

分类号:
密 级:

单位代码: 10389
学 号: 1221193002



福建农林大学

硕士学位论文

(学术学位)

基于深度强化学习的 SDN 网络 负载均衡研究

学 习 形 式: 全日制

一级学科名称: 计算机科学与技术

二级学科名称:

研 究 方 向: 计算机网络与信息安全

研究生姓名: 邱彩钰

指 导 教 师: 舒兆港 副教授

完 成 时 间: 二〇二五年五月

Research on SDN Network Load Balancing Based on Deep Reinforcement Learning

By

Caiyu Qiu

Supervised by Associate Professor 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 (May, 2025)

Commencement Date (May, 2025)

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：邱彩钰

日期：2015年5月23日

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：邱彩钰

日期：2015年5月23日

指导教师亲笔签名：[Signature]

日期：2015年5月23日

目 录

摘 要	V
ABSTRACT	VI
1 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 基于静态方法的 SDN 网络负载均衡	3
1.2.2 基于动态方法的 SDN 网络负载均衡	3
1.3 研究内容和组织架构	6
2 相关理论基础	8
2.1 软件定义网络	8
2.1.1 SDN 架构简介	8
2.1.2 QoS 保障机制	10
2.2 深度强化学习概述	12
2.2.1 深度学习	12
2.2.2 强化学习	13
2.2.3 深度强化学习	16
2.3 模糊层次分析	17
2.4 最小费用最大流问题	17
2.4.1 网络最大流问题基本概念	18
2.4.2 最小费用最大流问题的描述与解决方法	20
2.5 本章小结	20
3 基于模糊层次分析的流转发路径探索算法	21
3.1 引言	21
3.2 问题描述与建模	21
3.2.1 问题描述	22
3.2.2 系统框架	22
3.2.3 问题建模	23
3.3 流转发路径探索算法	25
3.3.1 算法描述	26
3.3.2 算法复杂性分析	30
3.4 仿真设计与性能分析	30
3.4.1 实验环境和参数设置	31
3.4.2 对比算法介绍	32

3.4.3 评价指标	32
3.4.4 实验结果与分析	32
3.5 本章小结	34
4 基于最小费用最大流的转发路径动态调整算法	36
4.1 引言	36
4.2 问题描述与建模	36
4.2.1 问题描述	36
4.2.2 系统框架	36
4.2.3 问题建模	38
4.3 流转发路径动态调整算法	41
4.3.1 算法描述	43
4.3.2 复杂性分析	47
4.4 仿真设计与性能分析	48
4.4.1 实验环境与参数设置	48
4.4.2 对比算法介绍	49
4.4.3 评价指标	49
4.4.4 实验结果与分析	49
4.5 本章小结	51
5 总结和展望	52
5.1 总结	52
5.2 展望	52
参考文献	54
攻读学位期间的学术论文与研究成果	59
致谢	60

图录

图 1-1 SDN 架构下基于负载均衡技术的方法分类.....	3
图 1-2 论文的组织结构与章节安排.....	7
图 2-1 SDN 架构.....	9
图 2-2 人工神经元的数学模型	12
图 2-3 多层神经网络示意图	13
图 2-4 智能体学习过程示意图	15
图 2-5 RL 和 DL 结合示意图.....	16
图 2-6 AHP 层次结构图	17
图 2-7 容量网络.....	18
图 2-8 截集示意图.....	19
图 2-9 增广链示意图.....	19
图 3-1 SDN 路由框架.....	22
图 3-2 系统处理模型框架	23
图 3-3 流转发路径探索算法流程图.....	25
图 3-4 请求的 FAHP 层次结构.....	26
图 3-5 负载均衡系数 k 的对比	33
图 3-6 转发成本的对比.....	33
图 3-7 优化总目标的对比	34
图 3-8 请求迟到数的对比	34
图 4-1 SDN 路由框架.....	37
图 4-2 系统处理模型框架	38
图 4-3 流转发路径动态调整算法流程图	41
图 4-4 MCMF 流程图.....	42
图 4-5 负载均衡系数 k 的对比	49
图 4-6 转发成本的对比.....	50
图 4-7 优化总目标的对比	50
图 4-8 时延的对比.....	51

表录

表 2-1 DRL 分类.....	16
表 3-1 主要参数和意义.....	24
表 3-2 标度及含义.....	27
表 3-3 实验参数.....	31
表 3-4 处理顺序对比.....	32
表 4-1 主要参数及意义.....	40
表 4-2 实验参数设置.....	48

摘 要

随着网络应用程序的急速增长,网络流量需求不断加剧,传统的网络结构难以应对日益膨胀的流量和需求变化的压力。软件定义网络(Software Defined Network, SDN)作为新型网络架构,已被应用在许多场景中,其在提高网络灵活性等方面表现出巨大潜力。然而,随着网络流量的爆发式增长,以及网络用户在异构网络下的不同需求,SDN 也面临着如何合理分配网络带宽与流量路径选择的挑战,主要表现为流量负载不平衡导致网络拥塞,从而降低网络服务质量,特别是多媒体流服务应用场景中的用户体验。

为了缓解 SDN 在多媒体流服务场景中,报文流请求转发过程中因优先级划分不合理而导致网络链路拥塞、链路负载不均衡,以及链路转发成本开销过大的问题,本文主要引入深度强化学习技术,研究更为有效的流量负载均衡方法。本文的主要研究内容和创新之处如下:

1、针对已有方法无差别地对待每个报文流,导致部分请求因优先级划分不合理而超出时限要求,引起网络链路拥塞的问题,本文提出了一种基于模糊层次分析(Fuzzy Analytic Hierarchy Process, FAHP)的流转发路径探索算法。算法首先综合考虑不同类型请求的服务质量(Quality of Service, QoS)需求,如带宽、时延、请求类型等,通过 FAHP 进行请求间的两两比较,计算各需求因素的权重从而得到请求的优先级。然后按照优先级顺序为请求探索转发路径,引入深度 Q 网络(Deep Q-Network, DQN)进行节点探索,并基于高链路负载均衡程度、低转发成本原则来调整当前路径,最终得到所有请求的转发路径策略。实验表明,与已有的 QLLB(Q-Learning Load Balance)和 HiQoS(Hierarchy Quality of Service)算法相比,该算法所得的负载均衡系数和转发成本之和分别降低 6.49%和 9.74%。

2、在大规模流量并发的情况下,依靠逐点探索路径的方法难以协调全局资源,在流量调度过程中可能导致请求超时以及链路负载不均衡的问题,当大量请求集中在相同链路上时,也可能会进一步增加整体转发成本。为此,本文提出了一种基于最小成本最大流(Minimum Cost Maximum Flow, MCMF)和深度强化学习协同优化的全局流量调度方法。算法首先利用 MCMF 计算全局流量分配方案,确保所有请求在满足带宽和时延需求的前提下,获得较优的初始转发路径。随后,结合实时网络状态,采用深度强化学习对流量调度策略进行动态调整,以适应流量变化,实现更合理的链路负载均衡与转发成本优化。

综上所述,本文面向 SDN 中多媒体流服务场景下的流量负载均衡问题,分别从基于优先级的逐点路径探索与全局路径分配后的策略动态调整两个层面,设计了相应的优化算法,并以负载均衡系数、转发成本与时延等指标对性能进行评估。实验结果表明,与现有算法相比,所提方法在提高网络资源利用率、降低路径开销方面表现出更优性能,尤其在复杂网络环境下展现出更强的调度效率与适应能力。

关键词: SDN 负载均衡; 深度强化学习; 流量分配; 路径选择; 多目标优化

ABSTRACT

With the rapid growth of network applications, the demand for network traffic is constantly increasing. Traditional network structure is difficult to cope with the growing expansion of traffic and demand changes. Software defined network (SDN), as a new network architecture, has been applied in many scenarios and has shown great potential in improving network flexibility and other aspects. However, with the explosive growth of network traffic and the different needs of network users in heterogeneous networks, SDN also faces the challenge of how to allocate network bandwidth and choose traffic paths reasonably, mainly manifested as network congestion caused by imbalanced traffic load, thereby reducing network service quality, especially the user experience in multimedia streaming service application scenarios.

In order to alleviate the problems of network link congestion, link load imbalance, and excessive cost of link forwarding caused by unreasonable priority allocation during the request forwarding process of SDN in multimedia streaming services, this paper mainly introduces deep reinforcement learning technology to study more effective traffic load balancing methods. The main research content and innovations of this article are as follows:

1. To address the issue where existing methods treat each data flow indiscriminately—causing some requests to exceed latency requirements due to improper priority classification and leading to network congestion—this paper proposes a flow forwarding path exploration algorithm based on the Fuzzy Analytic Hierarchy Process (FAHP). The algorithm first comprehensively considers the Quality of Service (QoS) requirements of different request types, such as bandwidth, latency, and request category. Using FAHP, it performs pairwise comparisons between requests to calculate the weights of various QoS factors and determine the priority of each request. Forwarding paths are then explored in priority order. A Deep Q-Network (DQN) is introduced for node exploration, and the current path is adjusted based on the principles of high link load balancing and low forwarding cost. Finally, the forwarding path strategy for all requests is obtained. Experimental results show that, compared with existing algorithms such as QLLB (Q-Learning Load Balance) and HiQoS (Hierarchy Quality of Service), the proposed method reduces the sum of the load balancing coefficient and forwarding cost by 6.49% and 9.74%, respectively.

2. In the case of large-scale concurrent traffic, relying on point by point exploration of paths is difficult to coordinate global resources, which may lead to the problem of request timeout and unbalanced link load in the process of traffic scheduling. When a large number of requests are concentrated on the same link, it may further increase the overall forwarding cost. To this end, this article proposes a global traffic scheduling method based on Minimum Cost Maximum Flow (MCMF) and deep reinforcement learning collaborative optimization. The algorithm first uses MCMF to calculate the global traffic allocation scheme, ensuring that all requests obtain optimal initial forwarding paths while meeting bandwidth and latency requirements. Subsequently, combined with real-time network status, deep reinforcement learning is used to dynamically adjust traffic scheduling strategies to adapt to traffic changes and achieve more reasonable link load balancing and forwarding cost optimization.

In summary, this paper focuses on the problem of traffic load balancing in SDN under multimedia streaming service scenarios. Two optimization algorithms are designed from the

perspectives of priority-based point-by-point path exploration and dynamic adjustment after global path allocation. The performance is evaluated using metrics such as load balancing coefficient, forwarding cost, and latency. Experimental results show that, compared with existing algorithms, the proposed methods achieve better performance in improving network resource utilization and reducing path overhead, and exhibit stronger scheduling efficiency and adaptability, especially in complex network environments.

KEY WORDS: SDN load balancing; deep reinforcement learning; traffic distribution; path selection; multi-objective optimization

1 绪论

1.1 研究背景及意义

在计算机网络领域，负载均衡可优化流量路径，减少网络拥塞，提升数据传输效率与服务质量（Quality of Service, QoS）。伴随互联网的快速发展，网络规模不断扩大，流量激增，应用类型也日益丰富多样。而随着应用程序的急速增长，流量需求不断加剧，网络需求不断变化，用户对 QoS 的要求变得更加严格。一方面，不同的应用业务对网络的底层资源有着多样化的需求。另一方面，用户对网络的期望已经从简单的连接转换为对高品质服务的需求。无论是在线视频、音频流、在线游戏还是其他实时应用，都对网络的性能和负载均衡提出了极高的要求。用户希望能够快速、稳定地访问各种服务，这就需要网络能够在高负载情况下依然保持卓越的性能。而高流量可能导致负载不均衡或网络拥塞，从而降低网络的性能和效率。因此，采用高效的路由和负载均衡模型来解决这些挑战性问题至关重要。

然而传统的负载均衡方法难以适应网络环境的动态性和不确定性，流量的不断变化、网络拓扑的调整等因素都带来了负载均衡问题的复杂性。传统路由中嵌入的传统拥塞避免和控制机制对新一代网络难以迅速适应。软件定义网络（Software Defined Network, SDN）已成为一种有效的解决方案^[1]。然而，尽管 SDN 依靠集中控制实现了网络管理的灵活性，但面对网络环境的动态变化，它在带宽分配和流量路径选择上仍存在诸多挑战。首先，流量在实际网络中的分布往往具有高度的动态性和不确定性，尤其在多媒体流服务等对时延和带宽敏感的应用场景中，突发性的流量增长和业务请求的不均衡可能会导致部分链路在短时间内出现拥塞，而其他链路资源则被闲置，严重影响整体网络的负载均衡效率。其次，随着网络规模的扩大，SDN 控制器需要计算全局最优的流量调度策略，计算和决策开销随之增加。此外，SDN 在流量调度时需要兼顾全局最优和局部最优的平衡，以适应网络状态的实时变化。不同应用场景对 QoS 的要求各不相同，如多媒体流服务需要低时延，而大数据传输则关注高吞吐，这使得 SDN 在带宽分配和路径选择时面临更复杂的优化问题。随着人工智能与机器学习的发展，智能负载均衡成为研究热点，通过流量分配与自适应调度，提高系统自主优化能力。因此，在这种动态且不确定的环境下，如何利用智能化方法进行更精准、高效的负载均衡成为一个关键研究方向。

本文针对 SDN 在多媒体流服务场景中的带宽分配与流量路径选择问题，设

计有效的负载均衡策略，以提升 QoS 性能并优化计算成本。

1.2 国内外研究现状

随着 SDN 的兴起，国内外学者对 SDN 架构下网络的负载均衡进行了深入研究。他们关注于如何利用 SDN 的灵活性和可编程性，设计高效的负载均衡算法，以提高网络性能和适应不同应用场景^[2-5]。

