此, 算法 4-2 的计算开销为 $O(A_{con} + B_{con}(E_{con} + N_{con}) + B_{con})$ 。对于算法 4-1 的最后两个 **for** 循环, 它们分别需要迭代 $|M_r^m|, |P_r^m|$ 次。由于 $B_{con}(E_{con} + N_{con})$ 小于 $\log_2 |SQ|, A_{con}, (|E| + |N|), |M_r^m|, |P_r^m|$, 因此保守迁移策略的时间复杂度为 $O(|SQ| \log_2 |SQ| + |SQ|(|E| + |N_V|) + A_{con} + B_{con}(E_{con} + N_{con}) + B_{con} + B_{con} + |M_r^m| + |P_r^m|)$
 $= O(|SQ| B_{con}(E_{con} + N_{con}))$

算法 4-1 保守的 SFC 迁移算法

输入： 初始的物理网络 G ，服务队列 SQ 中 SFC 请求的信息，包括 $c_r, b_r, \mathcal{F}_r, P_r^o, M_r^o, s_r, d_r, \tau_r$ 。

输出： 经过 SFC 迁移后的物理网络 G' ，迁移开销 \mathcal{O}_G 。

- 1: 初始化：令 $\mathcal{O}_G = 0$ ；
 - 2: 根据 \mathcal{Y}_r 对服务队列中的所有请求进行升序排序；
 - 3: **While** $SQ \neq \emptyset$ **do**
 - 4: 选择 \mathcal{Y}_r 最小的请求，调用算法 4-2，然后令 $G = G_{rec}, M_r^m = \{\}, p_r^m = \{\}$ ；
 - 5: 令 $m_r^s = M_r - m_{L_r}^r, p_r^s = (s_r, \dots, m_{pen}^r)$ ；
 - 6: 调用算法 4-3，然后获得 p_r^{re}, m_r^{re} ；
 - 7: 令 $M_r^m = m_r^s \cup m_r^{re}, P_r^m = p_r^s \cup p_r^{re}$ ；
 - 8: **for** $i \in M_r^m$ **do**
 - 9: $C_i = C_i - c_r$ ；
 - 10: **end for**
 - 11: **for** $(i, j) \in P_r^m$ **do**
 - 12: $B_{i,j} = B_{i,j} - b_r$ ；
 - 13: **end for**
 - 14: 根据式子 (4-11) 计算请求 r 的迁移开销 o_r ；
 - 15: $\mathcal{O}_G = \mathcal{O}_G + o_r, SQ = SQ - r$ ，迁移下一个 SFC 请求；
 - 16: **end while**
 - 17: **return** G', \mathcal{O}_G
-

算法 4-2 资源回收算法

 输入: $G, b_r, c_r, P_r^o, M_r^o$ 。

 输出: G_{rec} (经过资源回收后的物理图)

```

1: for  $(i, j) \in E$  do
2:   if  $(i, j) \in P_r^o$  then
3:      $B_{i,j} = B_{i,j} + b_r$ ;
4:   end if
5: end for
6: for  $i \in N_V$  do
7:   if  $i \in M_r^o$  then
8:      $C_i = C_i + c_r$ ;
9:   end if
10: end for
11: return  $G_{rec}$ 
    
```

算法 4-3 资源分配方案重计算算法

 输入: $G, b_r, c_r, p_r^s, m_r^s, d_r$ 。

 输出: p_r^m, m_r^{re} 。

```

1: 初始化: 令  $s' = p_r^s$ , 可行传输路径集合  $\mathcal{P}_r = \{\}$ ,  $\mathcal{N}_r = N_V' - (N_V' \cap m_r^s)$ ;
2: for  $i \in (p_r^s - s_r')$  do
3:   移除节点  $i$  及其直连的链路;
4: end for
5: 移除图中不能满足请求  $r$  带宽需求的链路;
6: 选取  $\mathcal{N}_r$  中剩余物理资源最多的物理节点  $i_{most}$ , 令  $m_r^{re} = i_{most}$ ;
7: 结合  $G, s', d_r$ , 调用 DFS 算法, 然后获取可行路径的集合  $P_r^c$ ;
8: for  $p \in P_r^c$  do
9:   if  $(p \cup p_r^s)$  能满足约束 4-4,4-5, do
10:     $\mathcal{P}_r = \mathcal{P}_r + (p \cup p_r^s)$ ;
11:   end if
12: end for
13: 在  $\mathcal{P}_r$  中选择剩余带宽资源标准差最小的元素  $p$ , 令  $p_r^m = p$ ;
14: return  $p_r^m, m_r^{re}$ 
    
```

面向关键 SFC 的迁移策略

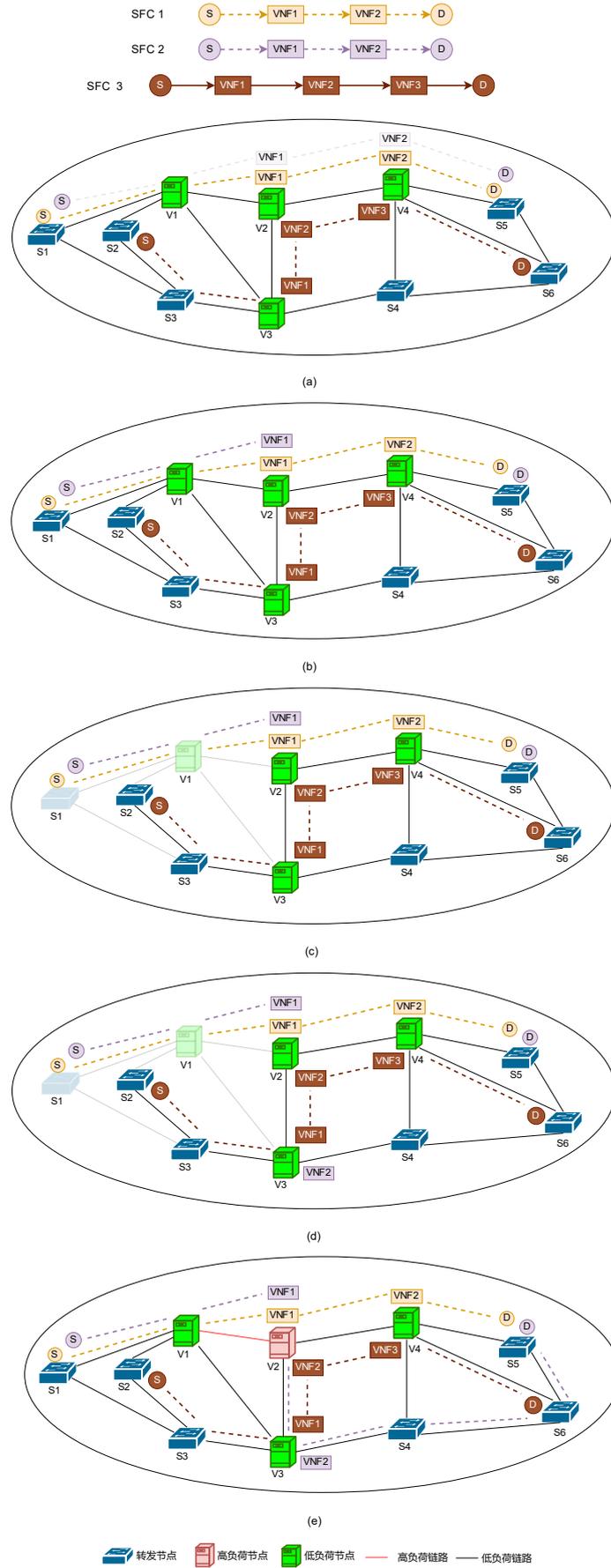


图 4-2 保守 SFC 迁移策略说明图

4.3.2 一种激进的迁移策略

为了尽可能地减少迁移开销，保守的迁移策略仅迁移 SFC 链中的最后一个 VNF，因此在迁移结束后，物理网络中仍可能存在过负载的节点或链路，如图 4-2 (e) 所示。为了让物理资源分布得更加均衡，本文提出了一种激进的迁移策略，该策略按照以下步骤来迁移 SFC 链上的全部 VNFs。

1. 获取资源需求低的 SFC 请求。与保守的迁移策略一样，激进的迁移策略也是优先迁移资源需求低的 SFC 请求。
2. 资源回收。在迁移开始前，MANO 同样需要对已分配出去的资源进行回收。
3. 计算新的资源分配方案。为了解决 SFC 迁移问题，本文把它划分成 VNF 迁移问题以及虚拟路径迁移问题。对于 VNF 迁移问题，本文贪婪地把 VNFs 迁移至剩余计算资源多的物理节点上。对于虚拟路径迁移问题，本文首先移除物理网络中不满足带宽约束的物理链路以此得到一个剪枝的物理网络 G_{p_1} 。然后在 G_{p_1} 中利用 DFS 算法计算出 s_r 到 d_r 的所有满足约束 (4-4), (4-5) 的可行路径 \mathcal{P}_r 。最终，本文把 P_r^m 放置在 \mathcal{P}_r 中能使剩余带宽资源标准差最小的路径 p 上。
4. 资源重映射。在经过步骤 3 后，MANO 需要把拟分配给 SFC 请求的资源重新映射在新的物理节点及物理链路上。

算法 4-4 描述了激进的迁移策略的处理过程。其中，第 5 到 16 行被用于计算 SFC 请求 r 的新的资源分配方案 P_r^m, M_r^m 。第 17 到 22 行被用于执行步骤 4。同样地，本文用图 4-1 中 SFC 2 的迁移过程来阐述所提的激进的迁移策略。如图 4-3 (a) 所示，MANO 需要先回收已分配给 SFC 2 的物理资源。然后，MANO 把 \mathcal{F}_2 中的元素放置在剩余计算资源较多的物理节点上，因此 $M_2^m = \{V1, V2\}$ ，如图 4-3 (b) 所示。紧接着，MANO 先移除物理网络中不能满足带宽约束的物理链路，然后在物理网络中计算从 s_2 到 d_2 的所有可行路径。由于路径(S1→V1→V3→S4→S6→S5)的剩余物理带宽资源的标准差是最小的，因此 MANO 会把虚拟路径放置在其上。最终，MANO 会根据 M_2^m, P_2^m 来进行资源的重映射，如图 4-3 (c) 所示。值得注意的是，在经过 SFC 迁移后，物理网络中不再有过负荷的物理节点和链路，因此激进的迁移策略能使得物理资源的分布更加均衡。

类似地，本文分析所提激进的迁移策略的时间复杂度。同样地，排序过程的时间开销为 $O(|SQ| \log_2 |SQ|)$ ，第 1 个 **while** 循环需要迭代 $|SQ|$ 次，算法 4-2 的计算开销为 $O(|E| + |N_v|)$ 。第 1, 2, 3, 4 个 **for** 循环分别需要迭代

$|\mathcal{F}_r|, |P_r^c|, |P_r^m|, |M_r^c|$ 次。然后, 本文使用 E_r, N_r 来代表物理图 G_{p_i} 中的链路和节点的集合, 且令 $\mathcal{F} = \max_{r \in [1, |SQ|]} |\mathcal{F}_r|, E_{agg} = \max_{r \in [1, |SQ|]} |E_r|, N_{agg} = \max_{r \in [1, |SQ|]} |N_r|$ 及 $B_{agg} = \max_{r \in [1, |SQ|]} |P_r^c|$ 。因此, 激进的迁移策略的时间复杂度为 $O(|SQ| \log_2 |SQ| + |SQ| (\mathcal{F} + B_{agg} (E_{agg} + N_{agg}) + B_{agg} + |P_r^m| + |M_r^m|)) = O(|SQ| B_{agg} (E_{agg} + N_{agg}))$ 。

值得注意的是, 在执行 DFS 算法之前, 保守的迁移策略从物理网络中移除了更多的物理节点和链路, 因此激进的迁移策略的时间复杂度会略高于保守的迁移策略。

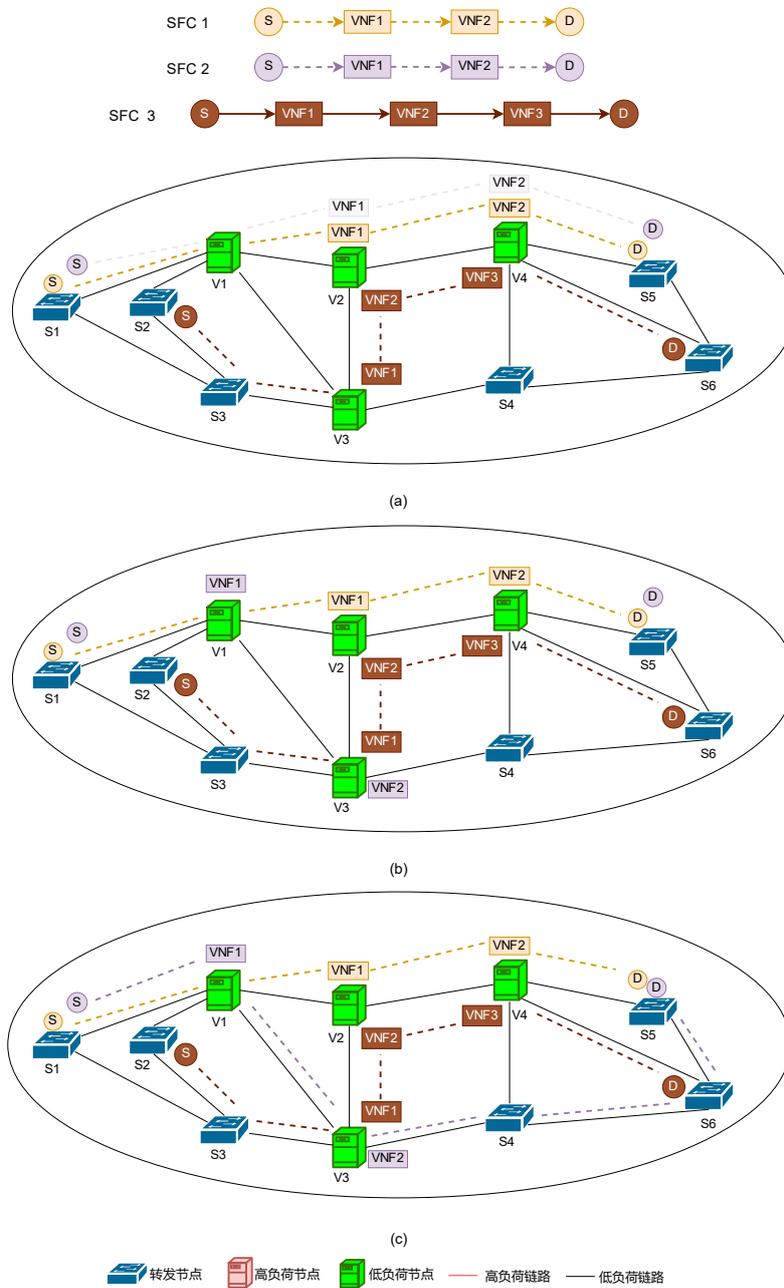


图 4-3 激进 SFC 迁移策略说明图

算法 4-4 激进的 SFC 迁移算法

输入： 初始的物理网络 G ，服务队列 SQ 中 SFC 请求的信息，包括 $c_r, b_r, \mathcal{F}_r, P_r^o, M_r^o, s_r, d_r, \tau_r$ 。