从 SDN 架构下网络负载均衡的研究内容上看，当前网络负载均衡研究中，主要包括网络流量负载均衡和网络链路负载均衡两个方面内容^[6, 7]。在 SDN 环境下，研究人员致力于如何通过 SDN 控制器实现流量的动态调度和智能负载分配^[8]。流量负载均衡关注数据在网络中的分发与调度，流量负载均衡策略能够实时感知网络的负载状态，并动态调整数据包的路径，以实现网络资源的高效利用。链路负载均衡主要研究网络链路的资源分配和调度。研究人员关注数据在网络链路上传输的均衡性，通过设计智能路由、流量调度和负载分担机制，以确保网络链路的高效利用和避免瓶颈的出现。这包括对数据包的动态路由、拥塞控制等方面的研究。网络流量负载均衡与链路负载均衡两者相辅相成、相互影响。流量负载均衡更偏重于全局视角下的数据流路径选择和调度决策，而链路负载均衡则更侧重于局部链路资源的实际使用情况和负载状态。二者结合能够实现从全局到局部的协同优化：通过全局流量调度策略合理分配路径，同时在具体链路层面进行资源调控，提升整个网络系统的性能与可靠性。为此，研究人员也不断探索如何将负载感知机制、动态拓扑调整 and 智能算法应用于两者之间的协同优化中^[9]。

从网络负载均衡方法的角度上看，在现有研究中，SDN 网络负载均衡方法可以从多个角度进行分类，例如按控制策略（基于规则、基于学习）、决策层次（控制层面、数据层面）、实现方式（集中式、分布式）等，这些分类方式反映了不同研究在设计理念、实施策略与系统结构上的差异。然而，这些分类方式存在一定的交叉关系，例如，基于规则的负载均衡策略通常是静态的，而基于学习的策略往往采用动态调整机制，同时，基于学习的方法也可部署于控制层或数据层，形成不同的优化架构。

为了更直观地分析 SDN 网络负载均衡方法，全面了解网络负载均衡研究现状，本文首先基于负载均衡技术的分类标准，将现有网络负载均衡方法划分为静态负载均衡方法和动态负载均衡方法^[10]，以对现有的 SDN 网络负载均衡方法进行系统总结和对比分析，SDN 架构下基于负载均衡技术的方法分类如图 1-1 所示。接着，本文将进一步聚焦于动态方法中以深度强化学习（Deep Reinforcement Learning, DRL）为核心的研究路径，对现有基于深度强化学习的网络负载均衡研究进展进行梳理。分析当前基于深度强化学习的 SDN 网络负载均衡方法的应用

优势与不足，总结这类方法在不同场景下的适用性与扩展潜力，以为后续算法优化与系统实现提供坚实的理论支撑与技术参考。

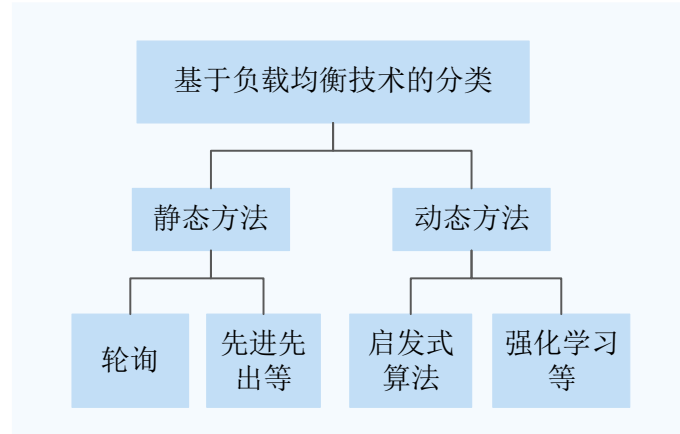


图 1-1 SDN 架构下基于负载均衡技术的方法分类

1.2.1 基于静态方法的 SDN 网络负载均衡

在网络负载均衡研究中，静态负载均衡方法主要关注在网络运行时不考虑流量动态变化的情况下，在初始阶段进行流量分配和路径规划。静态负载均衡策略不依赖于网络的当前负载状态，而是基于网络拓扑结构、链路带宽、交换机处理能力等静态属性进行流量调度。这些策略不允许在流量转发过程中动态调整资源分配，因而易于实现，并适用于流量模式稳定的小规模网络。然而，由于静态方法不考虑网络实时负载信息，在复杂和动态变化的网络环境中，其性能往往受限。一些常见的静态网络负载均衡方法包括先进先出(First-In First-Out, FIFO)^[11, 12]、轮询法(Round Robin)^[13-15]、MinMax^[16, 17]和 MaxMin^[18]。在静态负载均衡策略下，所有流量调度决策均基于预先收集的网络拓扑和流量信息，因此在初始阶段能够减少计算开销和决策时间。然而，这些方法在处理突发流量或网络拓扑变化时可能无法快速适应，导致链路过载或资源浪费。此外，由于静态策略未能根据实际流量动态调整路径，部分流量可能出现长时间的排队等待，影响服务质量。由于静态负载均衡策略不依赖于网络的当前状态，因此它们在状态主动发生变化的分布式云计算环境中，被证明并不实用^[19]。

1.2.2 基于动态方法的 SDN 网络负载均衡

为弥补静态方法反应迟缓、缺乏适应性的不足，一些研究尝试将静态策略与动态机制相结合，形成动静结合的负载均衡方法。此类方法通常在网络初始阶段通过静态策略进行基本资源分配，再在运行过程中引入动态优化机制，实时调整

负载状态,从而兼顾稳定性与灵活性。胡逸飞等人^[20]提出了一种在改进遗传算法基础上动静态结合的负载均衡算法。该算法在低负载状态下采用静态加权轮询算法,根据网络拓扑和链路容量进行流量均衡;在高负载情况下,则基于 CPU 利用率、交换机缓存占用率、链路带宽等实时指标,动态调整流量调度策略,以优化整体网络性能。虽然动静结合策略在一定程度上提高了系统响应能力和资源利用效率,但其设计与实现复杂度较高,且在处理大规模、多业务类型并发请求时仍存在一定瓶颈。因此,研究重心逐渐转向更具智能性、自适应能力更强的动态负载均衡方法。总体而言,动态负载均衡研究致力于通过智能、自适应的方法,使系统更好地适应实时的、不断变化的负载条件,以提高链路利用率、吞吐量和服务质量。研究人员关注如何在 SDN 环境下实现基于网络资源的实时负载均衡,以提升流量调度效率和资源利用率^[21],确保数据流均衡分布,减少网络时延和拥塞。在任务感知方面,研究人员关注如何基于流量类型、数据传输需求和 QoS 要求,智能地分配网络资源。例如,对视频流、游戏流和文件传输等不同流量采用差异化的调度策略,以优化用户体验^[22]。基于反馈的动态负载均衡是网络负载均衡的关键研究方向之一。通过监测流量状态、链路负载、队列积压情况等实时指标,动态调整流量分配策略,以适应网络的变化。例如,基于带宽预测和流量趋势分析的负载均衡方法,可以在高负载链路出现拥塞前,提前调整流量路由,降低丢包率和时延。此外,在考虑环境友好性的趋势下,研究人员致力于设计能耗感知的负载均衡算法,以在维持性能的同时降低系统的能耗^[23]。受自然现象启发的智能优化算法也已被广泛应用于网络负载均衡,例如遗传算法、蚁群算法和蜂群算法等。Ragmani 等人^[24]提出了一种基于模糊逻辑和蚁群算法的混合算法。该算法利用 Taguchi 实验设计优化参数,以提高网络响应时间和吞吐量。Aloboud 等人^[25]提出了一种受布谷鸟寄生行为启发的网络负载调度算法。通过将低优先级流量拆分为多个子任务分配到不同链路上,以提高资源利用率并避免高优先级任务完全抢占低优先级任务的带宽。算法通过“寻找适合链路”的方式,保证了流量转发的灵活性,同时在高优先级任务到来时能够处理抢占情况。Pradhan 等人^[26]提出了一种结合并行粒子群优化算法和 DRL 技术的混合方法,以降低网络时延和成本,同时提高负载均衡效率。

可以发现,人工智能技术,尤其是与其他优化方法的结合,已被广泛应用于网络负载均衡领域。利用机器学习技术预测负载变化和优化负载均衡策略,有助于提高决策的准确性和网络的自适应能力,已成为当前研究的重要方向^[27]。

其中, DRL 因其无需依赖先验知识或精确数学模型,在复杂动态环境下依然能够高效决策和优化,展现出显著优势。得益于其优异的环境适应能力, DRL 已被集成到多种网络负载均衡优化方案中,以应对复杂和动态变化的流量调度问题。因此,在 SDN 架构下应用 DRL 方法优化网络资源利用率、动态调整流量调

度策略，并提升整体网络性能，正成为当前网络负载均衡研究的热点。

Sanchez 等人^[28]考虑链路和队列指标的多目标函数驱动，在深度强化学习智能体代理的指导下，通过智能分配流量来优化路由决策，考虑了带宽链路利用率、端到端时延和丢包率等指标。但该方法主要依赖于历史数据训练，对于突发流量的适应能力较弱。Zhang 等人^[29]提出了基于强化学习的关键流重路由方案，学习为每个给定流量矩阵自动选择关键流的策略，最后通过制定和解决线性规划问题，重新路由这些选定的关键流，以平衡网络的链路利用率，但该方法仅对关键流进行优化，可能导致部分次要流量的 QoS 下降。Rezaei 等人^[30]提出了一种多任务学习框架，将对流的带宽和持续时间的预测与流量分类相结合。根据流量指标采用五种不同的流量类别来进行分类，以增强网络性能。该方法主要依赖于预定义的流量类别，对于未知流量类型或突发性流量，分类精度可能降低，影响负载均衡效果。Raikar 等人^[31]应用三种监督学习模型对软件定义网络架构中的数据流量进行分类，通过捕获流量并创建传递给强化学习分类器进行预测的流量特征来研究网络流量，有助于实现网络资源的最佳利用。但该方法主要针对流量分类，对流量路径的实时优化能力有限。Samal 等人^[32]提出了一种基于无线体域网（Wireless Body Area Network, WBAN）的流量优先级负载均衡调度算法（Traffic Priority Load Balancing Scheduling, TPLBS），在该方法中，每个个性化设备（Personalized Devices, PD）维护本地和远程队列，并根据数据包的紧急程度将其分类为紧急、周期性、非周期性和非医疗四类流量。通过动态计算通信和计算负载，PD 选择负载最低的下流节点进行数据转发，同时采用概率优先级调度机制优先处理远程队列中的高优先级数据，从而在保证紧急数据低延迟传输的同时实现多 WBAN 的负载均衡。然而，该方法在数据包优先级的考量上仅考虑时延因素，此外，在动态调整缓冲区大小时可能引入额外的计算开销，且未考虑移动场景下网络拓扑变化对负载均衡的影响。Ke 等人^[33]提出基于深度强化学习的 SDN 大流量路由策略，优先调度大流量并自适应生成路由策略。该方法主要针对大流量优化，缺乏对不同流量类型的精细优先级管理。Liu 等人^[34]提出了使用多智能体深度强化学习的在线路由算法来解决多类型服务需求的在线路由优化问题，但学习过程收敛速度较慢。Ruelas 等人^[35]在知识定义网络的背景下，使用人工神经网络根据请求的延迟和带宽指标来预测网络性能，以选择负载最小的路径。Ejaz 等人^[36]提出了一种基于 SDN 和网络功能虚拟化（Network Function Virtualization, NFV）协同的流量负载均衡方法，通过将 SDN 控制器虚拟化为 VNF（vSDN）并在检测到拥塞时动态创建次级 vSDN 来分担负载，实验表明该方法能降低平均负载和延迟。然而，该方法还存在动态扩展的实时性不足的挑战，该方法中 vSDN 复制依赖 NFV 编排流程，可能引入延迟，尤其在突发流量场景下难以保证即时响应。

基于上述分析,当前针对 SDN 架构下基于深度强化学习的网络负载均衡研究中,在数据层面,研究人员更多关注于流量转发的高效性,在控制层面则关注路由决策算法的优化^[37]。当前研究中关注流量处理(如:流量分类、设置流量优先级)的网络负载均衡方法大多弱化了路由策略的考虑^[29-33],且在请求优先级设置方面通常仅依据流量类型或紧急程度,考虑标准较为单一,难以全面反映不同请求在带宽、时延等多维 QoS 需求上的差异。针对控制层面提出优化,为 SDN 控制器制定路由策略的网络负载均衡方法,在共享有限资源时大多平等对待所有数据流,因而无法保证每个数据流的 QoS 需求^[35, 36]。从评价指标和优化目标来看,现有研究多聚焦于均衡带宽利用率、降低端到端时延等目标,但忽略了带宽利用率平衡与路径转发成本之间存在的矛盾关系。在追求链路带宽利用率均衡的同时,可能导致路径过长等问题,从而增加整体转发成本。

1.3 研究内容和组织架构

根据 SDN 网络负载均衡问题的研究现状和发展趋势,当前大多数 SDN 网络负载均衡策略仍以单一层面优化为主,对控制层面的路由优化策略,在共享有限资源时平等对待所有数据流,缺乏对不同应用程序 QoS 需求的区分,难以实现服务质量的差异化保障;针对数据层面的优化策略,大多仅针对数据流的分类、优先级划分等处理展开研究,而忽略控制决策层面的优化。此外,从优化目标来看,当前多数方法侧重于提升链路带宽利用率或降低平均时延,但往往忽视了路径转发成本的显式建模与控制。在实际网络中,链路负载均衡与转发成本常常存在冲突:过度追求链路利用率均衡可能导致路径冗长,从而显著增加网络的调度和传输开销。因此,本文将针对控制层面和数据层面的同步优化展开 SDN 的流量负载均衡策略研究。在应用程序急速增长、数据流需求多样性的背景下,本文针对 SDN 在多媒体流服务场景中的带宽分配与流量路径选择问题,研究有效的基于 DRL 的 SDN 网络负载均衡算法,通过 DRL 优化 SDN 中的流量调度、路径选择和资源分配策略,以达到提升 QoS 性能并优化计算成本的研究目标。