输出： 经过 SFC 迁移后的物理网络 G' ，迁移开销 \mathcal{O}_G 。

- 1: 初始化：令 $\mathcal{O}_G = 0$ ；
- 2: 根据 \mathcal{Y}_r 对服务队列中的所有请求进行升序排序；
- 3: **While** $SQ \neq \emptyset$ **do**
- 4: 选择 \mathcal{Y}_r 最小的请求，调用算法 4-2，然后令 $G = G_{rec}, M_r^m = \{\}, \mathcal{P}_r = \{\}, \mathcal{N}_r = \mathcal{N}_V$ ；
- 5: **for** $f_i \in \mathcal{F}_r$ **do**
- 6: 从 \mathcal{N}_r 中选择剩余计算资源最多的物理节点 i_{most} ；
- 7: 令 $M_r^m = M_r^m + i_{most}, \mathcal{N}_r = \mathcal{N}_r - i$ ；
- 8: **end for**
- 9: 结合 G, s', d_r ，调用 DFS 算法，然后获取可行路径的集合 P_r^c ；
- 10: **for** $p \in P_r^c$ **do**
- 11: **if** p 满足约束 (4-4), (4-5), **then**
- 12: $\mathcal{P}_r = \mathcal{P}_r + p$ ；
- 13: **end if**
- 14: **end for**
- 15: 在 \mathcal{P}_r 中选择使得剩余带宽资源标准差最小的路径 p ，令 $P_r^m = p$ ；
- 16: **for** $(i, j) \in P_r^m$ **do**
- 17: $B_{i,j} = B_{i,j} - b_r$ ；
- 18: **end for**
- 19: **for** $i \in M_r^m$ **do**
- 20: $C_i = C_i - c_r$ ；
- 21: **end for**
- 22: 根据式子 (4-11) 计算请求 r 的迁移开销 o_r ；
- 23: $\mathcal{O}_G = \mathcal{O}_G + o_r, SQ = SQ - r$ ，迁移下一个 SFC 请求；
- 24: **end while**
- 25: **return** G', \mathcal{O}_G 。

4.4 仿真与性能分析

本节证明了所提迁移策略的有效性。首先，本节介绍了仿真实验拓扑，原始队列中 SFC 请求及后续 SFC 请求的具体参数，然后介绍了两个先进的资源映射算法，并设置了一系列的评价指标。最后，本节对比分析了保守迁移策略与激进迁移策略的性能差异。

本章所提算法的运行环境如下：操作平台: Windows 11; 硬件平台: Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz; 编程语言: Python 3.6; 具体算法实现如算法 4-1 至 4-4 所示。此外，本章所设计的仿真实验、后续 SFC 请求的资源分配算法和评价指标具体内容如章节 4.4.1, 4.4.2 和 4.4.3 所示。

4.4.1 实验设置

本文构建了一个被广泛应用于网络切片及 SFC 研究的网络拓扑^[31,84]来用于仿真实验，如图 4-4 所示，且该拓扑的具体参数信息如表 4-3 所示。在物理网络中，一共有 7 个具备网络功能虚拟化能力的物理节点，这些节点可以提供各种各样的网络服务。当前，本文并不关注不同的物理节点及物理链路之间的资源上限的差异，因此，具备网络功能虚拟化能力的物理节点的资源上限被设置为 10000。由于一些研究人员认为，每处理一个单位的流量需要一个单位的计算能力^[21]，因此本文把物理链路的资源上限也设置为 10000。此外，物理链路的时延为 1ms，且对于任意 SFC 请求，本文假设其迁移过程均可以在一个时间窗内完成。

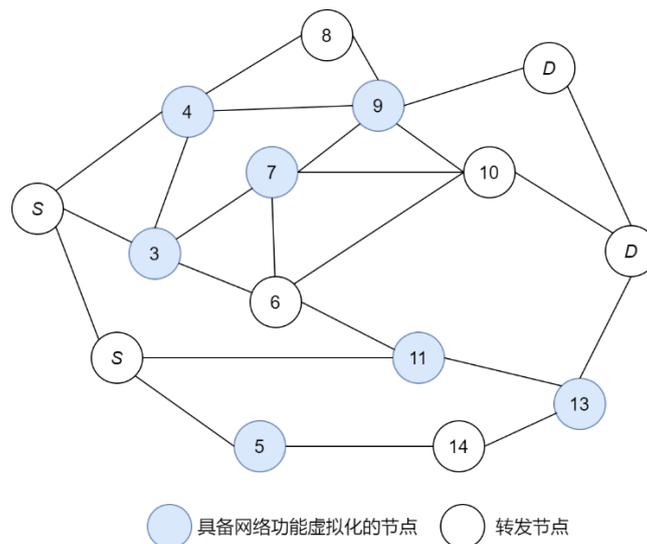


图 4-4 SFC 迁移仿真实验拓扑

表 4-3 SFC 迁移仿真实验拓扑参数

参数	数值
节点数	15
链路数	27
能提供网络功能的节点数	7
β	7
提供网络功能的物理节点的资源上限	10000
物理链路的资源上限	10000
链路时延	1ms

不同类型的 SFC 请求的参数如表 4-4 所示。在现实的应用场景中，会有永久性的业务及临时性的业务^[85,86]。根据第 3 章的分析，网络中的大部分流量是由小部分的业务流产生的，并且这部分业务流是相对稳定的。因此，本文设计了一种永久性的 SFC 请求，这部分请求会占据较多的物理资源，但是相对地，该类型 SFC 请求的数量会相对较少。而其他的 SFC 请求则为临时请求，该类型 SFC 请求的资源需求将远小于永久请求的资源需求，但是网络中的临时请求的数量会远多于永久请求的数量。此外，SFC 请求 r 的带宽资源需求与其计算资源需求有强相关性，即 $b_r = \beta c_r$ 。由于本文假设每处理一个单位的流量需要一个单位的计算能力^[21]，因此本文把 β 设置为 1。最后，所有 SFC 请求的源节点和目的节点均在物理网络中的 S, D 节点中随机生成。

表 4-4 SFC 迁移仿真实验中各类型 SFC 请求的参数

参数	类型 1	类型 2	类型 3
VNFs 的个数	{2,3,4,5,6}	{1,2,3}	{2,3,4,5}
每个 VNF 的资源需求	U (900,1000)	U (10,20)	U (100,200)
带宽资源需求	$\beta \cdot c_r$	$\beta \cdot c_r$	$\beta \cdot c_r$
时延需求	$\leq 20\text{ms}$	$\leq 20\text{ms}$	$\leq 20\text{ms}$
平均生命周期	永久	≥ 100	≥ 100

本文假设在初始的状态下，服务队列中一共有 6 个永久 SFC 请求。在满足资源映射约束的前提下，这些 SFC 请求的初始资源分配方案是随机生成的，因此初始状态下，物理资源的分布情况是随机的。随后，MANO 会随机收到类型为 2, 3 的 SFC 请求，且这些请求的产生将服从泊松分布^[64]，其中 $\lambda = 5$ 。

4.4.2 后续 SFC 请求的资源分配算法

本文利用了 BestFit^[80,81]及 CN^[20]算法来为后续的 SFC 请求进行资源分配，下面本小节将对这两种算法进行简单的介绍。

1. BestFit。首先 BestFit 会把 SFC 请求的 VNFs 放置在剩余物理资源最多的物理节点上，然后它在遵守带宽和时延约束的前提下，计算节点之间的最短路径，并把这些 VNFs 串联起来。
2. CN。类似地，CN 也把资源分配分成两个阶段。第一阶段，CN 会根据节点的度，中介中心性，剩余计算资源，直连链路的剩余带宽资源来为每个节点计算一个评价指标。然后 VN 会把 SFC 请求的 VNFs 放置在评价指标高的物理节点上。随后，在满足带宽资源及时延约束的前提下，CN 会计算节点之间的最短路径，并把 VNFs 进行串联。

4.4.3 评价指标

1. 物理资源分布的标准差。物理资源分布的标准差 σ_G 用于衡量物理资源分布的均衡性，且 σ_G 可以通过式子 (4-10) 来进行计算。

2. 总迁移开销。总迁移开销 \mathcal{O}_G 用于衡量对 SFC 请求进行迁移时产生的额外开销，且 \mathcal{O}_G 可以通过式子 (4-12) 来进行计算。

3. 接收率。截止至时间窗 T ，SFC 请求的接收率可以使用式子 (4-15) 进行定义，其中， NUM_NSR_S, NUM_NSR 分别表示在时间窗口 T 内成功被接收的 SFC 请求的数量及 MANO 一共收到的 SFC 请求的总数。

$$AC(T) = \frac{\sum_{t=0}^T NUM_NSR_S}{\sum_{t=0}^T NUM_NSR} \quad (4-15)$$

4. 节点资源利用率。截止至时间窗 T ，具备网络功能虚拟化能力的物理节点的计算资源利用率可以表示为式子 (4-16)。

$$NU(T) = \frac{\sum_{r \in SQ_T} \sum_{\pi \in \mathcal{F}_r} c_r}{\sum_{i \in N_V} C_i} \quad (4-16)$$

5. 链路资源利用率。截止至时间窗 T ，物理链路的带宽资源利用率可以表示为式子 (4-17)。

$$LU(T) = \frac{\sum_{r \in SQ_T} \sum_{(i,j) \in P_r} b_r}{\sum_{(i,j) \in E} B_{i,j}} \quad (4-17)$$

6. 长期收益。截止至时间窗 T ，网络运营商共获取到的收益可以利用式子 (4-18) 来进行表示，其中， W_{cpu}, W_{bw} 表示每租赁 1 单位的计算资源及 1 单位的带宽资源所获取到的收益， P_r^V 表示 SFC 请求 r 的虚拟路径。在本文中， W_{cpu}, W_{bw} 分别设置为 0.1, 0.05。

$$Pro(T) = \int_{t=0}^T \sum_{r \in SQ_T} (W_{cpu} \sum_{\pi \in \mathcal{F}_r} c_r + W_{bw} \sum_{(r, \pi_m) \in P_r^V} b_r) \quad (4-18)$$

4.4.4 仿真结果及分析

本文首先对比在两种迁移策略下进行 SFC 迁移前后，物理网络的资源分布的变化。如图 4-5 所示，在单个实验实例中，本文所提的激进的迁移策略可以令物理资源的分布变得更加均衡。此外，本文还随机产生了 20 个实验实例来阐述本文所提迁移策略的有效性，如图 4-6 所示，与保守的迁移策略对比，激进的迁移策略的 σ_G 平均要比保守迁移策略的 σ_G 低 14.991%。

然而，由于本文所提的迁移策略会对整条 SFC 链进行迁移，因此会造成更多的迁移开销，如图 4-7 所示。在 20 个实验实例中，激进的迁移策略的平均迁移开销比保守迁移策略的平均迁移开销高 25.5%，如图 4-8 所示。上述结果的总结见表 4-5。

表 4-5 20 个实验实例的平均 std 及迁移开销

物理资源状态	物理资源状态的平均 std	平均迁移开销
初始	2863.012	/
经过保守迁移后	2379.958	44176.363
经过激进迁移后	1950.773	55441.179