本文的具体研究内容和各个章节安排如图 1-2 所示:

第一章首先介绍了 SDN 架构下网络负载均衡的研究背景与研究意义,指出了负载均衡问题的重要性。其次总结了当前针对网络负载均衡问题的国内外研究现状。最后介绍了本文的研究内容和组织架构。

第二章首先阐述了 SDN 的架构及其优势,同时介绍了 QoS 相关理论,分析其在网络流量调度中的关键作用。随后,针对深度强化学习进行了概述,包括深度学习、强化学习及深度强化学习的基本概念。此外,本章介绍了模糊层次分析,为优先级调度策略奠定基础。最后,讨论了最小费用最大流问题的基本概念及求

解方法，将其引入网络流量优化中。这些相关基础的介绍为本章后续工作提供了支撑。

第三章介绍了一种基于模糊层次分析的流转发路径探索算法，该算法利用模糊层次分析法（Fuzzy Analytical Hierarchy Process, FAHP）计算不同请求的 QoS 需求权重，确定请求的优先级。其次，引入深度 Q 网络（Deep Q-Network, DQN）根据优先级进行转发节点探索，将链路带宽利用率的标准差和转发成本通过奖励函数设置，以权衡链路负载均衡和转发成本。与已有的 QLLB 和 HiQoS 算法相比，所提算法在负载均衡系数和转发成本的最小化目标上具有更好效果。

第四章提出了一种基于最小费用最大流（Minimum Cost Maximum Flow, MCMF）的流转发路径动态调整算法，以缓解在大规模流量并发的情况下，仅依靠逐点优化的方式可能难以从全局视角协调多个流的资源调度的问题，并优化链路资源利用率。首先，计算 MCMF 路径作为强化学习的初始解，以最小化转发成本并最大化网络流量。其次，结合深度强化学习方法，根据实时网络状态调整转发路径，实现全局近似最优的路径调整策略。

第五章对本文的工作进行了总结，并基于对本文算法的分析展望了未来可能的研究方向。

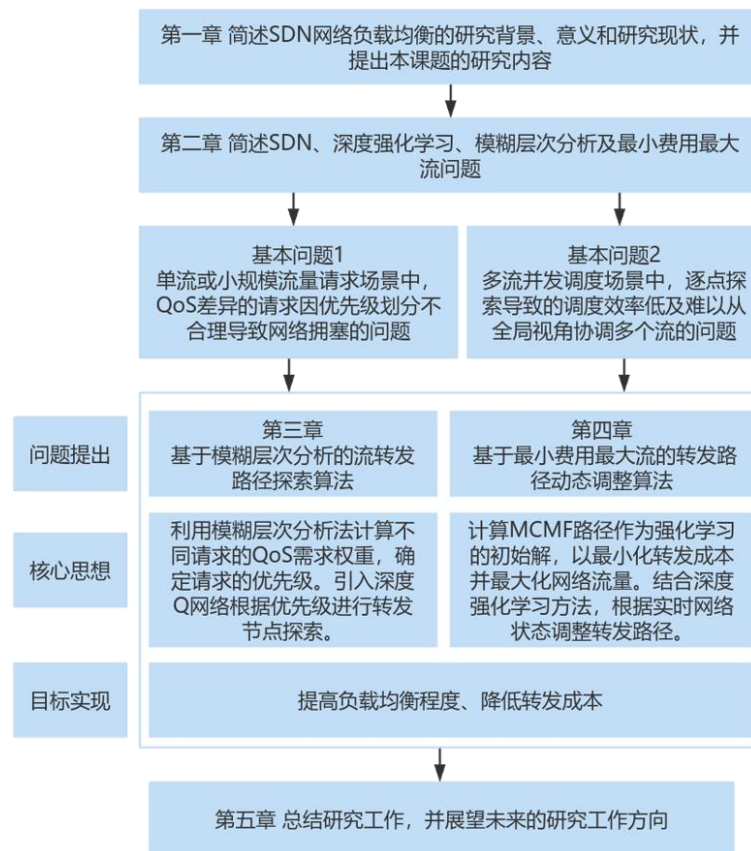


图 1-2 论文的组织结构与章节安排

2 相关理论基础

2.1 软件定义网络

SDN 是一种创新的网络设计理念，其核心特征在于实现了控制平面与数据平面的分离，并引入了集中化控制机制。在传统的网络架构中，控制平面和数据平面通常是紧密耦合的，网络设备既执行数据转发任务，又负责网络的决策与控制。相比之下，SDN 架构通过将控制平面从具体的网络设备中解耦，并将其集中化到一个或多个控制器上，从而使得网络的决策过程更加灵活和集中。具体来说，控制平面负责进行高层次的网络决策，包括路由选择、流量工程、策略制定等，而数据平面的主要任务是根据控制平面下发的指令来执行转发数据包的操作，包括实际的网络硬件组件，如路由器、交换机和中间盒。

SDN 架构的这种设计方式通过控制平面与数据平面的分离，赋予了网络更高的灵活性与可编程性。网络管理员或应用程序能够通过编程接口对网络进行精细的控制和管理，从而能够更加快速、灵活地响应网络流量变化或用户需求的变化。与传统网络架构相比，SDN 在提升网络效率、优化资源配置以及增强网络管理能力方面具有显著优势，尤其在支持动态配置、网络自动化和集中化监控等方面展现出独特的价值。这种架构的引入为网络的智能化和自动化管理奠定了重要基础。

2.1.1 SDN 架构简介

在 SDN 架构中，网络的整体结构通常分为 3 个层次：应用层、控制层和数据层^[38]。这三个层次分别承担不同的功能，从而实现了 SDN 的高度灵活性、可编程性和集中控制，SDN 架构如图 2-1 所示。

1、应用层。应用层位于 SDN 架构的最上层，直接与用户需求和网络服务交互，具有丰富的功能和灵活性，负责根据网络的实际需求和应用场景定义高层次的网络功能和策略。该层包括各种网络应用和服务，例如流量工程^[39-41]、网络虚拟化^[42, 43]、安全防护^[44, 45]、流量监控^[46]等，旨在通过对网络资源的优化配置和调度，提升网络的效率和可用性。应用层中，SDN 应用通过北向协议与控制层交互，向控制层发出需求请求和策略指令，指引控制器如何管理和优化网络资源。控制层应用程序为应用层提供了开放、可编程的网络编程接口，通过编程接口与控制层交互，应用层能够灵活应对网络需求的变化，自动配置和管理网络。高度

的灵活性和可编程性是应用层的主要特点，使得网络可以根据实时需求和当前网络状态进行快速调整和扩展。

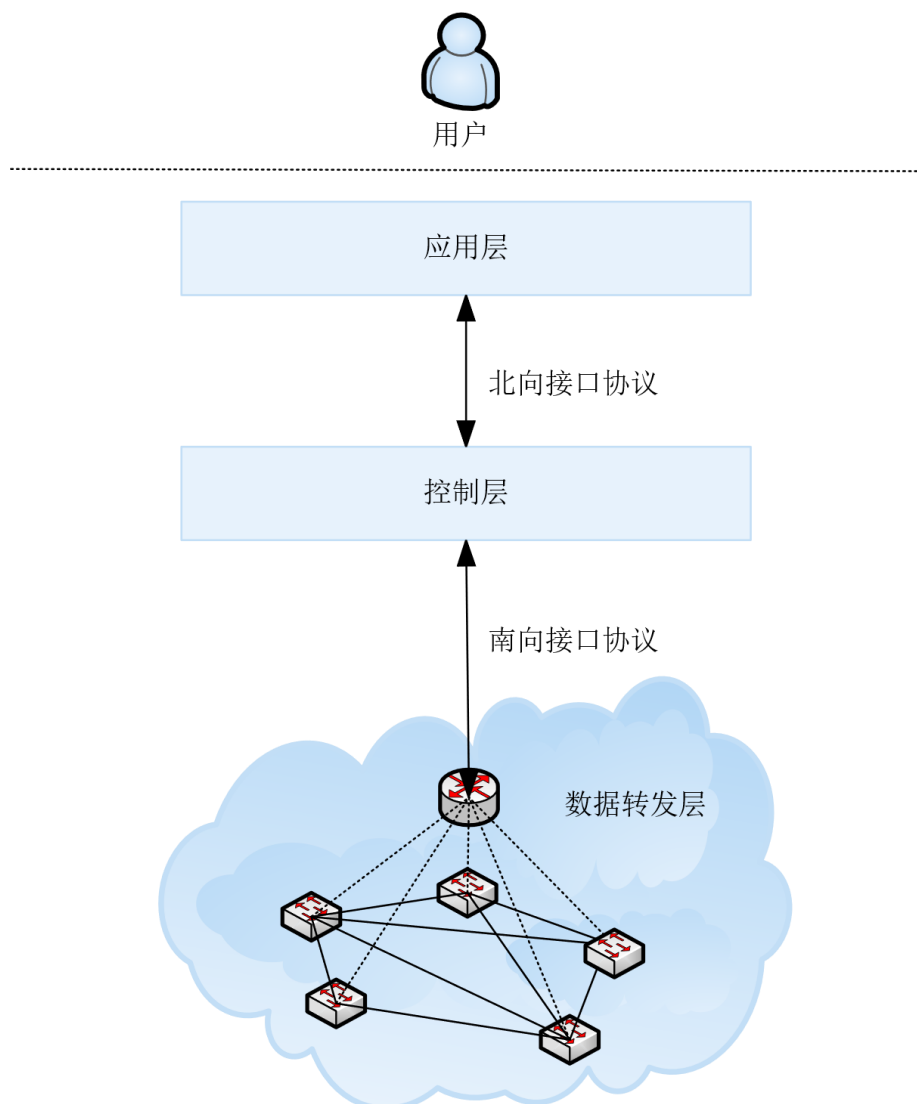


图 2-1 SDN 架构

2、控制层。控制层作为 SDN 架构的核心，负责集中处理网络决策与控制任务。控制层由 SDN 控制器组成，它通过对网络状态的实时感知和分析，制定全局优化策略，并将这些策略下发到数据层的设备。控制层不仅负责流量调度、路径选择、网络拓扑管理，还承担着网络故障处理、流量优先级控制等任务。控制器通过标准化协议（如 OpenFlow）与数据层的交换机、路由器等设备进行通信，动态地调整网络配置，以应对不同的流量需求和网络拓扑变化。由于控制层集中管理网络的决策，网络管理的复杂性得以大幅简化，增强了网络的灵活性、可扩展性与安全性。控制层的集中化控制和可编程性使得网络能够高效应对复杂的应用场景，支持自动化、动态调整和智能化管理。

3、数据层。数据层是 SDN 架构的底层，负责执行控制层下发的网络控制策略和数据转发任务。数据层主要由网络设备组成，如交换机、路由器等。这些设备根据控制层提供的流表规则、路由策略和流量管理指令，执行具体的数据包转发和路由选择任务。数据层的设备通过维护流表，根据数据包的特征（如源 IP 地址、目标 IP 地址、端口号、MAC 地址等）匹配流表规则，决定数据包的转发路径和处理方式。数据层的设备在控制层的指令下执行网络管理任务，如流量监控、负载均衡、带宽优化等。通过与控制层的密切配合，数据层能够灵活响应网络拓扑变化、流量波动等实时动态需求，确保网络的高效运行与灵活适应不同应用场景的需求。

SDN 架构通过应用层、控制层和数据层的分工合作，实现了网络管理的高度灵活性与可编程性。应用层定义了网络的策略和需求，控制层负责集中决策与网络优化，数据层则执行具体的网络转发任务。这样的分层设计使得 SDN 不仅能够简化网络管理、提升资源利用率，还能支持网络的动态调整和智能化控制，满足现代网络环境对可扩展性、灵活性与安全性的需求。

2.1.2 QoS 保障机制

QoS 保障机制是一种用于网络管理和控制的手段，旨在确保网络能够按照预定的服务质量标准（如带宽、延迟等）提供服务，满足不同应用和用户的需求^[47]。QoS 保障机制通过对网络流量进行分类、优先级调度、带宽分配等方法，优化网络资源的使用，确保关键应用能够获得足够的网络资源和性能保证。随着互联网业务的日益丰富，即时通信、多媒体传播等业务的占比不断增加，如何在多业务场景下实现网络资源的优化，已经成为现代网络研究的核心课题。与传统网络相比，现代网络中的多媒体传播业务在互联网应用和社会生活中的地位愈加重要，而这些业务对网络传输质量的要求也日益提高。传统网络时代，以邮件传输等业务为主，网络传输质量的要求相对较低，因此资源优化需求并不迫切。然而，在当前的网络环境中，多媒体业务已成为核心业务之一，其传输质量直接关系到用户体验，因此对网络资源的优化和 QoS 保障机制的发展提出了更高的要求。为了更好地衡量和优化网络服务质量，QoS 保障机制通常基于一系列关键的性能指标。QoS 性能指标是衡量网络服务质量的关键标准，用于评估网络在不同应用场景下的稳定性和可靠性，对于保证网络应用可靠性和用户体验具有重要意义^[48]。主要包括以下几个方面：

- 1、带宽。表示网络能够在单位时间内传输多少数据量，体现了网络传输数据的能力，是网络性能的重要指标，通常以每秒传输的比特数来衡量。带宽越大，单位时间内能传输的数据量越多，因此，带宽直接影响数据传输的速度。

- 2、时延。是指数据从源节点传输到目的节点所需的时间，通常以毫秒为单

位。它直接影响网络的响应速度和用户体验。时延越低，网络的响应速度就越快，用户体验更好，尤其在实时通信（如语音、视频通话）、在线游戏等应用以及要求实时反应的应用（如金融交易、医疗监控）中，低时延至关重要，因为时延的增加会影响系统的实时性和准确性。