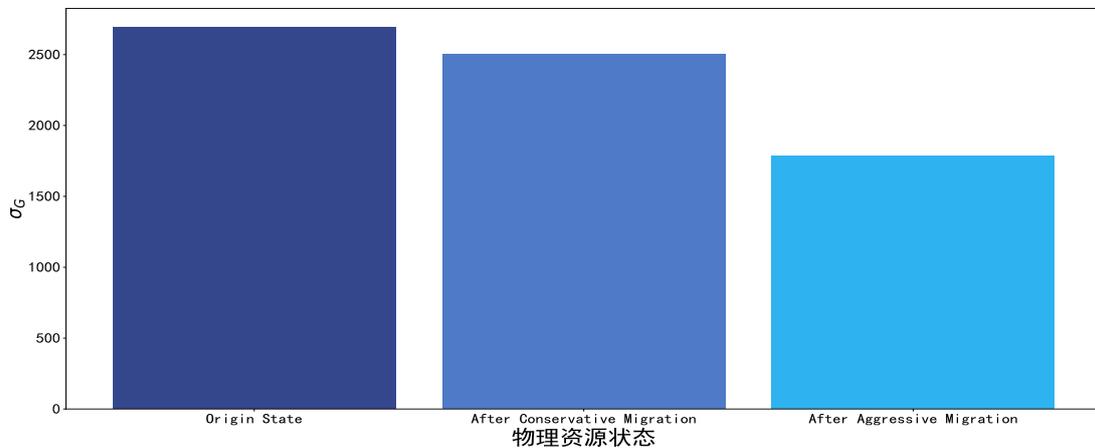


图 4-5 单个实验实例的物理资源状态

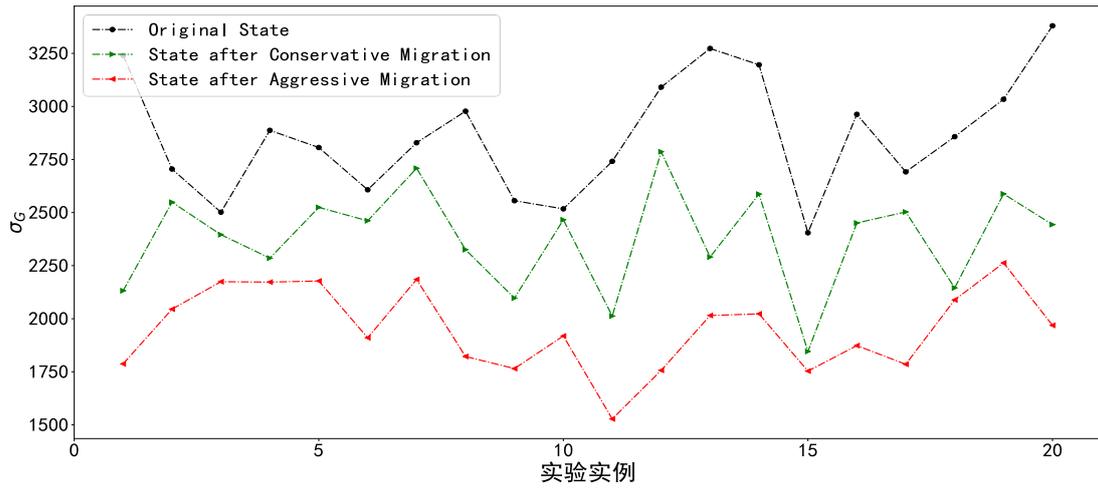


图 4-6 20 个实验实例的物理资源状态

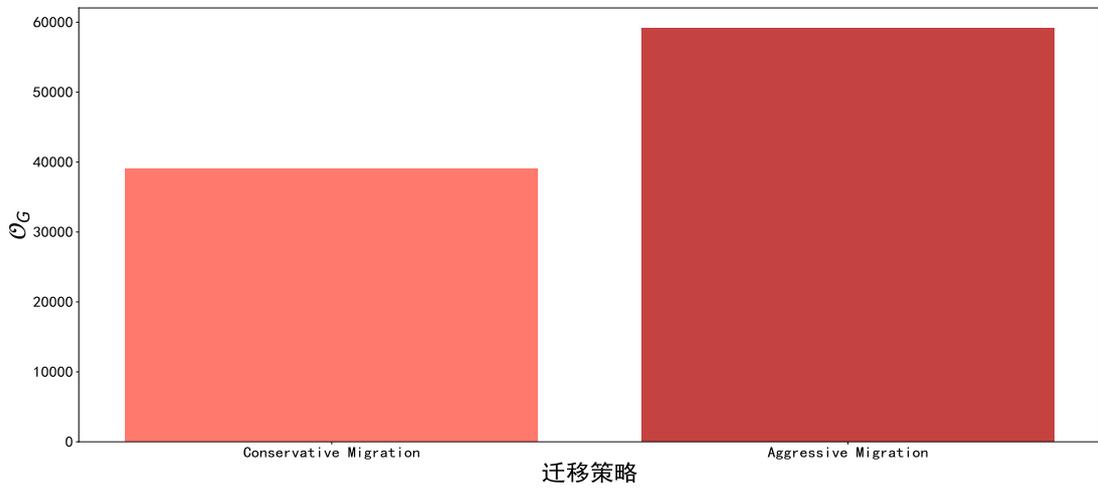


图 4-7 单个实验实例的迁移开销

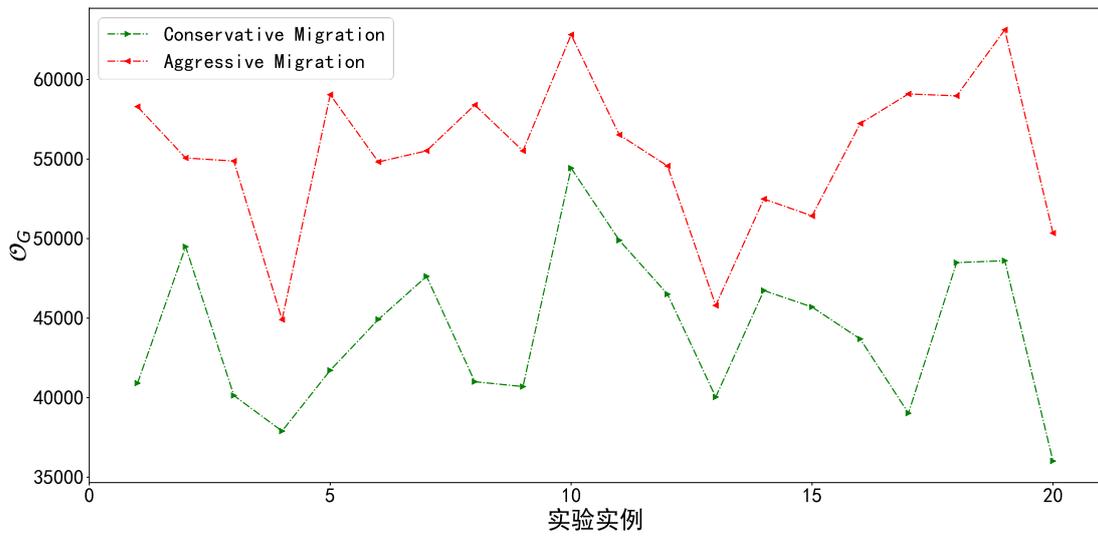


图 4-8 20 个实验实例的迁移开销

接下来, 本文对比在不同的迁移策略下, 后续 SFC 请求的接收率, 物理资源的利用率, 运营商的长期收益等评价指标的变化。

同样地, 本文先从单个实验实例的结果开始分析。如图 4-9, 图 4-10 所示, 随着时间的推移, MANO 会收到越来越多的 SFC 请求, 剩余物理资源会变得越来越少。因此, 在一段时间后, SFC 请求的接收率会开始下降。通过在初始时刻对服务队列中的 SFC 请求进行迁移, MANO 可以为更多的 SFC 请求提供服务, 从而提高最终的接收率。然而, 保守的迁移策略对最终的接收率有负面的影响。有时候, MANO 可能会接收一个类型为 3 的 SFC 请求, 以此来获取更高的收益, 在这种情况下, MANO 可能会拒绝多个类型为 2 的 SFC 请求。因此, 在少数情况下, 经过 SFC 迁移后接收率出现下降是正常现象。接下来, 本文分析物理资源的利用率和长期收益的变化。如图 4-11, 图 4-12 和图 4-13 所示, 对于“BestFit”和“CN”而言, 本文所提的激进的迁移策略可以提高最终的资源利用率及长期收益。尽管保守的迁移策略降低了“BestFit”的接收率, 但是它也提高了最终的物理资源利用率。有趣的是, 保守的迁移策略会使得物理资源利用率到达峰值的时间提前。从时间窗 30 到时间窗 44, “Conservative Migration + CN”策略的资源利用率会高于单纯“CN”策略的资源利用率。尽管在时间窗 45 后, 纯“CN”策略的资源利用率开始超过“Conservative Migration + CN”策略的资源利用率, “CN”策略仍需要花费一段时间来超过“Conservative Migration + CN”策略的长期收益。虽然从时间窗 45 到时间窗 60, “CN”策略与“Conservative Migration + CN”策略的长期收益的差距开始逐渐减少, 但是直到时间窗 60, “Conservative Migration + CN”策略的长期收益仍然高于“CN”策略的长期收益。

根据本文对长期收益的定义, 长期收益是一个与时间相关的单调增函数(在服务队列中存在任意 SFC 请求时成立)。因此, 如果 MANO 不再收到新的 SFC 请求及服务队列中的 SFC 请求保持稳定状态时, 长期收益会呈现出一条线性增长的直线。如图 4-13 所示, 从时间窗 40 到 46 时, “Conservative Migration + CN”策略的长期收益实际上要高于“Conservative Migration + BestFit”策略的长期收益。但是在时间窗 37 之后, 如图 4-11, 图 4-12 所示, “Conservative Migration + BestFit”策略的资源利用率开始高于“Conservative Migration + CN”策略的资源利用率, 这导致在时间窗 49 之后, “Conservative Migration + BestFit”策略的长期收益开始反超“Conservative Migration + CN”策略的长期收益。

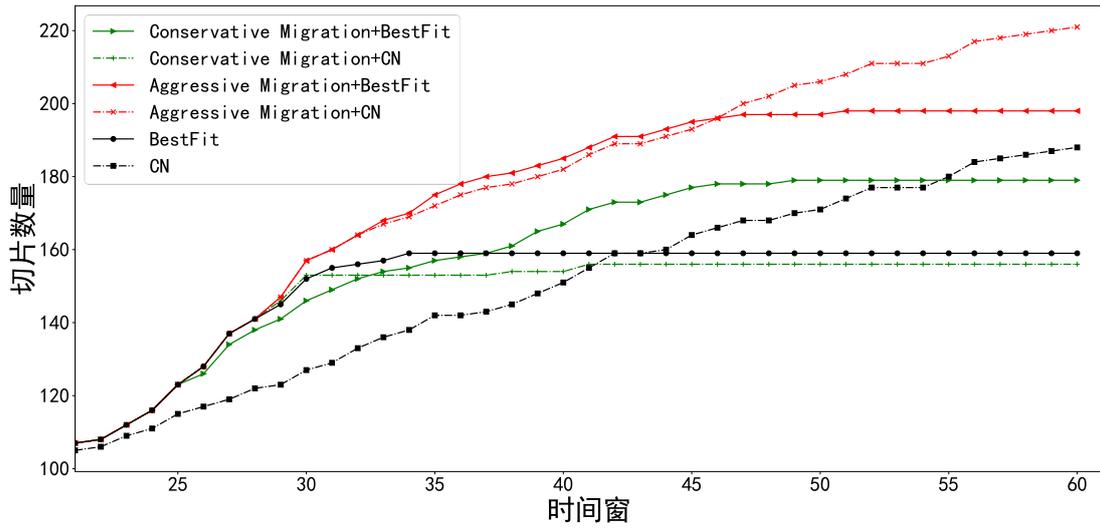


图 4-9 单个实验实例中服务队列中 SFC 请求个数的变化

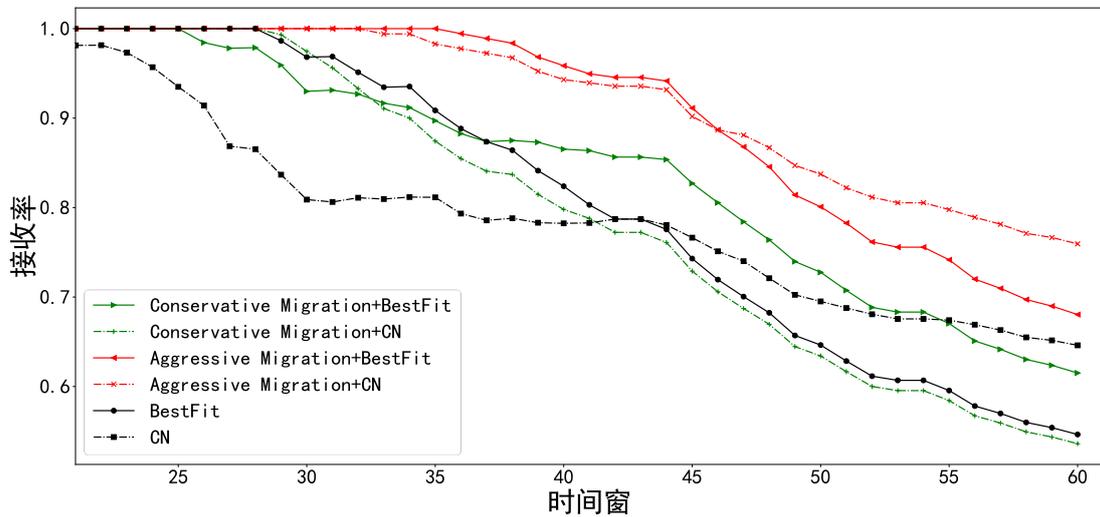


图 4-10 单个实验实例中 SFC 请求的接收率的变化

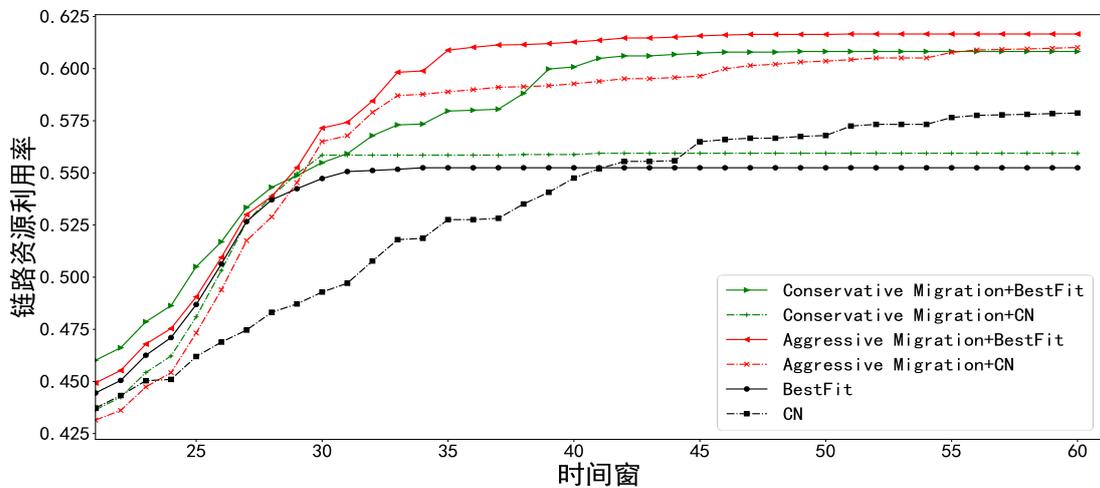


图 4-11 单个实验实例的链路资源利用率的变化

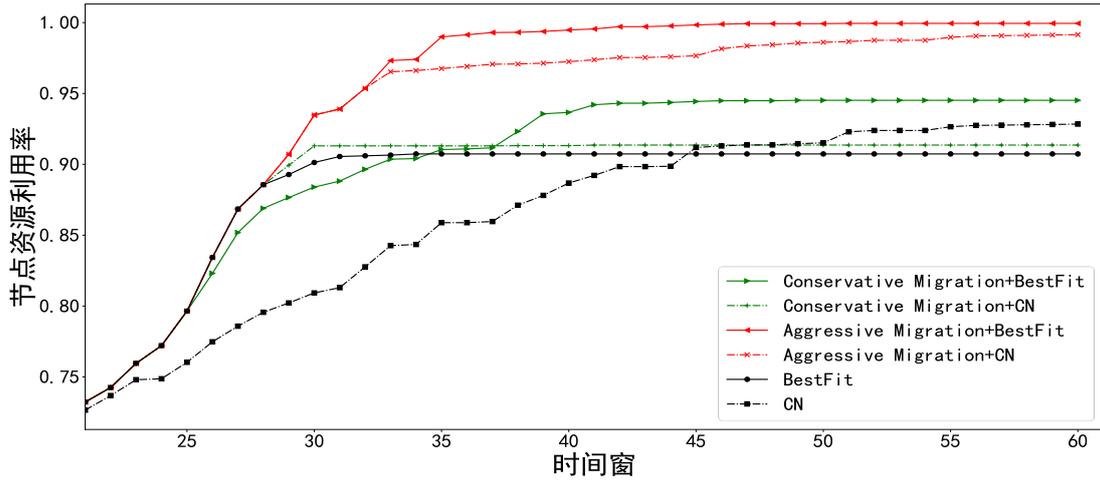


图 4-12 单个实验实例的节点资源利用率的变化

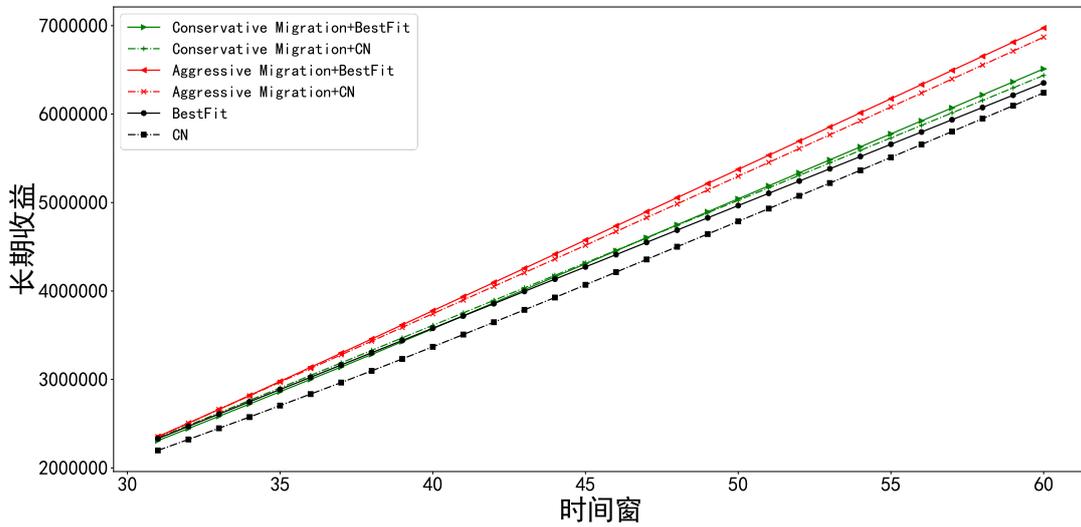


图 4-13 单个实验实例的网络运营商的长期收益的变化

接下来，本文分析另外 20 个实验实例在第 60 个时间窗的仿真结果。如图 4-14 所示，本文所提的激进的迁移策略在大部分情况下（除了实例 4，5 和 7）都能提高后续 SFC 请求的接收率。如前所述，在少数情况下，经过 SFC 迁移后接收率出现下降是正常现象。对“BestFit”而言，在经过激进的 SFC 迁移后，平均接收率提高了 11.75%，对“CN”，这个数字则变化为 6.84%。如图 4-16，图 4-17 所示，对于所有实验实例，经过激进的 SFC 迁移后，物理资源利用率及运营商的长期收益均得到的提升。对“BestFit”，节点资源利用率，长期收益分别增加了 8.14%，9.49%。对“CN”，这些数字则分别变化为 4.65%，7.14%。在部分情况下（如实例 3），激进的迁移策略会减少带宽资源的利用率，但是这并未对运营商的长期收益造成负面影响，这是因为虚拟路径的收益与其映射的物理链路的条数没有相关性。

同样地,对于所有的实验实例,保守的迁移策略均能提高运营商的长期收益,但提升的幅度要低于激进的迁移策略。对“BestFit”,长期收益的平均增益为3.05%,对“CN”,这个数字变化为4.62%。与激进的迁移策略类似,在经过 SFC 迁移后,SFC 请求的接收率,物理资源的利用率可能会下降,但是并未对长期收益造成负面影响。总而言之,在经过保守的 SFC 迁移后,“BestFit”的平均接收率,平均节点利用率,平均链路带宽资源利用率,平均长期收益分别提高了1.82%,1.59%,1.11%,3.05%。对“CN”,这些数字分别变化为2.54%,1.93%,1.74%,4.62%。

与保守的迁移策略相比,激进的迁移策略的性能会更加优异。对“BestFit”,激进的迁移策略对 SFC 请求的接收率,节点资源利用率,带宽资源利用率,长期收益的平均增益要比保守的迁移策略分别高9.93%,6.55%,3.2%,6.44%。对“CN”,这些数字分别变化为4.3%,2.72%,1.09%和2.52%。除此之外,在大部分情况下(除了实例5,7),经过激进的 SFC 迁移后,物理节点的资源利用率可以接近100%。尽管 SFC 迁移会带来一定的开销,但是相对于最终长期收益的增幅,这些开销是微不足道的。对“CN”而言,由保守迁移策略和激进迁移策略产生的迁移开销仅仅为最终原始长期收益的0.7074%和0.8844%。对于“BestFit”,这些数字则为0.7008%和0.8795%,上述分析结果的总结见表4-6。

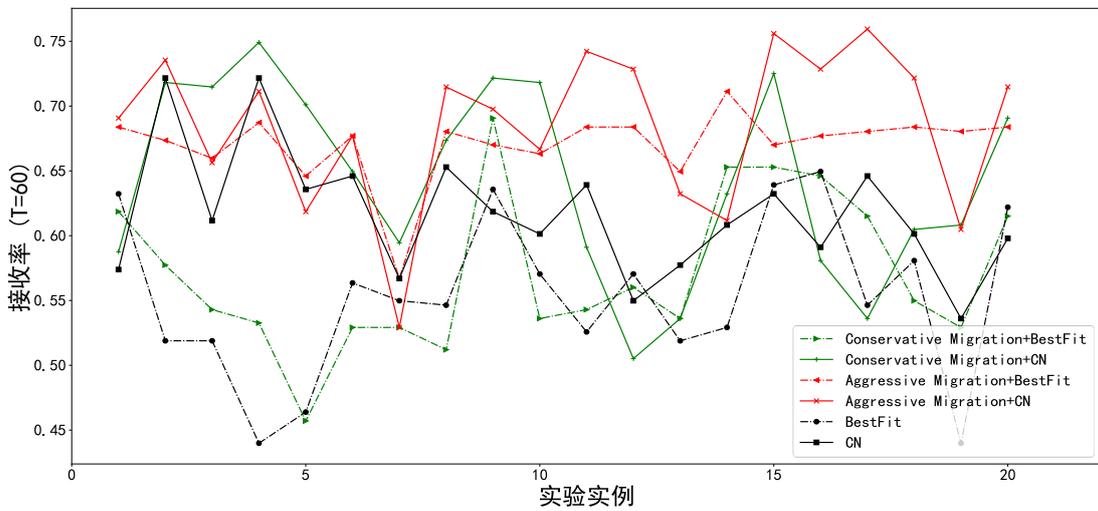


图 4-14 20 个实验实例在 $T = 60$ 时的 SFC 请求接收率

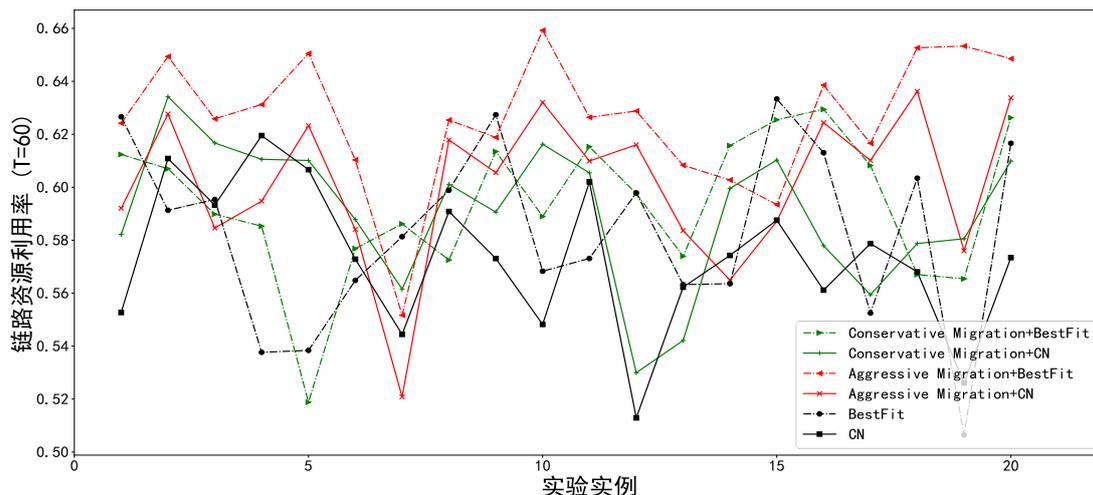


图 4-15 20 个实验实例在 $T = 60$ 时的链路资源利用率

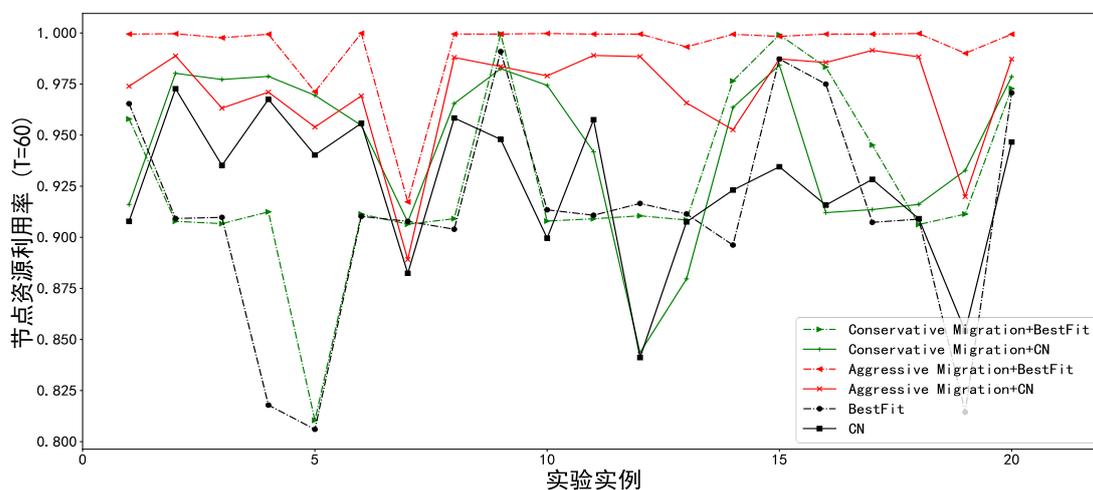


图 4-16 20 个实验实例在 $T = 60$ 时的节点资源利用率

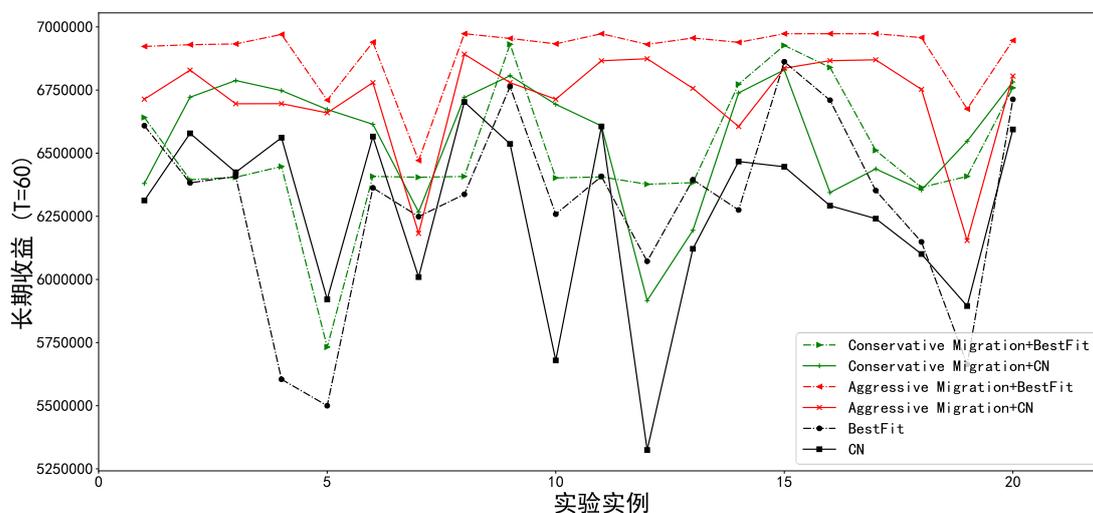


图 4-17 20 个实验实例在 $T = 60$ 时的长期收益

表 4-6 20 个实验实例的平均结果及增幅

物理网络状态及资源 分配方法	平均接收率 (增幅)	平均节点资源 利用率 (增幅)	平均链路资源 利用率 (增幅)	平均长期收益 (增幅)
初始 & BestFit (基线)	0.5531	0.7117	0.5827	6303710.1509
初始 & CN (基线)	0.6165	0.9243	0.5729	6268964.1997
经过保守迁移 & BestFit	0.5713 (1.82%)	0.9276 (1.59%)	0.5938 (1.11%)	6496126.1533 (3.05%)
经过保守迁移 & CN	0.6419 (2.54%)	0.9436 (1.93%)	0.5903 (1.74%)	6558302.0120 (4.62%)
经过激进迁移 & BestFit	0.6706 (11.75%)	0.9931 (8.14%)	0.6258 (4.31%)	6901688.4058 (9.49%)
经过激进迁移 & CN	0.6849 (6.84%)	0.9708 (4.65%)	0.6012 (2.83%)	6716408.8979 (7.14%)

4.5 本章小结

本章研究了物理网络资源分布的不均衡性对后续 SFC 请求及运营商所造成的影响。本章首先利用 ILP 模型对 SFC 迁移问题进行建模，然后，受到一些优秀相关工作的启发，本文设计了一个保守的 SFC 迁移策略以及提出了一个激进的 SFC 迁移策略。经过 SFC 迁移后，本文利用两个先进的 SFC 资源分配策略来为后续收到的 SFC 请求进行资源映射。仿真实验表明，物理资源分布的不均衡性会对 SFC 请求的接收率、运营商的长期收益，等造成负面影响。此外，本文所设计的保守迁移策略及所提出的激进迁移策略均能有效地消除这些负面影响。