3、丢包率。是指数据包在传输过程中丢失的比例。丢包率高会导致数据传输不完整，影响应用性能。在视频通话、语音通信等实时应用中，低丢包率是保证通信质量的关键。

4、可靠性。可靠性是指网络在数据传输过程中，能够保证数据无误差和丢失地到达的程度。是确保数据完整性和正确性的重要指标，尤其在银行、医疗等需要高度准确的数据传输的场景中至关重要。

在 SDN 架构下，QoS 保障机制是确保网络能够满足不同应用需求、优化网络性能的重要手段，可以整合为以下几个关键点：

1、流量分类与优先级调度。流量分类是 QoS 管理的基础。通过对网络流量进行深度包检查或者基于某些流特征（如 IP 地址、端口号、协议类型等）进行分类，SDN 控制器可以将不同流量划分为不同的类别。不同类别的流量可以被标记为不同的优先级，或根据业务需求进行相应的带宽分配。例如，实时视频流、VoIP 流量可能会被标记为高优先级流量，而普通的数据流则可以被标记为低优先级流量。通过流表和流量标记，SDN 控制器能够灵活管理带宽分配和流量调度，确保网络资源的高效使用。QoS 保障机制还可以通过设置服务等级协议（Service Level Agreement, SLA）来定义不同流量的服务质量要求。例如，某些关键业务（如金融交易、医疗应用）可以被分配较高的服务质量级别，而其他低优先级流量（如网页浏览、文件传输）则可以获得较低的服务质量保证。通过 SDN 控制器，管理员可以根据不同的应用需求，灵活地配置不同的优先级和资源分配，确保 SLA 的履行。

2、带宽管理和拥塞控制。通过在数据层（如交换机或路由器）上设置流表，控制器能够决定每个数据流的转发路径，并根据流量需求实时调整带宽分配。例如，实时性要求较高的应用（如视频通话、在线游戏）可以获得更多带宽，而低实时性的应用（如文件下载）则可以获得较少带宽。SDN 控制器能够实时检测网络拥塞情况，并根据当前网络状况调整数据流的路径，为关键流量分配更多带宽，减少拥塞现象。通过动态调整带宽分配和路径选择，QoS 保障机制可有效避免网络拥塞并优化带宽的使用，从而保持网络的稳定性和高效性。

综上所述，QoS 保障机制是网络管理中不可或缺的一部分，通过精确控制网络资源的分配，确保各类应用在不同网络环境下均能获得合适的服务质量保障。无论是在提供差异化服务、提高用户体验，还是在保证网络稳定性、支持商业协议方面，QoS 保障机制都发挥着重要作用。随着网络应用的多样化和对服务质量

的要求不断提高，QoS 保障机制将在未来的网络优化中继续扮演关键角色。

2.2 深度强化学习概述

2.2.1 深度学习

深度学习（Deep Learning, DL）是一类通过多层非线性变换从数据中自动提取特征的机器学习方法。它与传统机器学习方法的最大区别在于，通过多层神经网络的训练，深度学习模型能够自主学习数据的层次化特征表示，尤其在大规模数据和复杂任务中的表现优越^[49]。本质上是通过对大规模数据训练神经网络，让其学习数据的模式、特征和关系，进而用于分类、预测、生成、决策等多种任务。深度学习的基础是模仿人脑神经元的人工神经元，一个神经元通常包括输入、权重、加权求和、激活函数和输出^[50]，如图 2-2 所示是人工神经元的数学模型：

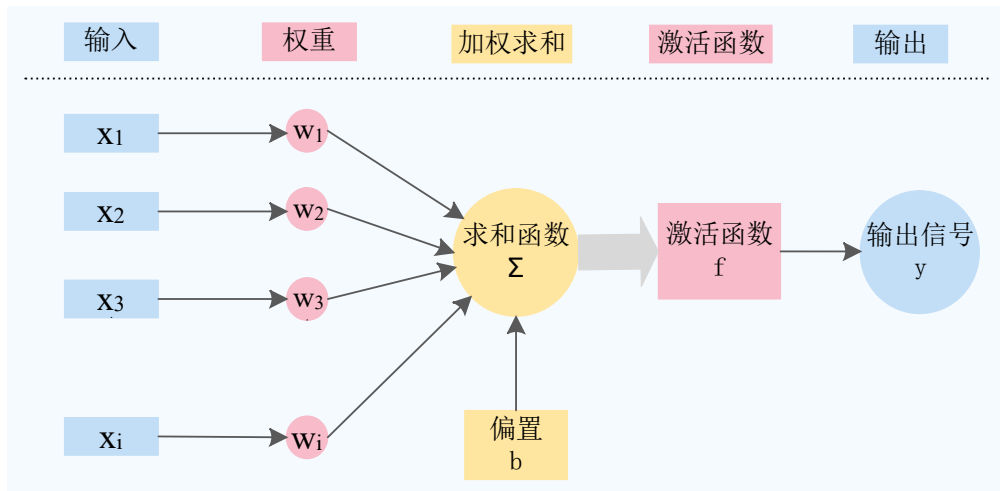


图 2-2 人工神经元的数学模型

图中过程可通过数学公式直观地表示为： $y = f\left(\sum \omega_i x_i + b\right)$ ，即神经元接收多个输入 x_i ，这些输入的数据可以是来自前一层神经元的输出或原始数据（如图像像素、文本特征）。每个输入都被赋予一定的权重 ω_i ，代表该输入的重要性，决定了该输入对最终输出的影响，通过权重对所有输入加权求和，并带上偏置值 b ，偏置项使得神经元在输入全为零时仍然可以激活。随后将加权结果输入非线性激活函数 f ，得到神经元的最终输出，输出结果可以传递给下一层神经元或作为最终预测结果。

由于单一神经元在信息处理能力上存在仅能解决线性可分问题、特征表达能力不足、泛化性较弱以及无法处理时序依赖等问题，单一神经元模型难以胜任复

杂的认知任务。为克服这些局限，研究人员将多个神经元相互连接形成神经网络，提出了多层神经网络模型，通过引入层级化结构与非线性激活函数，使神经网络具备更强的特征提取和模式识别能力。多层神经网络通常由输入层、一个或多个隐藏层、输出层组成，如图 2-3 所示为多层神经网络示意图：

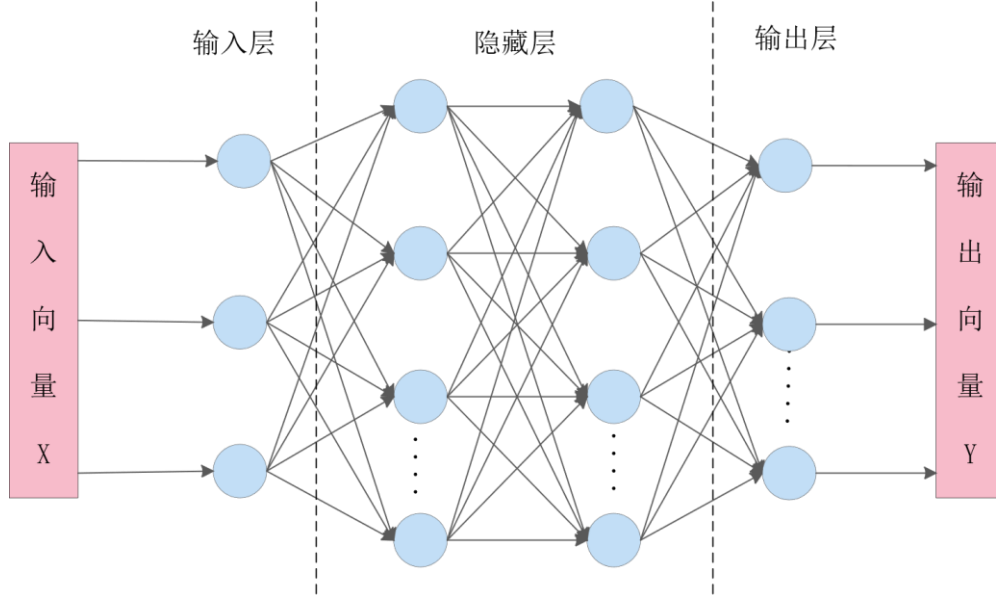


图 2-3 多层神经网络示意图

多层神经网络是深度学习的基础，它通过隐藏层和非线性激活函数突破了单层神经网络的局限，使神经网络具备学习非线性映射和层级特征提取的能力。同时，多层神经网络推动了反向传播、梯度下降等优化方法的发展，为更深层次的神经网络（如 GNN、CNN、RNN、Transformer）奠定了理论基础。

2.2.2 强化学习

强化学习（Reinforcement Learning, RL）也是一种机器学习方法，其核心思想是通过与环境的交互，使智能体在探索与利用之间权衡，学习最优策略以最大化累积奖励，通过奖励信号指导策略优化。由于其强大的决策能力，强化学习在机器人控制、自动驾驶等领域得到了广泛应用。

强化学习包含五个核心要素：智能体、环境、状态、行动和奖励，这些构成了强化学习框架的基本组成部分。强化学习的核心目标是找到最优策略，使得智能体在持续与环境交互的过程中能够获得最大化的长期奖励。为此，智能体需要基于环境反馈不断调整策略，这一过程构成了强化学习的学习机制。这一交互过程通常用马尔可夫决策过程（Markov Decision Process, MDP）进行数学建模。MDP 由一个五元组 $(S, A, P_{ss'}^a, R_{ss'}^a, \gamma)$ 组成，其中 S 表示状态空间，是智能体可能处于的状态集合； A 表示动作空间，是智能体在不同状态下可执行的动作集合； $P_{ss'}^a$ 表示状

态转移概率,指在当前环境状态 s 下执行动作 a 后转移到另一状态 s' 的概率; $R_{ss'}^a$ 表示智能体在状态 s 下采取动作 a 后,状态转移到 s' 时,由环境反馈的当前奖励; γ 为奖励折扣因子,用于权衡当前奖励和未来奖励,决定未来奖励的影响程度。

在强化学习中,策略可以是确定性的,即直接将状态映射到动作 ($a = \pi(s)$);也可以是随机性的,即基于概率选择动作 ($\pi(a|s) = p(a|s)$),在稍微复杂的强化学习问题中通常使用随机策略。而未来奖励通常以回报的形式描述,某一时刻可能得到的未来回报用 G_t 表示,如式(2-1)所示:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots = \sum_{k=1}^{\infty} R_{t+k} \quad (2-1)$$

在无限时间步的情况下,这种无折扣累积奖励的方式可能导致回报无限增长,难以优化。因此,在大多数情况下使用的是折扣累积奖励,引入了奖励折扣因子 $\gamma(0 < \gamma < 1)$,使得未来的奖励按指数递减,如式(2-2)所示:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} \quad (2-2)$$

其中, R_{t+k} 是第 $t+k$ 步的奖励。由于强化学习通常涉及未知环境,未来奖励不能直接计算,而是通过价值函数来估计。

价值函数的一种方法是对每个状态定义一个价值,以衡量智能体在某个状态 s 下,执行策略 π 后所能获得的长期奖励的期望值,定义状态价值函数 $V_{\pi}(s)$ 如式(2-3)所示:

$$V_{\pi}(s) = E_{\pi}[G_t | s_t = s] = E_{\pi}\left[\sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} | s_t = s\right] \quad (2-3)$$

另一种方法是对每个状态下可执行的每个动作都定义一个价值,衡量智能体在状态 s 下采取某个动作 a 后,并遵循策略 π 所能获得的长期奖励的期望值,定义状态-动作价值函数 $Q_{\pi}(s, a)$ 如式(2-4)所示:

$$Q_{\pi}(s, a) = E_{\pi}[G_t | s_t = s, a_t = a] = E_{\pi}\left[\sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} | s_t = s, a_t = a\right] \quad (2-4)$$

根据定义可得 $V_{\pi}(s)$ 和 $Q_{\pi}(s, a)$ 的关系如式(2-5)所示:

$$V_{\pi}(s) = \sum_{a \in A} \pi(a|s) Q_{\pi}(s, a) \quad (2-5)$$

根据式(2-2)式(2-3)可得递推关系式(2-6):

$$\begin{aligned} V_{\pi}(s) &= E_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s_t = s] \\ &= E_{\pi}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \dots) | s_t = s] \\ &= E_{\pi}[R_{t+1} + \gamma G_{t+1} | s_t = s] \xrightarrow{\text{全概率公式}} = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} P_{ss'}^a (R_{ss'}^a + \gamma E_{\pi}(G_{t+1} | s_{t+1} = s')) \end{aligned} \quad (2-6)$$

再由 $V_{\pi}(s)$ 的定义可得式(2-7):

$$V_{\pi}(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} P_{ss'}^a (R_{ss'}^a + \gamma V_{\pi}(s')) \quad (2-7)$$

根据式(2-5)和式(2-7)可得 $Q_{\pi}(s, a)$ 如式(2-8)所示:

$$Q_{\pi}(s, a) = \sum_{s' \in S} P_{ss'}^a (R_{ss'}^a + \gamma V_{\pi}(s')) \quad (2-8)$$

式(2-7)和式(2-8)是描述价值函数递归的贝尔曼方程, 即一个状态或动作的价值等于该状态或动作的当前奖励加上未来可能的状态或动作值的折扣价值期望。

强化学习的最优策略 π^* , 表示对任一状态, 策略 π^* 的价值函数值大等于其他任何策略 π 下的价值函数值, 即最优策略对应最大状态价值函数 $V^*(s)$, 如式(2-9), 以及最大状态-动作价值函数 $Q^*(s, a)$, 如式(2-10), $V^*(s)$ 与 $Q^*(s, a)$ 的关系如式(2-11)所示:

$$V^*(s) = \max_{\pi} V_{\pi}(s), \forall s \in S \quad (2-9)$$

$$Q^*(s, a) = \max_{\pi} Q_{\pi}(s, a), \forall s \in S, a \in A \quad (2-10)$$

$$V^*(s) = \max_{a \in A} Q^*(s, a) \quad (2-11)$$

将式(2-11)带入式(2-8), 即得到最优贝尔曼方程, 如式(2-12)所示:

$$Q^*(s, a) = \sum_{s' \in S} P_{ss'}^a (R_{ss'}^a + \gamma \max_{a' \in A} Q^*(s', a')), \quad (2-12)$$