5 基于资源预测机制的关键 SFC 迁移优化策略

5.1 引言

在第 3、4 章中，本文分别完成了网络中关键业务流的识别及面向关键网络服务功能链的迁移的相关研究工作，且本文假设物理资源需求大的 SFC 请求（关键业务流）为永久请求，相应地，这部分 SFC 请求的数量远小于其他 SFC 请求的总数。

然而，在现实场景中，网络流量的变化是有规律的。以带宽资源的利用率为例，白天的利用率要远高于晚上的利用率^[34]，这意味着在白天，SFC 请求的资源需求是高于其在晚上的资源需求的，这一现象为熟知的昼夜模式^[36]。因此，为了更加有效地利用有限的物理资源，本文考虑在进行 SFC 迁移前，先根据 SFC 请求的历史流量数据来预测其在后续一段时间的资源需求（目前本文的工作只关注资源需求收缩的场景，对于资源需求扩张的场景，本文并未进行考虑），然后再进行资源重映射。

本章结合预测机制来展开 SFC 迁移的相关研究。具体地，本章先把资源预测问题定义成时间序列的多步预测问题，接着利用皮尔森系数来剔除异常点，并利用常用的时间序列预测方法来对带宽资源需求进行预测。最后，本章把预测的结果应用于 SFC 迁移的相关策略中，并通过仿真实验来证明资源预测机制的必要性及有效性。

5.2 问题定义

如前所述，流量的昼夜模式是常见现象。本文需要根据 SFC 请求 i 的历史流量数据来预测出其在后续一段时间内的流量总量，以此来评估其后续的带宽资源需求。本文利用 $F_i = \{f_i^0, f_i^1, \dots, f_i^T\}$ 来表示 SFC 请求 i 流量的历史时间序列数据，其中 f_i^T 表示 SFC 请求 i 从时间窗 T 到 $T+1$ 所产生的流量的总合，其中， T 的单位为 1 小时。根据时间序列数据的定义，本章的问题可以被确定为，利用时间序列数据 F_i 来预测后 n 个时间序列数据 $[f_i^{T+1}, \dots, f_i^{T+n}]$ 。如图 5-1 所示，本文需要根据 SFC 请求 i （流量数据来源于 WIDE 网络^[37]）在 2021 年 8 月 2 日的流量走势来预测（ $n=3$ ）其在 2021 年 8 月 3 日 0 时至 3 时的流量数据。

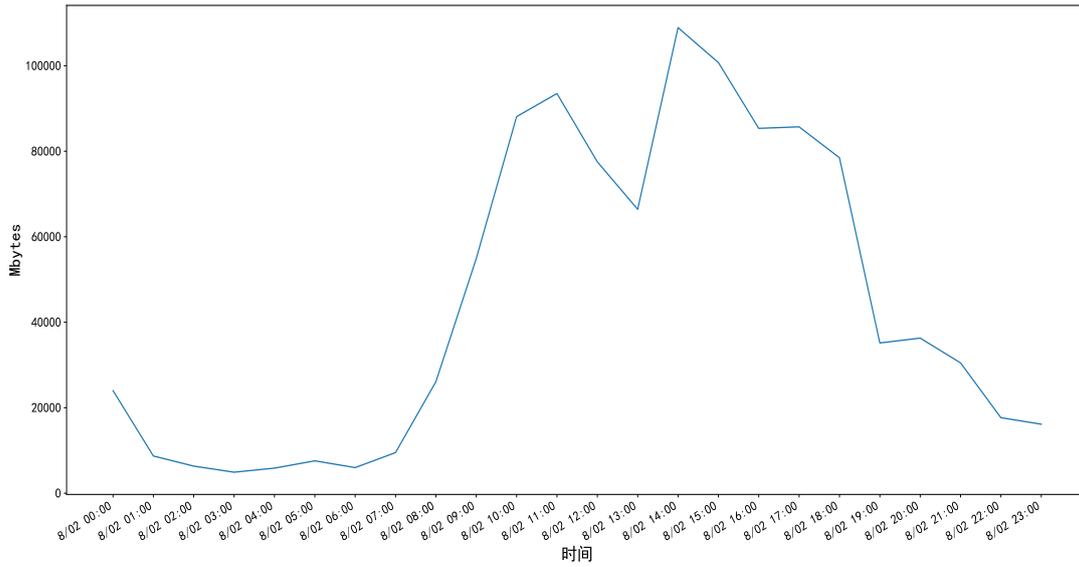


图 5-1 SFC 请求 i 的流量走势图

5.3 问题求解

本节将介绍数据处理的流程以及本章所使用的三种时间序列预测方法。

5.3.1 数据处理

本文定义 $D_{date} = \{f_i^j, f_i^{j+1}, \dots, f_i^{j+23}\} (j = 0, 24, 48, \dots)$ 为 SFC 请求 i 在某天的具体流量数据, 然后使用皮尔森系数 ρ 来定义该请求在不同的两天内的流量走势的相似性。 ρ 的计算方式如式子 (5-1) 所示, 其中 $Cov(X, Y)$ 表示 X, Y 之间的协方差, σ_X, σ_Y 分别为 X, Y 的标准差。

$$\rho = \frac{Cov(X, Y)}{\sigma_X \sigma_Y} \quad (5-1)$$

一般地, 当 $|\rho| > 0.6$ 时, 可以认为 X, Y 之间具有强关联性。本文选取 SFC 请求 i 中某天具备明显昼夜模式的流量数据作为基准 X , 并与其他天的流量数据 Y 进行走势相似度验证, 以有效地移除异常数据点。

由于本文并不考虑需求扩张的情况, 因此本文仅预测每个 SFC 请求从凌晨 2 点到凌晨 5 点的流量数据, 即在本文中, 预测步数 $n = 3$, 并把预测结果作为 SFC 迁移时进行资源重映射的参考依据。此外, 在大部分情况下, 业务在周末产生的流量与其在工作日产生的周末具有较大差异, 因此本文还移除了 SFC 请求在周末的时间序列数据。经过上述处理, SFC 请求 i 的时间序列数据为 $F_i' = \{f_i^2, f_i^3, f_i^4, \dots, f_i^{T+24}, f_i^{T+1+24}, f_i^{T+2+24}, \dots\}$ 。

5.3.2 预测方法

本章分别利用了三个时间序列预测常用的模型来预测 SFC 请求 i 的后续流量数据，接下来，本小节对这三个模型进行简单的介绍。

1. 差分整合移动平均自回归 (Autoregressive Integrated Moving Average, ARIMA)。ARIMA 是时间序列数据分析中的一个经典方法^[48]，一般地，ARIMA 通过差分处理来把非平稳的时间序列转化成平稳的时间序列，从而有效地对后续的数据进行预测。

2. 长短时记忆模型 (Long Short-Term Memory, LSTM)^[50]。LSTM 是递归神经网络的一种特殊形式，且能有效地对时间序列数据进行分析。一般地，一个 LSTM 单元包括一个输入门，一个输出门以及一个遗忘门。

3. Prophet^[53]。Prophet 是一种有效的的时间序列预测方法。一般地，Prophet 将预测问题转化为曲线的拟合问题，它综合考虑了时间序列走向的趋势，季节性和假日因素。

需要注意的是，上述方法的详细描述可以见本文第 2 章的第 3 小节。

5.4 仿真与性能分析

本节将展开仿真实验以验证资源预测机制的优越性。首先，本节将介绍本章所使用的数据集，然后介绍一系列的评价指标。最后，本节将在第 4 章的基础上加入资源预测机制并对比迁移算法性能的变化。

本章所利用的 ARIMA, LSTM 和 Prophet 算法的运行环境如下：操作平台: Windows 11; 硬件平台: Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz; 编程语言: Python 3.6。此外，本章所采用的数据集和评价指标的具体内容如章节 5.4.1, 5.4.2 所示。

5.4.1 数据集

本章使用的数据集是由 WIDE^[37]项目组提供的。从 2021 年 8 月 1 日开始，本文对一个由源 IP 和目的 IP 前缀定义的流在 WIDE 网络中的流量数据进行了为期两个月的监控，并且认为所收集到的数据为一个 SFC 请求的原始流量数据。

5.4.2 评价指标

为了客观地评价各个预测方法的性能，本文定义了两个评价指标 P, Q 。 P, Q 的计算方法如式子 (5-2), (5-3) 所示，其中 Predict_{\max} 表示在一个多步预测任务中，输出结果的最大值， Actual_{\max} 表示一个 SFC 请求在凌晨 2 点到凌晨 5 点的流量数据的最大值， b_r 表示 MANO 分配给 SFC 请求的原始带宽资源，一般 b_r 可以视为该 SFC 请求在前一天流量数据的峰值。

$$Q = \frac{b_r}{\text{Predict}_{\max}} \quad (5-2)$$

$$P = \frac{\text{Predict}_{\max}}{\text{Actual}_{\max}} \quad (5-3)$$

根据式子 (5-2), (5-3) 可以得出， P 用于衡量预测值可以在多大程度上满足一个 SFC 请求在后续一段时间内的带宽资源需求，而 Q 则用于衡量预测方法可以在多大程度上为 MANO 节约已分配出去的带宽资源。为了保障各个用户的服务质量，显然，在大部分情况下，预测结果在 $P \geq 1$ 时才有意义。

5.4.3 仿真结果及分析

接下来，本文对比所使用的三种预测方法在给定评价指标下的性能。本文先进行原始的数据集进行异常点移除的工作，如图 5-2 所示，在未移除周末的流量数据前，原始数据集中有部分流量的走势并不符合图 5-1 的趋势。当移除原始数据中周末的流量数据并再使用 2021 年 8 月 2 日的流量数据作为参照（由于 2021 年 8 月 1 日为周日，因此使用下一天的数据作为参照）时，可以发现原始数据集中绝大部分天数的流量走势均符合预期的假设，如图 5-3 所示。然后，本文还在图 5-3 的基础上再次移除 $|\rho| < 0.6$ 的流量数据，以确保数据集中不存在异常的流量数据。

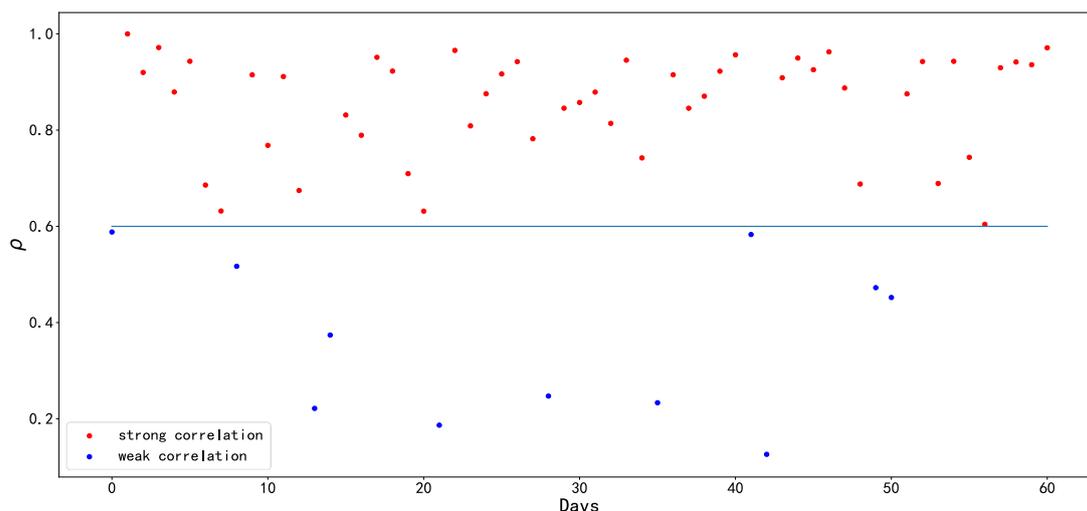


图 5-2 单个 SFC 请求在 2021 年 8 月 2 日与其他天数之间的皮尔森系数

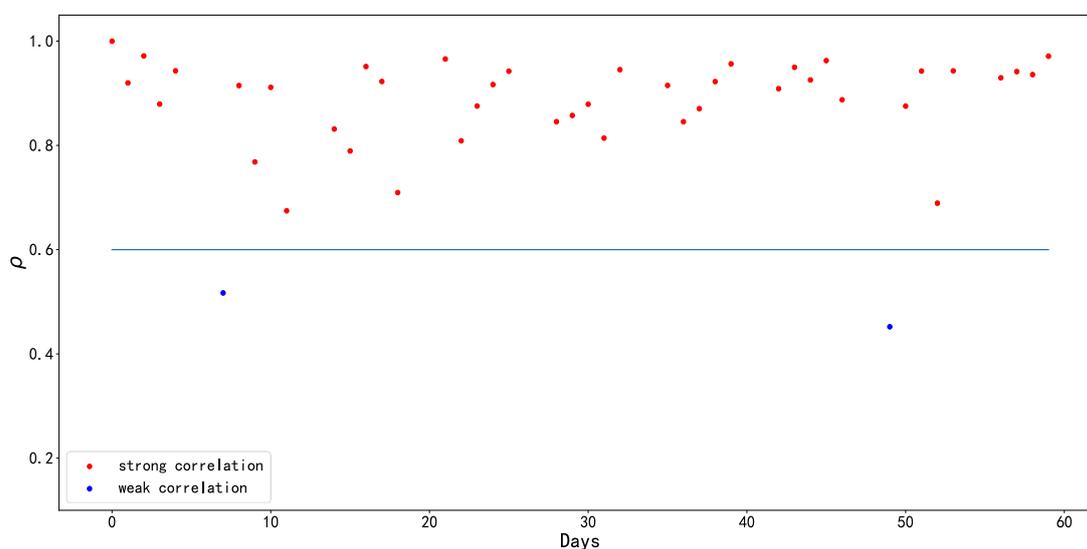


图 5-3 单个 SFC 请求在 2021 年 8 月 2 日与其他天数之间的皮尔森系数 (移除周末的流量数据)

接下来，本文需要确定 ARIMA 模型以及 LSTM 模型的具体参数。一般地，ARIMA 需要确定 p, d, q 三个参数。为了确定参数 d ，本文先对数据集 F' 进行一阶差分处理，然后使用增强 Dickey-Fuller (Augmented Dickey-Fuller, ADF) 检测来验证数据集的平稳性。由于计算结果为 $4.5729616 \times 10^{-13}$ ，远小于基线值 0.05，因此可以认为处理过的数据具备平稳性，因此令 $d = 1$ 。