最优贝尔曼方程是强化学习理论的核心基础, 它通过递归关系刻画最优状态价值函数 $V^*(s)$ 和最优动作价值函数 $Q^*(s, a)$, 从而提供了一种计算最优累积回报的数学框架, 通常先求出最优状态动作-价值函数, 再根据式(2-13)选择最优策略。

$$\pi^*(a|s) = \begin{cases} 1, & \text{如果 } a = \arg \max_{a' \in A} Q^*(s, a') \\ 0, & \text{否则} \end{cases} \quad (2-13)$$

该方程不仅为动态规划、值迭代、策略迭代等传统强化学习方法提供理论支撑, 也构成了 Q-learning 及深度强化学习 (如 DQN、Actor-Critic 等) 的基础。通过最优贝尔曼方程, 智能体能够在未知环境中学习最优策略, 实现对长期回报的优化, 因此在决策优化、机器人控制、游戏智能体等领域具有广泛应用价值。

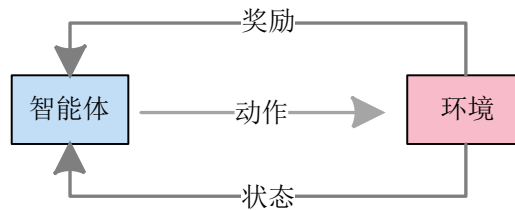


图 2-4 智能体学习过程示意图

总结强化学习即是通过智能体与环境的交互来学习最优决策的过程。智能

体依据策略从当前状态选择动作，环境执行该动作后反馈新的状态和奖励，智能体通过优化策略来最大化长期累积奖励，从而在未知环境中自主学习最优行为，学习过程示意图如图 2-4 所示。

2.2.3 深度强化学习

深度强化学习可依据算法类型分为基于值函数、基于策略梯度及值函数与策略结合的方法；从学习方式看，分为无模型方法和基于模型方法；按任务类型可分为离散/连续动作、单/多智能体及部分可观测环境的解决方案；模型结构上常采用 CNN 处理图像、RNN 处理时序、注意力机制或分层策略应对复杂任务，表 2-1 总结了 DRL 方法的大致分类。

表 2-1 DRL 分类

分类	主要方法	适用场景
基于值函数	DQN, DDQN, Rainbow DQN, Dueling DQN	离散动作空间
基于策略	PPO, TRPO	连续控制
值函数和策略结合	A2C, A3C, DDPG, SAC	复杂环境的决策优化
基于模型	MPC, MBPO	数据获取成本高的任务

DRL 结合了深度学习与强化学习两种机器学习技术，利用神经网络逼近值函数或策略，解决高维决策问题。传统强化学习在高维状态空间下难以表示复杂策略，而深度学习强大的特征提取能力能有效应对这一挑战。结合深度学习后，强化学习可以更好地处理连续状态和复杂环境，提高泛化能力；同时，强化学习赋予深度学习自主决策能力，使其在交互式任务中具备自适应性。深度学习和强化学习结合的示意图如图 2-5 所示^[51]。

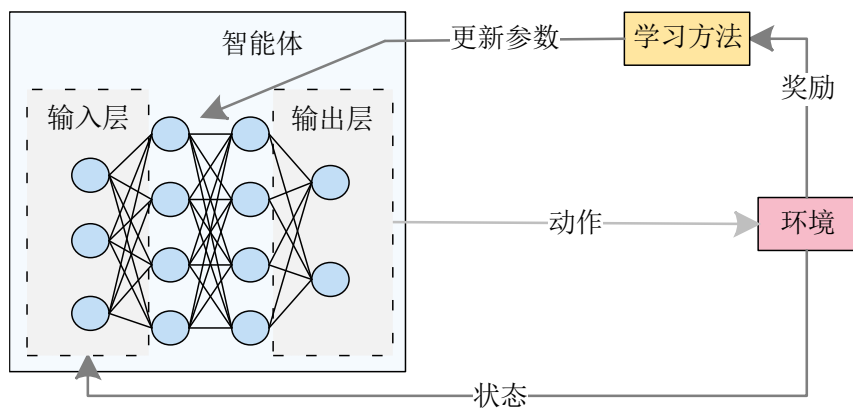


图 2-5 RL 和 DL 结合示意图

相比单独使用某一方法，DRL 兼具值函数方法的稳定性和策略方法的灵活性，能够更高效地学习复杂策略。应用场景涵盖游戏 AI、机器人控制及自动驾驶等，同时可结合模仿学习、元学习或多任务学习扩展能力。其核心在于针对问题特性（如动作空间、数据效率、环境复杂度）选择适配算法，平衡探索与利用以实现高效决策。

2.3 模糊层次分析

模糊层次分析法（Fuzzy Analytic Hierarchy Process, FAHP）是在层次分析法（Analytic Hierarchy Process, AHP）的基础上结合模糊数学发展而来的一种多准则决策方法。AHP 通过构建层次结构模型，如图 2-6 所示，同时对不同准则的重要性进行比较，并采用一致性检验来确保判断的合理性。然而，传统 AHP 在实际应用中存在一定局限性，如比较重要性时的主观性。

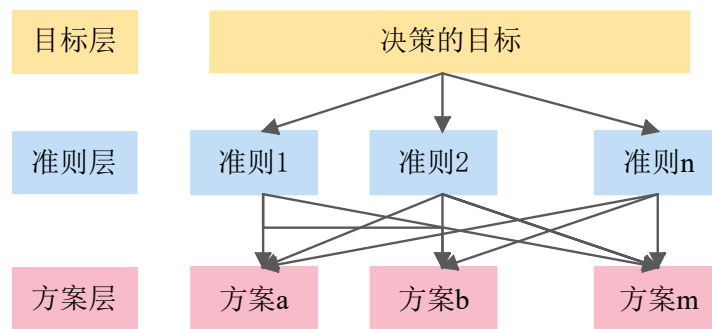


图 2-6 AHP 层次结构图

FAHP 通过引入模糊数来量化决策者的判断，并采用模糊合成法、模糊加权平均法等数学工具，降低主观判断的不确定性，提高决策过程的可靠性与稳定性。FAHP 在风险评估、智能决策等领域得到了广泛应用，尤其适用于信息不完全或不确定的复杂决策环境。FAHP 通过模糊一致性矩阵计算权重，并结合模糊排序方法，使得决策结果更符合认知和实际应用需求，其核心优势在于能够处理模糊性信息，提高决策科学性和合理性，在不确定性环境下为多准则决策提供更优化的解决方案。

2.4 最小费用最大流问题

最小费用最大流（Minimum Cost Maximum Flow, MCMF）是网络流理论中的一类组合优化问题，旨在为给定容量的流网络寻找从源节点到汇节点的最大可行

流，且该流的传输总费用达到最小。该问题结合了最大流问题与最短路问题的双重目标，是运筹学、计算机科学及工业工程领域的经典课题，广泛应用于物流规划、通信网络设计、资源分配等场景。要介绍 MCMF 首先要了解网络最大流问题。

2.4.1 网络最大流问题基本概念

1、容量网络。在一个有向图 $D=(V,A)$ 在 V 中指定了一点称为发点(记为 v_s)，另一点称为收点(记为 v_t)，其余的点叫中间点。对于每一个弧 $(v_i, v_j) \in A$ ，对应有一个 $c(v_i, v_j) \geq 0$ ，简写为 c_{ij} ，称为弧的容量。通常把这样的 D 叫作一个网络，记作 $D=(V,A,C)^{[52]}$ ，如图 2-7 所示：

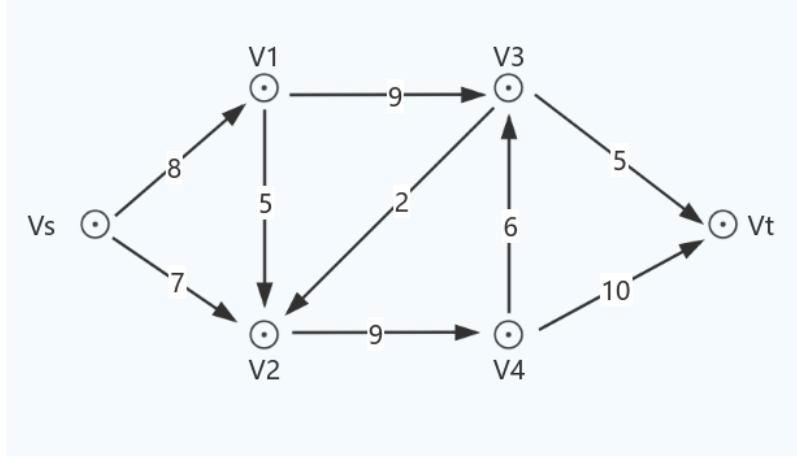


图 2-7 容量网络

2、流与可行流。流指加在网路中各条弧上的实际流量 $f = \{f(v_i, v_j)\}$ ，称 $f(v_i, v_j)$ 为弧 (v_i, v_j) 上的流量，简记为 f_{ij} 。将满足下列条件的流 f 称为可行流：

(1) 容量限制条件：对每一弧 $(v_i, v_j) \in A$

$$0 \leq f_{ij} \leq c_{ij} \quad (2-14)$$

(2) 平衡条件：

对于中间点：流出量等于流入量，即对每个 $i(i \neq s, t)$ 有

$$\sum_{(v_i, v_j) \in A} f_{ij} - \sum_{(v_j, v_i) \in A} f_{ji} = 0 \quad (2-15)$$

对于发点 v_s ，有

$$\sum_{(v_s, v_j) \in A} f_{sj} - \sum_{(v_j, v_s) \in A} f_{js} = v(f) \quad (2-16)$$

对于收点 v_t ，有

$$\sum_{(v_t, v_j) \in A} f_{tj} - \sum_{(v_j, v_t) \in A} f_{jt} = -v(f) \quad (2-17)$$

3、网络的最大流问题。是指求一个流 $\{f_{ij}\}$ 使其流量 $v(f)$ 达到最大，并且满足式(2-14)至(2-17)。

4、截集与截量。截集是指容量网络中的发点和收点分割开，并使 $v_s \rightarrow v_t$ 的流中断的一组弧的集合。截容量是组成截集合中的各条弧的容量之和，用 $c(V, \bar{V})$ 表示， $c(V, \bar{V}) = \sum_{(v_i, v_j) \in (V, \bar{V})} c(v_i, v_j)$ ，如图 2-8 所示， AA' 将网络上的点分隔成 $V, \bar{V}$ 两个集合，弧集合 $\{(v_1, v_3), (v_2, v_4)\}$ 就是一个截集，截流为 18。

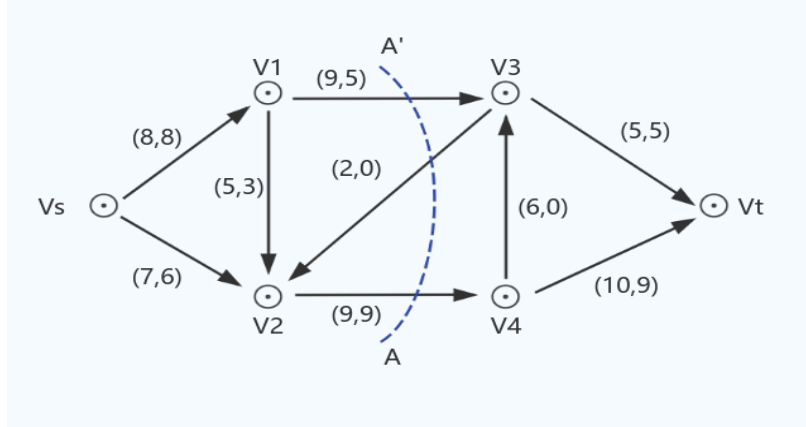


图 2-8 截集示意图

5、增广链。在网络的发点和收点之间能找到一条链，在该链上所有指向为 $v_s \rightarrow v_t$ 的弧，称为前向弧，记作 μ^+ ，存在 $0 \leq f_{ij} < c_{ij}$ ；所有指向为 $v_t \rightarrow v_s$ 的弧，称为后向弧，记做 μ^- ，存在 $0 < f_{ij} \leq c_{ij}$ ，则称这样的链为增广链。如图 2-9 中 $v_s \rightarrow v_2 \rightarrow v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_t$ 即为增广链，沿此路径可以增加流量，从而提升总流量或优化费用。

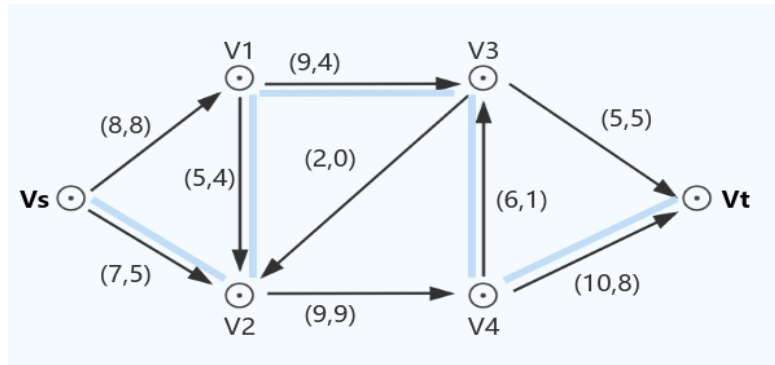


图 2-9 增广链示意图

6、相关定理。

- (1) 可行流 f^* 是最大流，当且仅当不存在关于 f^* 的增广链。
- (2) 最大流量最小截量定理：任一个网络 D 中，从 v_s 到 v_t 的最大流的流量等