对于其他两个参数，与相关工作^[87]类似，本文利用自相关函数 (Autocorrelation Function, ACF) 和偏自相关函数 (Partial Autocorrelation Function, PACF) 来对数据进行分析，如图 5-4，图 5-5 所示，本文令 $p = 1, q = 2$ 。

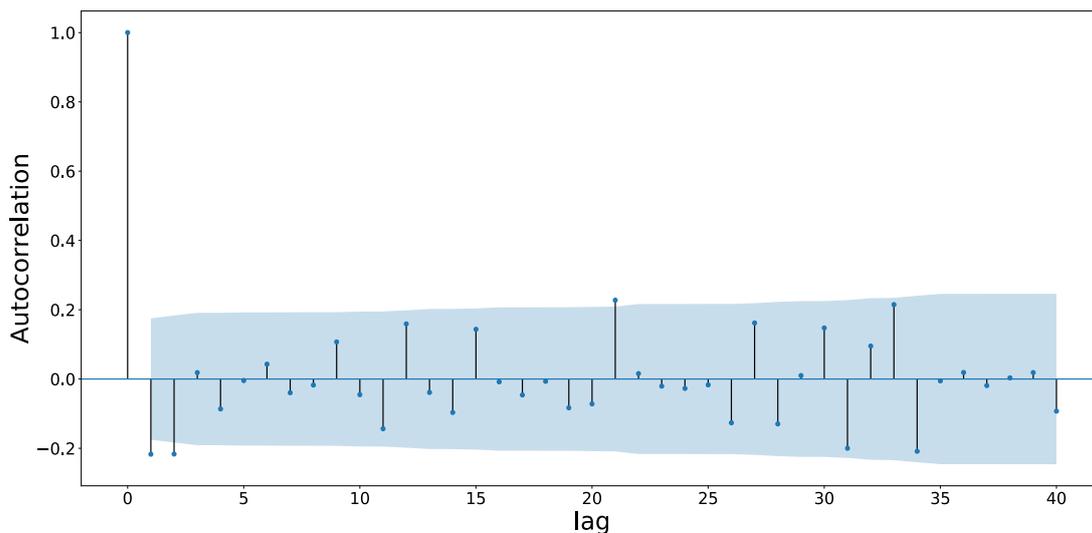


图 5-4 数据的 ACF

对于 LSTM 模型，本文的参数设置参考相关工作^[88]。由于 prophet 模型不需要确定具体参数，因此本文可以直接进行调用。

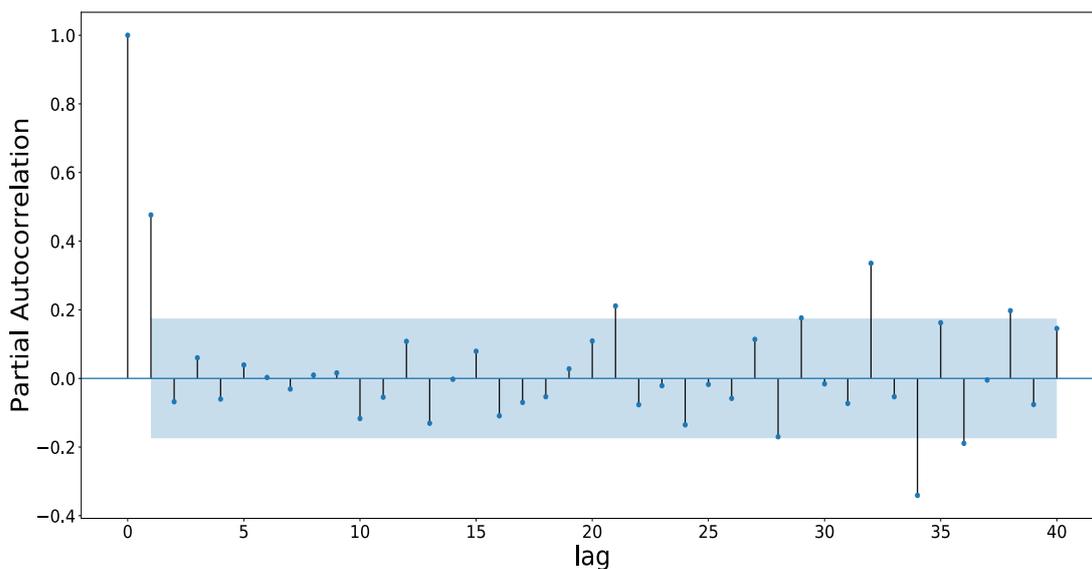


图 5-5 数据的 PACF

然后，本文保留了数据集中最后 5 天的数据作为验证集，并把其余的数据作为训练集，值得注意的是，由于 Prophet 模型可以分析数据集中的季节性和节假日因素，因此可以把原始的数据直接输入到 Prophet 模型中，最后的实验结果如图 5-6 所示。

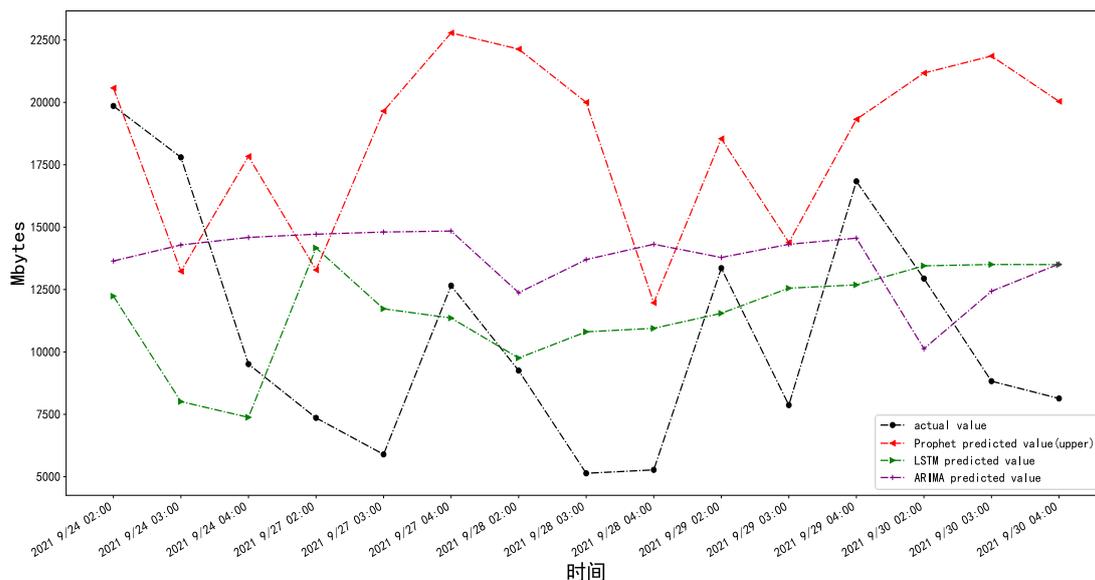


图 5-6 不同预测方法的预测结果

最后，本文利用 P, Q 指标来量化各个预测方法的性能，如表 5-1 所示。其中 Prophet 模型可以在最大限度上保证 SFC 请求的用户体验，而 LSTM 模型则更倾向于牺牲部分的用户体验来节约更多的带宽资源。相比之下，ARIMA 模型则介于两者之间。

表 5-1 不同方法在单个 SFC 请求的预测结果的对比分析

	第一天		第二天		第三天		第四天		第五天	
	P	Q								
Prophet(upper)	1.05	5.09	1.77	3.83	2.30	4.75	1.24	4.49	1.65	4.36
ARIMA	0.74	7.21	1.17	5.84	1.55	7.43	0.87	6.39	1.05	6.91
LSTM	0.62	8.57	1.12	6.12	1.18	9.72	0.75	7.33	1.04	6.92

由上分析得，Prophet 模型为最佳选择（不仅能保障业务的 QoE，且无需进行数据的预处理）。此外，本文在 WIDE 中收集了另外 6 个数据流的原始流量数据（进行为期两个月的监控），并将其视为第 2 章实验中 6 个永久 SFC 请求的流量数据。在保留最后一天的数据作为验证集的前提下，本文验证了 Prophet 模型的有效性，其预测结果如表 5-2 所示。

表 5-2 Prophet 在 6 个 SFC 请求中的预测结果分析

	SFC 1	SFC 2	SFC 3	SFC 4	SFC 5	SFC 6
P	1.21	3.44	2.79	1.44	1.85	3.36
Q	2.61	1.78	1.03	1.10	2.71	1.91

如前所述,在进行资源重映射前,本文将结合预测的结果来确定各个 SFC 请求在后续一段时间内的资源需求,包括计算资源以及带宽资源。具体地,对于 SFC 请求 r ,本文令 $b_r' = \frac{b_r}{Q} \cdot (Q-1), c_r' = \frac{c_r}{Q} \cdot (Q-1)$,其中, b_r', c_r' 分别表述资源重映射时 SFC 请求 r 在后续一段时间内的带宽资源需求及计算资源需求。在此条件下,最终后续 SFC 请求的接收率以及运营商的长期收益均可以得到进一步的提升(由于结合了预测机制后,物理资源的利用率将难以界定,因此本文并未为对物理资源利用率进行分析),如图 5-7,图 5-8 所示。

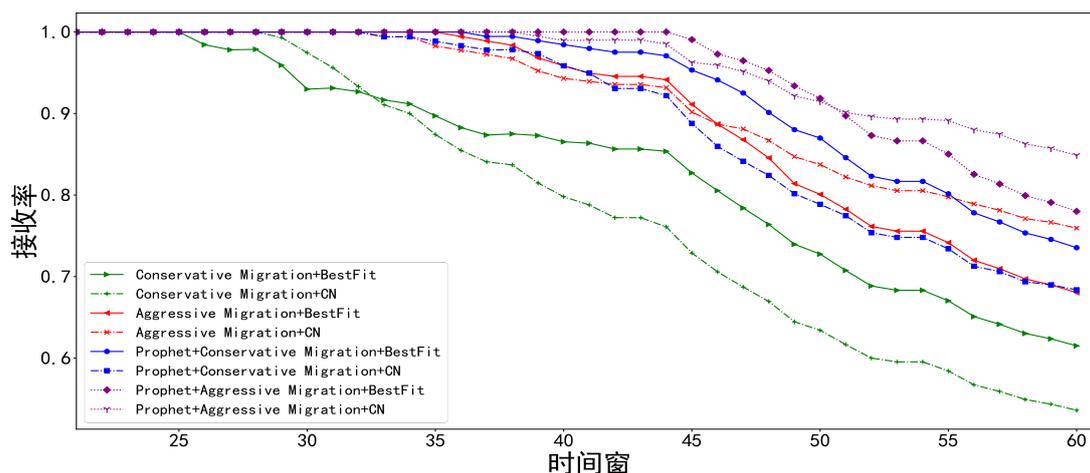


图 5-7 结合资源预测机制后单个实验实例中 SFC 请求的接收率的变化

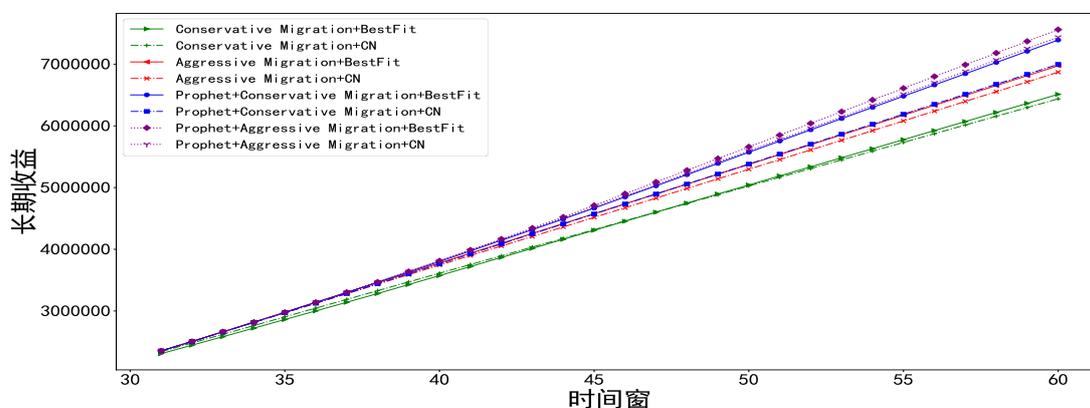


图 5-8 结合资源预测机制后单个实验实例的网络运营商的长期收益的变化

此外，本文还验证了在其余 20 个实验实例中，资源预测机制的有效性。如图 5-9，图 5-10 所示，尽管在一些实验实例中，结合了资源预测机制后，SFC 请求的接收率出现了下降（如实验实例 17，19），但由于接收了更多资源需求大的 SFC 请求，长期收益仍然呈上升趋势。在结合资源预测机制前后，无论是“BestFit”策略还是“CN”策略，平均的 SFC 请求接收率和运营商的长期收益均得到进一步的提升。其中，对于“Conservative Migration + BestFit”策略，在结合 Prophet 模型后，平均 SFC 请求接收率和平均长期收益分别进一步提高了 16.66%，14.33%。对于“Conservative Migration + CN”策略，这些数字变化为 11.02%，11.37%。对于激进的迁移策略而言，在结合 Prophet 后，“BestFit”的 SFC 请求接收率，长期收益分别进一步提高了 9.28%，8.52%，对“CN”而言，这些数字则变化为 8.54%，9.57%，上述结果见表 5-3 所示。