于分离 $v_s \rightarrow v_t$ 的最小截集的容量。

7、求解算法思想。由相关定理可得求解最大流的方法即由一个流开始，系统地搜寻增广链，然后在此链上增流，继续这个增流过程，直至不存在增广链。

2.4.2 最小费用最大流问题的描述与解决方法

1、问题描述。在最大流问题的基础上，对于每一弧 $(v_i, v_j) \in A$ ，增加一个单位流量的费用 $b(v_i, v_j) \geq 0$ ，简记为 b_{ij} ，最小费用最大流问题就是要求一个最大流 f ，使流的总传输费用 $b(f) = \sum_{(v_i, v_j) \in A} b_{ij} f_{ij}$ 取极小值。

2、解决方法。若 f 是流量为 $v(f)$ 的所有可行流中费用最小者（初始可行流费用最小，为 0），而 μ 是关于 f 的所有增广链中费用最小的增广链，那么沿 μ 去调整 f ，得到的可行流 f' ，就是流量为 $v(f')$ 的所有可行流中的最小费用流。这样，当 f' 是最大流时，它也就是所要求的最小费用最大流了。最小费用最大流算法通过迭代寻找费用最小的增广链来逐步调整流量，直到无法继续增广为止。

2.5 本章小结

本章首先梳理了 SDN 和 QoS 保障机制这两个 SDN 网络负载均衡中的核心内容，阐述 SDN 的架构优势及其动态资源管控能力，并结合 QoS 说明网络性能保障的关键机制，为流量调度奠定架构基础。接着，从深度学习与强化学习的融合机制出发，解析 DRL 在动态网络环境中的自适应优化潜力。介绍了深度强化学习网络流量优化与智能决策的核心理论基础，为后续研究提供多维度支撑。其次，结合模糊层次分析法，提出复杂网络场景下多目标决策的量化评估框架，提高决策合理性。最后，聚焦网络流量优化问题，详细剖析网络最大流与最小费用最大流的核心概念，通过增广链理论与残留网络构建方法，揭示流量分配与成本控制的数学建模逻辑，为后续工作提供支撑。

3 基于模糊层次分析的流转发路径探索算法

3.1 引言

由第一章的绪论和第二章的相关理论基础介绍可知，在多媒体应用中，用户请求存在多样性，不同请求的服务质量要求各不相同，如带宽、时延等要求可能存在较大差异。此外，在实际网络环境中，请求的源节点和目的节点之间通常存在多条可选路径，而不同时间段内，各路径的带宽和时延等网络资源状况可能动态变化，因此，如何进行合理的流量分配和智能的路由选择成为关键挑战。与此同时，网络中也存在着实时流量因优先级划分不合理而导致 congestion 的问题，特别是在多流共存的场景中，不同类型的业务流量可能相互竞争有限的网络资源。

近年来，深度强化学习在解决复杂的优化问题方面显示出巨大的潜力。以 Q 学习为代表的强化学习是人工智能的一个重要分支。它通过与环境的不断“试错”交互来学习最佳策略。然而，在复杂的网络环境中，由于网络状态的动态变化和高维特性，传统 Q 学习方法所需存储的 Q 表规模可能过大，难以实际应用。为此，深度 Q 学习采用神经网络来近似 Q 值函数，从而有效缓解了 Q 表存储空间过大的问题，使其能够在高维、复杂的网络环境中进行高效学习和决策。此外，在多目标优化问题中，传统方法往往难以直接对不同 QoS 标准进行量化比较，而 FAHP 能够在多标准决策问题中发挥优势，利用模糊数对不同 QoS 需求的重要性进行灵活评估，从而计算出请求的优先级，该方法提高了计算请求优先级的灵活性。

基于上述分析，本章提出了基于模糊层次分析的流转发路径探索算法，综合考虑请求的多个 QoS 需求，以确保不同类型的流量都能得到合理的资源分配。同时可以根据实时网络状态和历史经验不断优化路径策略，结合奖励设置，为请求探索出转发成本低、链路负载均衡程度高的路径，以实现低转发成本和高链路复杂均衡的目标。

3.2 问题描述与建模

本节中，首先对本研究所关注的和核心问题展开探讨，对研究问题的背景以及所面临的主要挑战进行系统性描述；随后将详细阐述本章所使用的系统框架；最后，在系统框架的基础上，对研究问题进行建模，明确模型中的关键参数、变量以及约束条件，为接下来的算法设计提供理论支撑。

3.2.1 问题描述

在基于 SDN 的多媒体流服务应用中，网络面临着动态流量调度和资源分配的挑战。由于多媒体应用涉及视频、音频、在线游戏、文件传输等多种类型的业务流。其服务质量需求各不相同，如带宽、时延等指标存在显著差异。在流量转发过程中，由于部分请求的优先级较低，可能导致其在竞争网络资源时处于劣势，进而引发关键链路的过载，而部分链路的利用率较低，导致网络资源利用不均衡。与此同时，优化链路负载均衡与控制链路转发成本之间存在明显矛盾：若倾向于提升负载均衡程度，可能会导致数据包绕行，从而增加传输时延和转发成本；若倾向于降低转发成本，往往会选择最短路径进行数据转发，可能加剧链路拥塞，影响整体网络性能和 QoS 保障。因此，如何在控制转发成本的同时有效提高链路均衡，成为亟待解决的问题。

3.2.2 系统框架

本文针对 SDN 控制层和数据层的同步优化研究 SDN 网络负载均衡策略。本章提出了流转发路径探索算法，图 3-1 是本章所提算法的 SDN 路由框架^[53]。

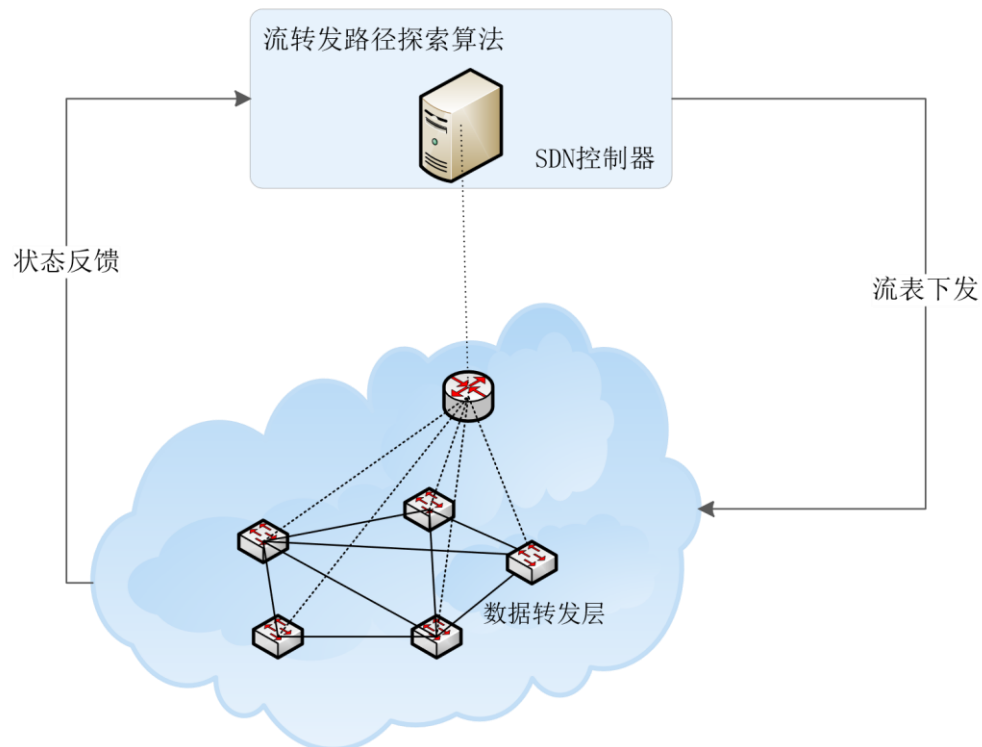


图 3-1 SDN 路由框架

本文进一步将控制器内部划分为链路感知模块、深度强化学习模块、流表下发模块^[54]，如图 3-2 所示。具有不同 QoS 要求的请求到达交换机端口后，首先

经过 FAHP 模块处理, 得到请求优先级。链路感知模块能够获取拓扑和链路信息; 深度强化学习模块则根据链路感知模块得到的拓扑和链路信息为每条请求探索最佳转发路径; 流表下发模块与路由信息组成流表并将流表下发给相应交换机。交换机转发端口根据深度强化学习模块输出的转发路径和链路感知模块得到的拓扑来确定。

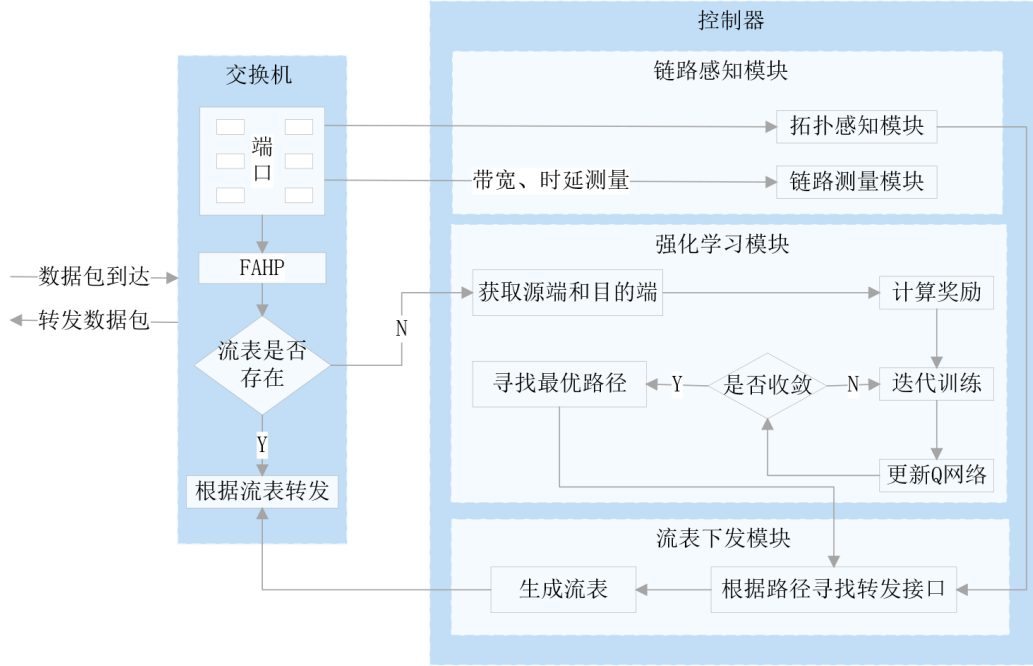


图 3-2 系统处理模型框架

3.2.3 问题建模

1、底层物理网络模型

本文使用无向图 $G(V, E)$ 表示图 3-1 中的 SDN 物理网络, 其中 V 表示物理节点的集合, E 表示物理链路的集合, $n=|E|$ 表示链路数量。 (i, j) 表示节点 i 、 j 间的链路, B_{ij} 、 D_{ij} 、 C_{ij} 分别表示链路 (i, j) 上的带宽、时延、成本。请求流集合由 F 表示, 请求 $k(k \in F)$ 由一个五元组表示 $(S_k, T_k, b_k, d_k, t_k)$, S_k 表示请求 k 的源节点, T_k 表示目的节点, b_k 表示请求 k 的带宽要求, d_k 表示请求 k 允许的最大时延, t_k 表示请求 k 的类型。二元变量 f_{ij}^k 表示请求 k 是否经过链路 (i, j) , $f_{ij}^k=1$ 表示请求 k 经过链路 (i, j) , $f_{ij}^k=0$ 表示请求 k 不经过链路 (i, j) 。 R 表示所有请求的最终转发路径集合, r_k 表示请求 k 的最终转发路径。

2、相关约束条件

网络中链路的使用带宽不能超过链路的最大带宽, 因此需要满足约束条件 (3-1):

$$\sum_{k \in F} \sum_{r_k \in R} \sum_{(i,j) \in r_k} f_{ij}^k \times b_k \leq B_{ij}, \forall i, j \in V \quad (3-1)$$

同时在使用路径 r_k 转发请求时, 请求 k 的带宽要求和时延要求要得到满足, 因此要满足约束条件(3-2)和(3-3):

$$\sum_{(i,j) \in r_k} D_{ij} \leq d_k, \forall i, j \in V \quad (3-2)$$

$$\forall_{(i,j) \in r_k} B_{ij} \geq b_k, \forall i, j \in V \quad (3-3)$$

3、优化目标

在保证 QoS 需求的前提下, 使网络中链路负载均衡。使用 $\frac{\sum_{k \in F} f_{ij}^k \times b_k}{B_{ij}} \times 100\%$ 表示链路 (i, j) 的带宽利用率 BR_{ij} , 本文将为所有请求探索完路径后的网络各链路的带宽利用率的标准差定义为负载均衡系数 k , 如公式(3-4)所示, 用以衡量整个网络的负载均衡程度。

$$k = \sqrt{\frac{\sum_{i \in V} \sum_{j \in V} (BR_{ij} - \overline{BR})^2}{n - 1}} \quad (3-4)$$

但单一追求网络流量的负载均衡会导致转发成本过高, 所以本文另一方面的优化目标在于降低转发成本, 转发成本用公式(3-5)表示。

$$C_{forward} = \sum_{k \in F} \sum_{(i,j) \in R} f_{ij}^k \times C_{ij} \quad (3-5)$$

因此优化目标包括负载均衡度和转发成本两个部分, 于是建立如下网络模型优化问题:

$$\text{Minimize} \quad \alpha k + (1 - \alpha) C_{forward} \quad (3-6)$$

$$\text{s.t.} \quad (3-1)-(3-3)$$

$$f_{ij}^k \in \{0, 1\}$$

$$\alpha \in [0, 1]$$

α 是网络负载均衡系数和转发成本的权重调节因子。本文涉及的主要参数和意义如表 3-1 所示。

表 3-1 主要参数和意义

类型	参数
$G(V, E)$	底层物理网络
V	物理节点的集合
E	物理链路的集合
(i, j)	节点 i 、 j 之间的链路
B_{ij}	链路 (i, j) 上的带宽

表 3-1 主要参数和意义（续）

类型	参数
D_{ij}	链路(i,j)上的时延
C_{ij}	链路(i,j)上的路由成本
F	请求流的集合
S_k	请求 k 的源节点
T_k	请求 k 的目的节点
b_k	请求 k 的带宽要求
d_k	请求 k 允许的最大时延
t_k	请求 k 的类型
f_{ij}^k	二元变量, 请求 k 经过链路(i,j), 则为 1
R	所有请求的最终转发路径集合
r_k	请求 k 的最终转发路径
BR_{ij}	链路(i,j)的带宽利用率
k	网络的负载均衡系数

上述优化目标受到请求 QoS 需求和链路负载约束。为了求解以上优化问题, 本文提出了基于模糊层次分析法的流转发路径探索算法, 下一节将详细介绍该算法的设计原理。

3.3 流转发路径探索算法

本章所提算法主流程图如图 3-3 所示。该图概括了本章所提出的流转发路径探索算法的整体流程。本章算法基于深度强化学习框架, 通过引入深度 Q 网络实现路径选择的智能优化。在接收到带有优先级的转发请求后, 系统根据当前的网络环境和已有策略, 动态调整路径选择, 从而适应复杂多变的网络状况。

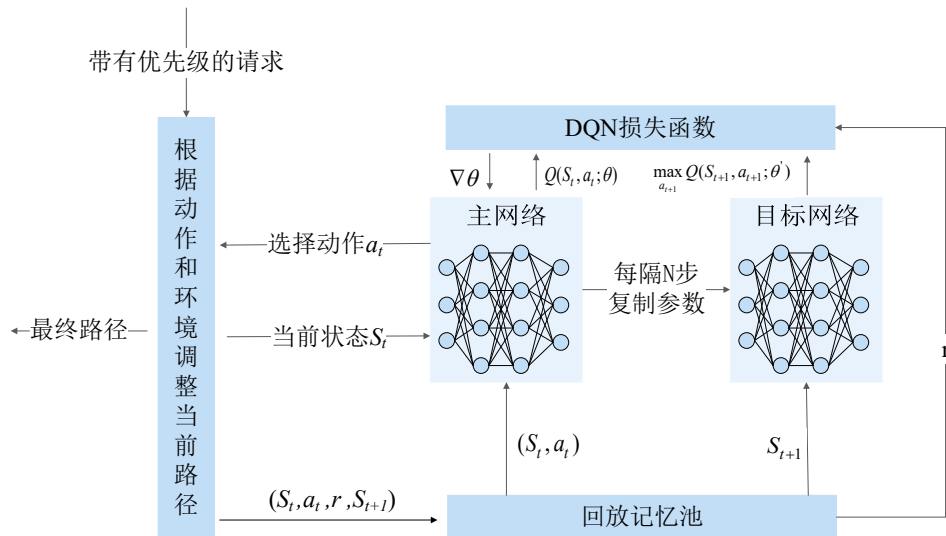


图 3-3 流转发路径探索算法流程图

3.3.1 算法描述

算法首先采用 FAHP 对网络中各请求的 QoS 进行综合考虑，赋予请求相应的优先级，得到带有优先级的请求。计算优先级的步骤如下：1、建立递阶层次结构；2、构造模糊一致判断矩阵；3、由判断矩阵计算该指标的相对权重；4、根据所有的指标权重矩阵计算优先级。

1、建立递阶层次结构。如图 3-4 所示为本章算法应用 FAHP 所构建的的层次结构^[55]，本章算法中目标层是所有请求的综合优先级，准则层本章算法考虑的是请求的带宽、时延和流类型，方案层即是接收到的所有初始请求,请求中包含了请求类型，由此可以对不同流类型或多条同一流类型的请求计算优先级。FAHP 能够根据不同类型流量的实际业务需求，灵活调整各因素的权重，实现对多类型请求场景下优先级的精细化区分，从而为后续的路径调度与资源分配提供更具针对性的决策依据。

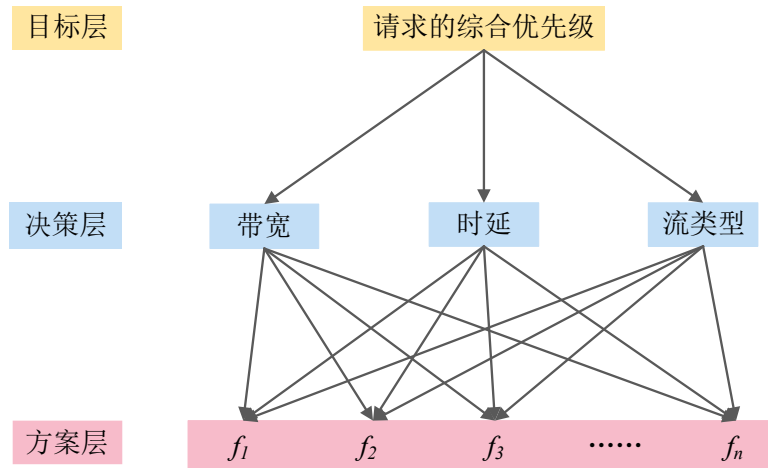


图 3-4 请求的 FAHP 层次结构

2、构造模糊一致判断矩阵。若上一层次的元素 C 同下一层次中的元素 $a_1, a_2, \dots, a_n$ 有关联，则构造模糊一致判断矩阵 R ，该矩阵通过引入模糊数对各决策因素之间的相对重要性进行描述，以量化各方案相对于某一决策因素的重要程度。

C	a_1	a_2	$\dots$	a_n
a_1	r_{11}	r_{12}	$\dots$	r_{1n}
a_2	r_{21}	r_{22}	$\dots$	r_{2n}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
a_n	r_{n1}	r_{n2}	$\dots$	r_{nn}

其中， r_{ij} 表示元素 a_i 和 a_j 相比较的重要程度，具有如下性质：(1) $0 \leq r_{ij} \leq 1$ ；(2) $r_{ii}=0.5$ ；(3) $r_{ij}+r_{ji}=1$ 。标度及其含义如表 3-2 所示^[56]。

表 3-2 标度及含义

标度 r_{ij}	含义
0.5	两元素相比较, 同等重要
0.6	两元素相比较, a_i 比 a_j 稍微重要
0.7	两元素相比较, a_i 比 a_j 明显重要
0.8	两元素相比较, a_i 比 a_j 强烈重要
0.9	两元素相比较, a_i 比 a_j 极端重要
0.1,0.2,0.3,0.4(互补)	若元素 a_i 与 a_j 相比较得到 r_{ij} , 则元素 a_i 与元素 a_j 相比较得到的为 $r_{ji}=1-r_{ij}$

3、由判断矩阵计算该指标的权重。具体计算方式为：首先，对模糊一致判断矩阵中某一指标对应行的所有元素进行求和，以反映该指标相对于其他指标的重要性之和；随后，在该求和结果的基础上加入一个与矩阵阶数相关的调整项；最后，将上述结果除以归一化系数，从而实现权重的标准化处理。通过该计算方法，可以有效获得每一评价指标在决策体系中的相对权重，为后续的综合评价与优先级排序提供量化依据。

4、根据所有的指标权重矩阵计算优先级。

计算优先级的过程如算法 1 所示。首先构建并利用目标层到决策层的模糊一致判断矩阵计算决策层的三个因素在计算综合优先级时的权重 $\omega_{\text{决策因素}}=[\omega_{\text{带宽}}, \omega_{\text{时延}}, \omega_{\text{请求类型}}]^T$ （第 1-2 行），其次构建决策层到方案层的模糊一致判断矩阵并计算方案层的各个请求针对决策层中的三个因素的权重，如当前请求集合 $F=\{f_1, f_2, f_3, f_4\}$ ，得到 $\omega_{\text{带宽}}=[\omega_{f1}, \omega_{f2}, \omega_{f3}, \omega_{f4}]^T$ ，同理得到 $\omega_{\text{时延}}, \omega_{\text{请求类型}}$ （第 3 至 24 行），最后列出权重矩阵计算综合权重达到请求的综合优先级。

算法 1 模糊层次分析 (FAHP)

输入： 初始请求集合 F

输出： 排序后的请求集合 *sorted_requests*

1. 设定目标层到决策层的模糊一致矩阵
2. 根据模糊一致矩阵计算带宽、时延、请求类型三个指标的权重 *index_weights*
3. 设定决策层到方案层的模糊一致矩阵
4. 根据请求的带宽计算每个请求在带宽中的模糊一致矩阵 *bandwidth_fuzzy_matrix*
5. for $i = 1$ to |请求数量|
6. for $j = 1$ to |请求数量|
7. 两两比较请求带宽需求大小得到带宽模糊矩阵
8. end for
9. end for
10. 根据 *bandwidth_fuzzy_matrix* 计算每个请求在带宽中的权重
11. 根据随机请求的时延计算每个请求在时延中的模糊一致矩阵 *delay_fuzzy_matrix*
12. for $i = 1$ to |请求数量|
13. for $j = 1$ to |请求数量|

```

14.         两两比较请求时延需求大小得到时延模糊矩阵
15.     end for
16. end for
17. 根据 delay_fuzzy_matrix 计算每个请求在时延中的权重 delay_weights
18. 设置不同请求类型之间的重要性字典 important_map
19. for   i = 1 to |请求数量|
20.     for   j = 1 to |请求数量|
21.         根据重要性字典两两比较请求的类型得到请求的模糊一致矩阵
                type_fuzzy_matrix
22.     end for
23. end for
24. 根据 type_fuzzy_matrix 计算每个请求在请求类型中的权重 type_weights
25. 由 index_weights, bandwidth_weights, delay_weights, type_weights 得到综合权重, 从而根据
    综合权重值对请求排序
26. return sorted_requests

```

接着流转发路径探索算法需要学习当前网络状态, 根据优化目标(3-4)和(3-5)探索下一步转发节点, 由此引入深度强化学习算法。在深度强化学习中, MDP 是常用于描述在动态环境中解决顺序决策问题的数学框架。MDP 通常使用动态规划或深度强化学习来解决^[57]。DRL 方法的无模型性质使其更适合具有大状态空间的问题, 可以通过收集经验更新 Q 值来学习最优策略^[58]。传统的强化学习方法通过 Q 表存储和更新 Q 值, 使得它们在处理状态空间大的问题时面临挑战。使用深度神经网络来近似价值 Q 函数的 DQN, 很好解决了上述问题, 且 DQN 适合解决动作空间为离散的问题, 本文的目标是探索路径, 动作空间是离散的, 因此本文将问题求解过程被建模为 MDP 并使用 DQN 进行求解。

MDP 模型将决策问题分解为状态空间、动作空间、转移概率、奖励和折扣因子, 来系统地描述智能体如何与环境交互, 并通过优化策略来实现目标。以下是对本文问题状态空间、动作空间和奖励的建模。由于在本文问题中, 网络链路状态随机变化, 因此状态转移概率是无法确定的。

状态空间。本文中状态空间包含两类信息。可定义为 $S(t) = \{n(t), b(t)\}$, 其中 $n(t) = [n_i]$, $i \in [0, |V|-1]$ 表示 t 时刻所在的节点 i , $b(t) = [b_{00}(t), b_{01}(t), \dots, b_{ij}(t)]$, 其中, $i, j \in [0, |V|-1]$ 表示 t 时刻网络的带宽矩阵, $b_{ij}(t)$ 表示 t 时刻链路 (i, j) 上的带宽大小, 带宽矩阵大小为 $|V| \times |V|$, 因此状态空间的维度为 $|V| \times |V| + 1$ 。这些信息更全面描述了当前状态, 对于路径探索十分重要。

动作空间。本文使用 DQN 为请求寻找下一转发节点。因此动作空间是基于 t 时刻当前状态中的当前节点 n_i 可选择的所有下一个节点, 定义为 $A(t) = V \setminus n(t)$ 。

奖励计算。在执行完选择的动作 a_t 后, 环境会反馈一个奖励 r 。本文中奖励考虑的指标是时延大小 *delay*, 负载均衡系数 k 和转发成本 *cost* 的结合, 若时延超出请求的时延要求, 则得到负奖励, 如公式(3-7)所示。

$$r(a, s) = \begin{cases} -1000, & \text{不满足请求时延} \\ \frac{1}{k} + \frac{1}{cost} + \frac{1}{delay}, & k, cost, delay \neq 0 \end{cases} \quad (3-7)$$

算法 2 描述了流转发路径探索算法的具体过程。首先,调用算法 1(第 1 行)得到带有优先级的请求集合;接着初始化拓扑环境和经验池(第 2-3 行),以生成状态空间和存储经验,同时初始化主网络和目标网络的参数(第 4-5 行);然后进行 `num_episode` 次数的训练,每一次训练将针对所有请求完成一次探索路径(第 8-20 行),探索过程中调用算法 3 进行路径调整。在训练过程中执行上述算法描述的步骤,最终得到所有请求的转发路径策略以及该策略对应的转发成本和负载均衡系数。