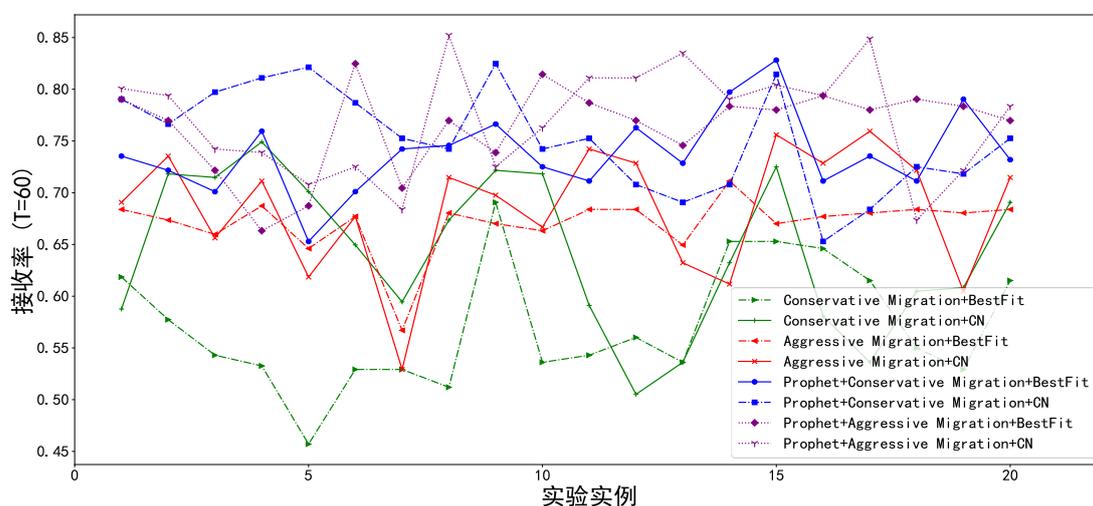


图 5-9 结合资源预测机制后 20 个实验实例在 $T = 60$ 时的 SFC 请求的接收率

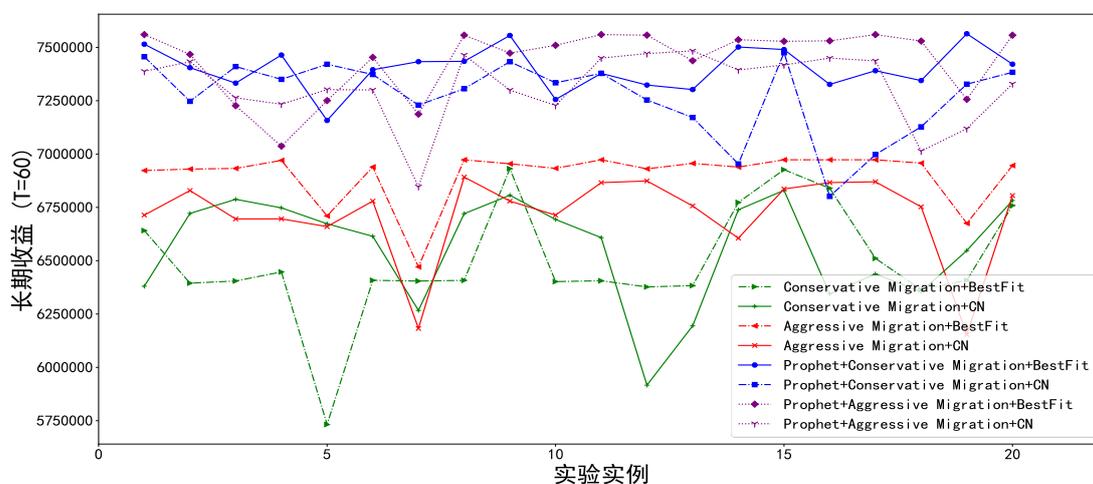


图 5-10 结合资源预测机制后 20 个实验实例在 $T = 60$ 时的长期收益

表 5-3 结合预测机制后 20 个实验实例的平均结果及增幅

物理网络状态及资源分配方法	平均接收率（增幅）	平均长期收益（增幅）
初始 & BestFit（基线）	0.5531	6303710.1509
初始 & CN（基线）	0.6165	6268964.1997
经过保守迁移& BestFit	0.5713 (1.82%)	6496126.1533 (3.05%)
经过保守迁移 & CN	0.6419 (2.54%)	6558302.0120 (4.62%)
经过激进迁移 & BestFit	0.6706 (11.75%)	6901688.4058 (9.49%)
经过激进迁移 & CN	0.6849 (6.84%)	6716408.8979 (7.14%)
Prophet+经过保守迁移 & BestFit	0.7379 (18.48%)	7399583.6915 (17.38%)
Prophet+经过保守迁移 & CN	0.7521 (13.56%)	7271458.5999 (15.99%)
Prophet+经过激进迁移 & BestFit	0.7634 (21.03%)	7438781.0881 (18.01%)
Prophet+经过激进迁移 & CN	0.7703(15.38%)	7316273.5158 (16.71%)

5.5 本章小结

本章结合了主流的时间序列预测技术对第 4 章中所提的两种 SFC 迁移策略进行优化，具体为，本章先根据各个 SFC 请求的历史流量数据来预测其在后续一段时间内的资源需求，然后再进行资源重映射。仿真实验结果表明，在资源需求发生收缩的阶段，结合资源预测机制后，SFC 迁移策略的性能可以得到进一步的提升，因此给后续的 SFC 请求以及运营商带来正向反馈。

6 总结与展望

6.1 主要工作总结

为了满足多样化的网络垂直业务，5G 网络采用 SFC 技术来为各个业务场景提供定制化的业务。然而，由于 SFC 请求的生命周期、资源调整等因素的影响，物理网络中的资源分布会变得不均衡，从而给后续的业务请求以及网络运营商带来负面影响。为消除这些负面影响，本文展开了面向 5G 网络服务功能链迁移机制的研究，具体工作内容如下。

1. 本文针对现有的面向 5G 网络服务功能链的资源分配机制，迁移机制等研究工作做了详细的调研，不仅分析了现有研究工作的各类应用场景、所用的理论知识，而且对比了现有研究与本文所展开的研究工作的差异。
2. 针对网络中关键业务的动态识别问题，本文首先使用 CMAB 模型来对问题进行建模。然后，本文设计了一个基于贪婪思想的通用摇臂选取算法，并提出了多种选取策略。最后展开了基于真实网络流量数据的实验来验证本文所设计算法的有效性。
3. 为了解决 SFC 迁移问题，本文先使用整数线性规划模型来对该问题进行刻画，接着设计了两种迁移策略来完成 SFC 的迁移工作。对于后续收到的 SFC 请求，本文利用了两个先进的资源分配算法来对其进行资源映射。最后本文通过仿真实验来证明 SFC 迁移的必要性以及验证本文所设计的 SFC 迁移策略的有效性。
4. 考虑到实际业务资源需求的昼夜现象，本文结合了当前主流的时间序列预测技术来进一步优化所提的两种 SFC 迁移策略。具体为，在资源需求收缩阶段，本文先利用时间序列预测技术预测 SFC 请求在后续时间段的资源需求，然后根据预测结果展开资源重映射工作。

6.2 创新点

本文瞄准具有广泛应用前景的网络服务功能链技术，本着“理论研究—技术实现—实际应用”相结合的思想 and 策略，研究过程中针对网络关键业务流的动态识别问题，网络服务功能链的迁移问题提出了一些新的解决策略和实现技术。具体包括以下 3 个方面：

1. 对组合多摇臂赌博机模型的修订。本文综合考虑到新型网络技术 SDN 的发展,网络运营商可以实时地收集到各业务的流量数据,基于这一优势,本文弱化了传统 CMAB 模型中探索与利用的约束条件。同时,为了更加契合网络流量数据的波动性,本文还为每个摇臂的奖赏分布增加了时变特性。
2. 新颖的 SFC 迁移场景。在大部分现有的 SFC 迁移工作中,研究人员考虑了由业务移动性引发的 SFC 迁移需求,而忽略了网络中可分配的物理资源的分布情况对运营商,后续 SFC 请求的影响。本文创新性地利用标准差来表征物理网络中的资源分布状态,并对 SFC 迁移时产生的开销进行了的定义。最后本文设计了两种有效的 SFC 迁移策略并通过仿真实验来证明了 a) 物理资源的分布情况对用户和网络运营商所造成的负面影响。b) 本文所设计的 SFC 迁移策略的有效性。
3. 综合考虑业务资源需求的昼夜现象。在现有的 SFC 迁移工作中,研究人员忽略了 SFC 请求的物理资源的时变性。本文结合主流的时间序列预测方法,对 SFC 迁移策略进行进一步的优化。

6.3 不足与展望

以实际的落地应用为目标,论文提出的面向 5G 网络服务功能链的迁移策略还存在以下 3 个需要改进的方面:

1. 对细颗粒度的关键业务流识别的不足。为了压缩解空间,本文把一条业务流定义成一个源节点到目的节点的组合,然而,在现实的应用场景中,一条业务流通常是用 5 元组(源节点,目的节点,源端口,目的端口,协议)来进行定义的。且本文并未考虑业务流的生命周期,因此在后续的落地应用中,还可以考虑结合睡眠机制对 CMAB 模型进行修订,并进一步的增加摇臂的数量,以增强对细颗粒度的关键业务流的普适性。
2. 忽略了用户在资源扩展阶段的服务质量。本文展开基于资源预测机制的 SFC 迁移策略的研究时,仅仅考虑到用户在资源收缩阶段的场景,而并未对用户处于资源扩张阶段时的资源需求进行评估。在后续工作中,需要对这一应用场景进行综合考虑。

此外,本文还需要展开以下方面的研究:

1. 更复杂的应用场景。目前,本文研究了核心云中的 SFC 迁移问题。但对于一些更复杂的应用场景,如网络切片请求,则需要同时考虑边缘云服务器和核心云服务器,因为它们可以形成具有不同服务链的复杂 VNFs 转发图。在这种情况下,本文定义的模型并不适合实际情况。因此需要

重新考虑新的约束条件，以设计一个更复杂的模型。

2. 进一步探索优化目标。如图 4-17 所示，虽然在大多数情况下消除物理资源分配不平衡对网络运营商确实有积极影响，但在少数情况下（如实例 7），这种改进并不明显。此外，本文目前的工作没有考虑不同物理节点、物理链路之间资源上限的差异以及拓扑结构可能对本文的优化目标造成的潜在影响。因此，要进一步研究物理资源分布的不均衡性与网络运营商长期收益之间的具体关系，在未来还需要进行大量的实验。最后还可以考虑物理节点和链路的一些有趣的属性，比如节点的度，中介中心性，来进一步调整迁移的目标。

参考文献

- [1] Taleb T, Kunz A. Machine type communications in 3GPP networks: potential, challenges, and solutions[J]. *IEEE Communications Magazine*, 2012, 50(3): 178-184.
- [2] Kazmi S M A, Khan L U, Tran N H, et al. *Network slicing for 5G and beyond networks*[M]. Berlin: Springer, 2019.
- [3] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks[J]. *ACM SIGCOMM computer communication review*, 2008, 38(2): 69-74.
- [4] Han B, Gopalakrishnan V, Ji L, et al. Network function virtualization: Challenges and opportunities for innovations[J]. *IEEE communications magazine*, 2015, 53(2): 90-97.
- [5] Yi B, Wang X, Li K, et al. A comprehensive survey of network function virtualization[J]. *Computer Networks*, 2018, 133: 212-262.
- [6] Fei X, Liu F, Jin H, et al. FlexNFV: Flexible network service chaining with dynamic scaling[J]. *IEEE Network*, 2020, 34(4): 203-209.
- [7] Li M, Zhang Q, Liu F. Finedge: A dynamic cost-efficient edge resource management platform for NFV network[C]//2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS). IEEE, 2020:1-10.
- [8] Wang T, Xu H, Liu F. Multi-resource load balancing for virtual network functions[C]//2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017: 1322-1332.
- [9] Alliance N. 5G white paper[J]. *Next generation mobile networks, white paper*, 2015, 1(2015).
- [10] J. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," IETF RFC 7665, 2015. [Online]. Available: <https://tools.ietf.org/html/rfc7665>. [Accessed: 10-Dec-2015].
- [11] Zhang Q, Xiao Y, Liu F, et al. Joint optimization of chain placement and request scheduling for network function virtualization[C]//2017 IEEE 37th international conference on distributed computing systems (ICDCS). IEEE, 2017: 731-741.
- [12] Jang I, Suh D, Pack S, et al. Joint optimization of service function placement and flow distribution for service function chaining[J]. *IEEE Journal on Selected Areas in Communications*, 2017, 35(11): 2532-2541.
- [13] Medhat A M, Taleb T, Elmangoush A, et al. Service function chaining in next generation networks: State of the art and research challenges[J]. *IEEE Communications Magazine*, 2016, 55(2): 216-223.
- [14] Santos G L, Bezerra D F, Rocha E S, et al. Service function chain placement in distributed scenarios: a systematic review[J]. *Journal of Network and Systems Management*, 2022, 30(1): 1-39.
- [15] Xiao Y, Zhang Q, Liu F, et al. NFVdeep: Adaptive online service function chain deployment with deep reinforcement learning[C]//Proceedings of the International Symposium on Quality of Service. 2019: 1-10.
- [16] Pei J, Hong P, Xue K, et al. Two-phase virtual network function selection and chaining algorithm based on deep learning in SDN/NFV-enabled networks[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(6): 1102-1117.

- [17] Cao J, Zhang Y, An W, et al. VNF-FG design and VNF placement for 5G mobile networks[J]. *Science China Information Sciences*, 2017, 60(4): 1-15.
- [18] Agarwal S, Malandrino F, Chiasserini C F, et al. VNF placement and resource allocation for the support of vertical services in 5G networks[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(1): 433-446.
- [19] Alliance, N. G. M. N. "Description of network slicing concept." NGMN 5G P 1.1 (2016).
- [20] Guan W, Wen X, Wang L, et al. A service-oriented deployment policy of end-to-end network slicing based on complex network theory[J]. *IEEE access*, 2018, 6: 19691-19701.
- [21] Zhang N, Liu Y F, Farmanbar H, et al. Network slicing for service-oriented networks under resource constraints[J]. *IEEE Journal on Selected Areas in Communications*, 2017, 35(11): 2512-2521.
- [22] Zhang Q, Liu F, Zeng C. Adaptive interference-aware VNF placement for service-customized 5G network slices[C]//*IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019: 2449-2457.
- [23] Shu Z, Taleb T. A novel QoS framework for network slicing in 5G and beyond networks based on SDN and NFV[J]. *IEEE Network*, 2020, 34(3): 256-263.
- [24] Taleb T, Ksentini A. Follow me cloud: interworking federated clouds and distributed mobile networks[J]. *IEEE Network*, 2013, 27(5): 12-19.
- [25] Aissioui A, Ksentini A, Gueroui A M, et al. On enabling 5G automotive systems using follow me edge-cloud concept[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(6): 5302-5316.
- [26] Addad R A, Dutra D L C, Bagaa M, et al. Towards studying service function chain migration patterns in 5G networks and beyond[C]//*2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019: 1-6.
- [27] Addad R A, Dutra D L C, Bagaa M, et al. Fast service migration in 5G trends and scenarios[J]. *IEEE Network*, 2020, 34(2): 92-98.
- [28] Mada B E, Bagaa M, Tale T, et al. Latency-aware service placement and live migrations in 5G and beyond mobile systems[C]//*ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020: 1-6.
- [29] Zhao D, Sun G, Liao D, et al. Mobile-aware service function chain migration in cloud-fog computing[J]. *Future Generation Computer Systems*, 2019, 96: 591-604.
- [30] Addad R A, Dutra D L C, Taleb T, et al. AI-based network-aware Service Function Chain migration in 5G and beyond networks[J]. *IEEE Transactions on Network and Service Management*, 2021, 19(1): 472-484.
- [31] Wei F, Feng G, Sun Y, et al. Network slice reconfiguration by exploiting deep reinforcement learning with large action space[J]. *IEEE Transactions on Network and Service Management*, 2020, 17(4): 2197-2211.
- [32] Zhang J, Ye M, Guo Z, et al. CFR-RL: Traffic engineering with reinforcement learning in SDN[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(10): 2249-2259.
- [33] Zhang Q, Liu F, Zeng C. Online Adaptive interference-aware VNF deployment and migration for 5G network slice[J]. *IEEE/ACM Transactions on Networking*, 2021, 29(5): 2115-2128.
- [34] Benson T, Akella A, Maltz D A. Network traffic characteristics of data centers in the wild[C]//*Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 2010: 267-280. Kazmi S M A, Khan L U, Tran N H, et al. Network slicing for 5G and beyond

- networks[M]. Berlin: Springer, 2019.
- [35] Koster A M C A, Kutschka M. Network design under demand uncertainties: A case study on the abilene and GEANT network data[C]//Photonic Networks, 12. ITG Symposium. VDE, 2011: 1-8.
- [36] G. Dan and N. Carlsson, "Dynamic content allocation for cloud- assisted service of periodic workloads," in Proc. IEEE Conf. Comput. Commun. Apr. 2014, pp. 853–861.
- [37] Sony C S L, Cho K. Traffic data repository at the WIDE project[C]//Proceedings of USENIX 2000 Annual Technical Conference: FREENIX Track. 2000: 263-270.
- [38] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: Network processing as a cloud service," ACM SIGCOMM Comput. Commun. Rev., vol. 42, no. 4, pp. 13–24, Sep. 2012.
- [39] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.
- [40] Han B, Gopalakrishnan V, Ji L, et al. Network function virtualization: Challenges and opportunities for innovations[J]. IEEE communications magazine, 2015, 53(2): 90-97.
- [41] K. Pentikousis, Y. Wang, and W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks," IEEE Commun. Mag., vol. 51, no. 7, July 2013, pp. 44–53.
- [42] China Mobile Research Institute, "C-RAN The Road towards Green RAN," China Mobile White Paper, Oct. 2011.
- [43] ETSI, "Network Functions Virtualisation (NFV); Architectural Framework," GS NFV 002 (v. 1.1.1), Oct. 2013.
- [44] Xia W, Wen Y, Foh C H, et al. A Survey on Software-Defined Networking[J]. Communications Surveys & Tutorials IEEE, 2015, 17(1):27-51.
- [45] Mahajan A, Teneketzis D. Multi-armed bandit problems[J]. Foundations and applications of sensor management, 2008: 121-151.
- [46] Chen W, Wang Y, Yuan Y. Combinatorial multi-armed bandit: General framework and applications[C]//International conference on machine learning. PMLR, 2013: 151-159.
- [47] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50: 159-175.
- [48] Box G E P, Jenkins G M, Reinsel G C, et al. Time series analysis: forecasting and control[M]. John Wiley & Sons, 2015.
- [49] Hyndman R J, Athanasopoulos G. Forecasting: principles and practice[M]. OTexts, 2018.
- [50] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM[J]. Neural computation, 2000, 12(10): 2451-2471.
- [51] Müller B, Reinhardt J, Strickland M T. Neural networks: an introduction[M]. Springer Science & Business Media, 1995.
- [52] Mikolov T, Zweig G. Context dependent recurrent neural network language model[C]//2012 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2012: 234-239.
- [53] Taylor S J, Letham B. Forecasting at scale[J]. The American Statistician, 2018, 72(1): 37-45.
- [54] Sivaraman A, Subramanian S, Alizadeh M, et al. Programmable packet scheduling at line rate[C]//Proceedings of the 2016 ACM SIGCOMM Conference. 2016: 44-57.
- [55] Farris I, Taleb T, Khettab Y, et al. A survey on emerging SDN and NFV security mechanisms for IoT systems[J]. IEEE Communications Surveys & Tutorials, 2018, 21(1): 812-837.

- [56] Afolabi I, Taleb T, Samdanis K, et al. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions[J]. *IEEE Communications Surveys & Tutorials*, 2018, 20(3): 2429-2453.
- [57] Shu Z, Taleb T, Song J. Resource allocation modeling for fine-granular network slicing in beyond 5G systems[J]. *IEICE Transactions on Communications*, 2022, 105(4): 349-363.
- [58] Prados-Garzon J, Taleb T. Asynchronous time-sensitive networking for 5G backhauling[J]. *IEEE Network*, 2021, 35(2): 144-151.
- [59] Abbou A N, Taleb T, Song J S. Towards SDN-based Deterministic Networking: Deterministic E2E Delay Case[C]//2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021: 1-6.
- [60] Chowdhury S R, Bari M F, Ahmed R, et al. Payless: A low cost network monitoring framework for software defined networks[C]//2014 IEEE Network Operations and Management Symposium (NOMS). IEEE, 2014: 1-9.
- [61] Van Adrichem N L M, Doerr C, Kuipers F A. Opennetmon: Network monitoring in openflow software-defined networks[C]//2014 IEEE Network Operations and Management Symposium (NOMS). IEEE, 2014: 1-8.
- [62] Clemm A, Chandramouli M, Krishnamurthy S. DNA: An SDN framework for distributed network analytics[C]//2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015: 9-17.
- [63] Mestres A, Rodriguez-Natal A, Carner J, et al. Knowledge-defined networking[J]. *ACM SIGCOMM Computer Communication Review*, 2017, 47(3): 2-10.
- [64] Bega D, Gramaglia M, Banchs A, et al. Optimising 5G infrastructure markets: The business of network slicing[C]//IEEE INFOCOM 2017-IEEE conference on computer communications. IEEE, 2017: 1-9.
- [65] Sciancalepore V, Samdanis K, Costa-Perez X, et al. Mobile traffic forecasting for maximizing 5G network slicing resource utilization[C]//IEEE INFOCOM 2017-IEEE conference on computer communications. IEEE, 2017: 1-9.
- [66] Yang T, Zhang H, Li J, et al. HeavyKeeper: an accurate algorithm for finding Top- k elephant flows[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(5): 1845-1858.
- [67] Audibert J Y, Bubeck S, Munos R. Best arm identification in multi-armed bandits[C]//COLT. 2010: 41-53.
- [68] Bubeck S, Wang T, Viswanathan N. Multiple identifications in multi-armed bandits[C]//International Conference on Machine Learning. PMLR, 2013: 258-265.
- [69] Zhang M, Ong C S. Quantile bandits for best arms identification[C]//International Conference on Machine Learning. PMLR, 2021: 12513-12523.
- [70] Zhuang H, Wang C, Wang Y. Identifying outlier arms in multi-armed bandit[J]. *Advances in Neural Information Processing Systems*, 2017, 30.
- [71] Ksentini A, Taleb T, Letaif K B. QoE-based flow admission control in small cell networks[J]. *IEEE Transactions on Wireless Communications*, 2015, 15(4): 2474-2483.
- [72] Taleb T, Ksentini A. VECOS: A vehicular connection steering protocol[J]. *IEEE Transactions on Vehicular Technology*, 2014, 64(3): 1171-1187.
- [73] Allesiaro R, Féraud R, Maillard O A. The non-stationary stochastic multi-armed bandit problem[J]. *International Journal of Data Science and Analytics*, 2017, 3: 267-283.
- [74] Besbes O, Gur Y, Zeevi A. Stochastic multi-armed-bandit problem with non-stationary

- rewards[J]. *Advances in neural information processing systems*, 2014, 27.
- [75] Lin T, Li J, Chen W. Stochastic online greedy learning with semi-bandit feedbacks[J]. *Advances in Neural Information Processing Systems*, 2015, 28.
- [76] Lattimore T, Szepesvári C. *Bandit algorithms*[M]. Cambridge University Press, 2020.
- [77] Sun L, Hou J, Shu T. Spatial and temporal contextual multi-armed bandit handovers in ultra-dense mmWave cellular networks[J]. *IEEE Transactions on Mobile Computing*, 2020, 20(12): 3423-3438.
- [78] Sutton R S, Barto A G. *Introduction to reinforcement learning*[M]. Cambridge: MIT press, 1998.
- [79] Eramo V, Lavacca F G, Catena T, et al. Application of a Long Short Term Memory neural predictor with asymmetric loss function for the resource allocation in NFV network architectures[J]. *Computer Networks*, 2021, 193: 108104.
- [80] Mijumbi R, Serrat J, Gorricho J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions[C]//*Proceedings of the 2015 1st IEEE conference on network softwarization (NetSoft)*. IEEE, 2015: 1-9.
- [81] Dolati M, Hassanpour S B, Ghaderi M, et al. DeepVINE: Virtual network embedding with deep reinforcement learning[C]//*IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019: 879-885.
- [82] Ebrahimi S, Zakeri A, Akbari B, et al. Joint resource and admission management for slice-enabled networks[C]//*NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020: 1-7.
- [83] Sedgewick R. *Algorithms in C, part 5: graph algorithms*[M]. Pearson Education, 2001.
- [84] Wang G, Feng G, Quek T Q S, et al. Reconfiguration in network slicing—Optimizing the profit and performance[J]. *IEEE Transactions on Network and Service Management*, 2019, 16(2): 591-605.
- [85] Wang G, Feng G, Tan W, et al. Resource allocation for network slices in 5G with network resource pricing[C]//*GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017: 1-6.
- [86] Wen R, Feng G, Tang J, et al. On robustness of network slicing for next-generation mobile networks[J]. *IEEE Transactions on Communications*, 2018, 67(1): 430-444.
- [87] He Q, Moayyedi A, Dán G, et al. A meta-learning scheme for adaptive short-term network traffic prediction[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(10): 2271-2283.
- [88] Siami-Namini S, Tavakoli N, Namin A S. A comparison of ARIMA and LSTM in forecasting time series[C]//*2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2018: 1394-1401.

攻读学位期间的学术论文与研究成果

■ 学术论文

- [1] Z. Shu, **H. Feng**, T. Taleb, and Z. Zhang, "A Novel Combinatorial Multi-Armed Bandit Game to Identify Online the Changing top-K Flows in Software-Defined Networks," submitted to Computer Networks (accepted).
- [2] **H. Feng**, Z. Shu, T. Taleb, Y. Wang and Z. Liu, "An Aggressive Migration Strategy for Service Function Chaining in the Core Cloud," in IEEE Transactions on Network and Service Management. doi: 10.1109/TNSM.2022.3231186.
- [3] **Feng H**, Shu Z, Zhang Z, et al. Bandwidth Resources Prediction for Dynamic Networking Slicing Based on Time Series Analysis[C]//2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM). IEEE, 2022: 1-5.
- [4] **Feng H**, Shu Z, Zhang Z, et al. Predict Critical Node Pairs Based on Reinforcement Learning in SDN[C]//2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT). IEEE, 2022: 979-982.

■ 科研项目

- [1] 福建省自然科学基金项目，“基于 SDN/NFV 的面向 5G 切片的网络资源动态分配与优化研究”，2020J01574，2021.6-2023.12。
- [2] 国家产学研创新基金，“基于机器学习的跨域 5G 网络切片机制研究”，2021FNA05003，2022.3-2023.2。
- [3] 教育部 CERNET 下一代 IPv6 科研项目，“基于 ONOS 控制器的 IPv6 与 IPv4 互联平台”，NGII20190102，2019.12-2020.12。

■ 知识产权与获奖

- [1] 发明专利：舒兆港，**冯浩贤**，陈日清. 基于 SDN 的 IPv4 及 IPv6 的管理系统及方法. 2022.12.09，中国，ZL 2020 1 1427051.4

致 谢

光阴似箭，岁月如梭，短短的三年硕士学习生活即将过去，在这即将毕业之际，思绪万千，感慨良多！尽管在科研的道路上遇到了非常多难以叙说的困难与挑战，但众多老师、朋友和家人的关心和帮助却铭记在心，在此致以最真诚的感谢！

首先衷心感谢恩师舒兆港教授，您严谨的治学作风、兢兢业业的科研态度、高瞻远瞩的科研眼光让学生深感敬佩，是学生长期学习的楷模；您对弟子的关怀备至和语重心长，极大地鼓舞着我在今后的科研和生活道路上奋勇前进。舒兆港教授在我论文研究方向确定、论文选题、资料查阅、论文总体框架建立、关键问题解决、论文写作与定稿等各个阶段，都付出了大量的心血与劳动；舒兆港教授在我研究方向选择、论文写作与修改的过程给予了很多宝贵的意见，让我大有收获、深受启发。若没有他悉心的指导，本文不可能如此顺利地完成。他敏锐的眼光、敬业的精神、不断的求知探索与实践，极大激励着我勇登科学的高峰。他待我亦师亦友，不仅把我引入计算机网络研究领域的大门，还教会了我许多为人处事的道理，学生会永远记住您的栽培！知遇之恩，学生永志难忘！

此外，我由衷地感谢刘仲健老师对我的帮助。在我本科及研究生的生涯中，刘教授都给予了我极大的鼓舞。在生活上，刘教授对我的关心无微不至，在科研生活中，刘教授向我娓娓道来他所做过的科学研究，为我树立起正确的科学观，在遇到困难时，是您强有力的双手，引领我走出困境。您是我科研路上的灯塔，学生会永远铭记您的恩情！

我还需要感谢我的家人。没有父母的养育之恩，姐姐和各位表哥的关怀，科研道路我很难坚持得住。感谢你们的爱与陪伴，你们是最坚强的后盾。

最后，感谢各挚友对我的关心，感谢你们在我寝食难安的岁月中给予我的关怀！