算法 2 流转发路径探索算法

输入: 训练次数 `num_episode`, 更新步长 `step`

输出: 请求的最优路径 `path`, 转发成本 `cost`, 负载均衡系数 `k`

- 1.调用算法 1, 得到带有优先级的请求集合 `sorted_requests`
 - 2.初始化拓扑环境
 - 3.初始化经验池 `ReplayBuffer`
 - 4.初始化主网络参数 θ
 - 5.初始化目标网络参数 $\theta' = \theta$
 - 6.for episode= 1 to `num_episode`
 7. 初始化状态空间
 8. 对所有请求获取请求的源节点 `start`、目的节点 `end`、时延要求 `delay_require`
 9. 获取初始状态 S_a , 路径 `path=[start]`
 10. while `path[-1]≠end`
 11. 在主网络中采用 ϵ -Greedy 策略选择动作 a_t
 12. 调用算法 3 调整路径
 13. 得到的经验(S_t, a_t, r_t, S_{t+1})存储到经验池中
 14. 从经验池中随机采样一批经验, 更新主网络参数
 15. 若达到更新步长, 则将主网络参数 θ 复制到目标网络 θ'
 16. 状态设为下一状态
 17. end while
 18. 输出 `path`
 19. end for
 20. return 所有请求转发路径 `path, cost, k`
-

算法 3 在决策模块选定具体动作后执行, 选择的动作即下一转发节点。由于 DQN 根据当前状态下各可能动作的 Q 值进行评估并选择最大 Q 值对应的动作作为最优策略, 这个动作可能已经在当前路径中或者与当前节点之间不直连。算法 3 首先对此做出判断, 若选择的节点已在路径中, 可能导致路径形成环路或重复跳转, 则重新基于当前状态选择下一节点并给予负奖励; 若不在, 则在环境执行动作。判断当前节点与下一节点是否直连, 若直连, 直接从当前带宽邻接矩阵和

成本邻接矩阵中得到该链路的带宽、成本和时延值；若不直连，根据成本邻接矩阵计算两点间转发成本最低的路径，带宽邻接矩阵和当前请求累加成本、时延根据两点间的成本最低路径更新，最后将选择的动作加入路径中得到奖励并判断是否到达目的节点。

算法 3 路径调整算法

输入：当前节点，下一节点，目的节点，当前累加成本 $cost$ ，当前 k , 当前 $path$

输出：奖励，是否到达目的节点，更新后的累加成本 new_cost ，更新后的 new_k ，下一状态，更新后的 $path$

- 1.if 下一节点已在路径中
 2. **return** 负奖励，未到达目的节点，下一状态
 - 3.end if
 - 4.if 当前节点与下一节点直连
 5. 由带宽、成本、时延邻接矩阵更新带宽、累加成本、累加时延
 - 6.else
 7. 根据成本邻接矩阵计算当前节点与下一节点的成本最小路径，更新带宽、累加成本、累加时延
 - 8.end if
 - 9.将下一节点加入 $path$ 中
 10. 判断是否到达目的节点
 - 11.根据更新后的带宽邻接矩阵计算 k
 - 12.使用公式(3-7)计算奖励
 - 13.**return** 奖励，是否到达目的节点，更新后的累加成本 new_cost ，更新后的 new_k ，下一状态，更新后的 $path$
-

3.3.2 算法复杂性分析

流转发路径探索算法在请求的优先级排序阶段，共有 $|F|$ 条请求到达，需要双层遍历构建模糊一致判断矩阵，即算法 1 所示，而算法 1 需要进行 $|F| \times |F|$ 次循环，所以算法 1 的时间复杂度为 $O(|F|^2)$ 。在对选择的动作调整时，调用算法 3，其时间复杂度为 $O(|V| \times \log|V|)$ ， $|V|$ 为物理节点数量。因此本文所提出的算法时间复杂度为 $O(num_episode \times |V|^2 \times \log|V|)$ ，其中 $num_episode$ 为训练迭代次数。

3.4 仿真设计与性能分析

本节对提出的基于模糊层次分析的流转发路径探索算法进行仿真和性能分析，为全面准确公正地衡量算法的性能表现，本研究引入了两种现有负载均衡算法作为对比基准，并从负载均衡系数、转发成本、优化总目标、请求迟到数的维度进行分析。

3.4.1 实验环境和参数设置

本文采用由 ROCKETFUEL 收集的拓扑 Ebone^[29]，该拓扑含有 23 个节点，76 条链路，链路成本值、带宽值和时延值根据文献[29]设置。每个请求通过选择 V 中的节点作为源节点 S_k 和目的节点 T_k ，随机选择流类型 t_k ，流类型根据文献[59]设置，同时根据文献[59]分配不同流类型的带宽和时延要求。本文根据拓扑节点、流类型以及对应类型的带宽时延条件设置了请求生成器函数，可以根据所需的请求数量生成请求。本文优化目标是最小化负载均衡系数和转发成本之和，在实验中， α 设为 0.5，表明负载均衡系数和转发成本在总目标中具有同等权重。在 DQN 算法中，使用 Pytorch 构建了 3 层全连接网络，输入层与状态空间维度相同，隐藏层包含 64 个神经元，输出层与动作空间维度相同，其余参数如表 3-3 实验参数所示。

表 3-3 实验参数

参数	值
请求类型	视频、游戏、文件、VoIP
请求带宽要求	10Mbps~120Mbps
请求时延要求	40ms~400ms
目标权重因子	$\alpha=0.5$
初始探索率	0.9
最小探索率	0.05
折扣因子	0.9
批量采样大小	32

在调用 FAHP 算法对请求进行优先级排序时，对于同一请求中的三个 QoS 指标进行了重要性评估，本文考虑的模糊矩阵标度从高到低为：时延、流类型、带宽，这种标度设置反映了在多媒体流服务场景下，对低延迟的偏好，计算得到三个指标的权重向量 $\omega=(0.383,0.334,0.283)^T$ ；为进一步提升排序的进度和合理性，本文还将不同请求在同一指标上做比较，本文考虑的模糊矩阵标度从高到低为：游戏、视频、文件、VoIP，这一顺序反映了不同应用类型对网络性能的敏感程度差异。通过调用 FAHP 算法，可以得到一个较为合理的优先级。以实验中的 5 个请求为例，算法调用前后的请求处理顺序如表 3-4 所示。可以看到，具有较高时延要求的请求 5 能够被赋予高优先级从而获得优先调度权。原始处理顺序中，请求是按照其到达时间进行排队，未充分考虑其服务质量需求。调用 FAHP 算法后，请求的处理顺序发生了明显变化，体现了算法对多 QoS 指标综合考虑的能力。

表 3-4 处理顺序对比

原始处理顺序	调用 FAHP 后处理顺序
id:1;S ₁ :10;T ₁ :4;d ₁ :184;b ₁ :64;t ₁ :VoIP	id:5;S ₅ :13;T ₅ :17;d ₅ :45;b ₅ :76;t ₅ :游戏
id:2;S ₂ :3;T ₂ :12;d ₂ :130;b ₂ :69;t ₂ :视频	id:3;S ₃ :1;T ₃ :5;d ₃ :50;b ₃ :54;t ₃ :游戏
id:3;S ₃ :1;T ₃ :5;d ₃ :50;b ₃ :54;t ₃ :游戏	id:2;S ₂ :3;T ₂ :12;d ₂ :130;b ₂ :69;t ₂ :视频
id:4;S ₄ :5;T ₄ :14;d ₄ :223;b ₄ :61;t ₄ :文件	id:1;S ₁ :10;T ₁ :4;d ₁ :184;b ₁ :64;t ₁ :VoIP
id:5;S ₅ :13;T ₅ :17;d ₅ :45;b ₅ :76;t ₅ :游戏	id:4;S ₄ :5;T ₄ :14;d ₄ :223;b ₄ :61;t ₄ :文件

3.4.2 对比算法介绍

为了确保比较的有效性和公正性，本文基于同一仿真环境和评估标准，选取了文献[54,60]中两种被认可的算法作为比较对象：

Q-Learning Load Balancing (QLLB): 该方法是一种基于 Q-Learning 的智能负载均衡算法，通过 Q 表存储 Q 值，不断学习网络状态，优化任务调度策略，能够在一定程度上减少网络拥塞。

Hierarchical Quality of Service (HiQoS): HiQoS 是一种分层 QoS 感知调度算法，侧重于通过分层决策机制优化不同优先级流量的调度，采用 Dijkstra 算法选择最短路径，在一定程度上提高了资源利用率和 QoS 保障能力。

3.4.3 评价指标

负载均衡系数 k ，表征网络拓扑上某一时刻各链路带宽利用的平衡程度，由式(3-4)所得， k 越小，表明链路越均衡；转发成本，即所有请求转发路径的成本之和，可以由式(3-5)所得；优化总目标，本文的优化总目标是最小化负载均衡系数和转发成本之和，定义如式(3-6)所示；请求迟到数，即在转发过程中，超出请求时延要求的请求数量。

3.4.4 实验结果与分析

本章实验的请求数量从 5 条以公差 5 递增至 50 条，在不同请求规模下比较算法性能。

负载均衡系数 k 对比：如图 3-5 所示本文算法的平均负载均衡系数分别比 HiQoS 算法和 QLLB 算法低 15.54%和 5.56%。由于 HiQoS 算法采用 Dijkstra 得到一对源目的节点间的间隔最短路径，使得具有相同源目的节点对的请求（如：节点 3-节点 9）或包含这个源目的节点对（如：节点 2-节点 3-节点 9-节点 10）的请求将重复使用节点 3-节点 9 的同一路径，造成网络拥堵，因此 HiQoS 算法下负载均衡系数高于其他两种算法。QLLB 采用 Q 表存储和访问 Q 值，通过遍

历所有可能的状态-动作对进行学习，当请求数量增多时，状态-动作对增多，导致 Q 表访问效率下降，存在超出请求时延要求的情况，导致网络拥堵，因此负载均衡系数高于本文算法。

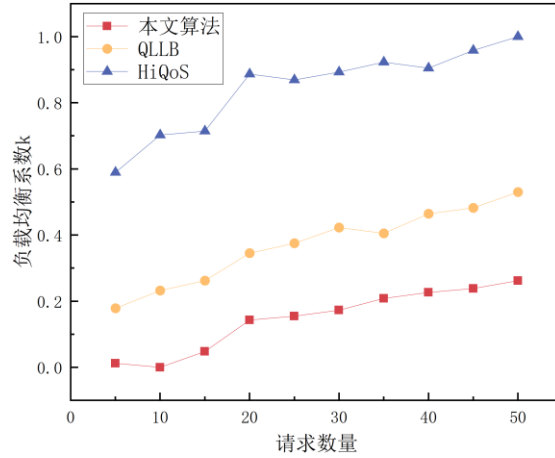


图 3-5 负载均衡系数 k 的对比

转发成本对比：链路的转发成本表征使用该链路的所有代价总和。如图 3-6 所示，本文算法的平均转发成本比 QLLB 算法低 10.2%。这种优势主要来源于路径选择策略的差异，与本文算法相比，QLLB 算法对于不直连的两个节点间的路径选择，采取随机路径选择机制，未能充分考虑链路的代价，从而导致整体转发成本偏高。而本文算法选择两点间成本最低路径，因此本文算法的平均转发成本比 QLLB 算法低。而 HiQoS 算法对所有请求源节点和目的节点都采用最短路径，因此转发成本低于本文算法。

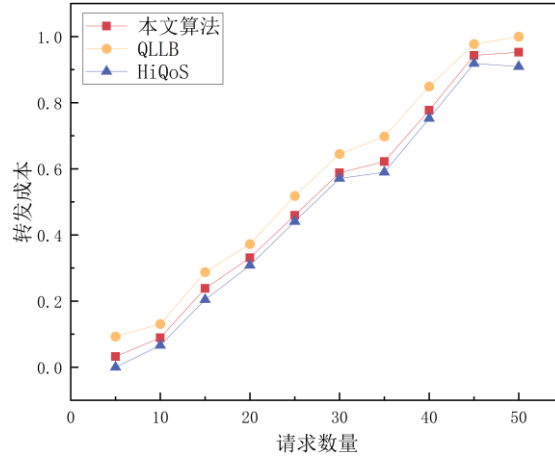


图 3-6 转发成本的对比

总目标对比：本文优化总目标是在保证网络性能的前提下，尽可能地最小化负载均衡系数和转发成本之和。如图 3-7 所示，本文算法的平均总目标比 QLLB 算法和 HiQoS 分别优化了 6.49%和 9.74%。通过比较，本文基于 FAHP 和 DQN 的流转发路径探索算法在提高负载均衡程度的同时又不使转发成本过高，很好的平衡了二者的关系。

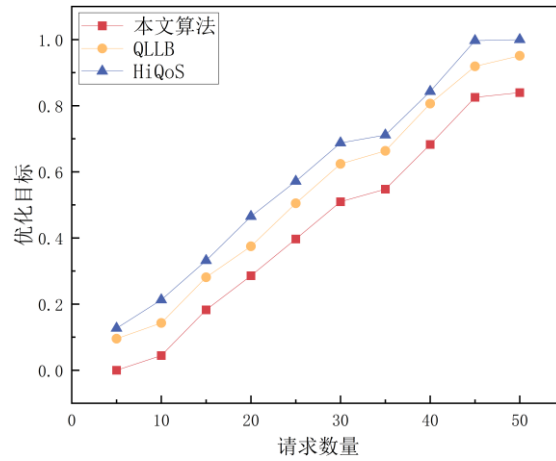


图 3-7 优化总目标的对比

请求迟到数对比：如图 3-8 所示，当请求数量增大时，本文流转发路径探索算法的请求迟到数均低于 QLLB 算法和 HiQoS 算法。当请求数量未到 20 前，由于 HiQoS 算法对流量做了差分处理，具有高时延要求的请求会得到优先处理，因此其请求迟到数低于 QLLB 算法，但当请求数量达到 20 以后，由于 HiQoS 算法优先考虑最短路径转发，导致网络中存在最短路径路段的拥堵，从而其请求迟到数又高于 QLLB 算法。综上，本文所提出的算法能够有效缓解多媒体场景中请求因优先级划分不合理而导致网络堵塞的问题。

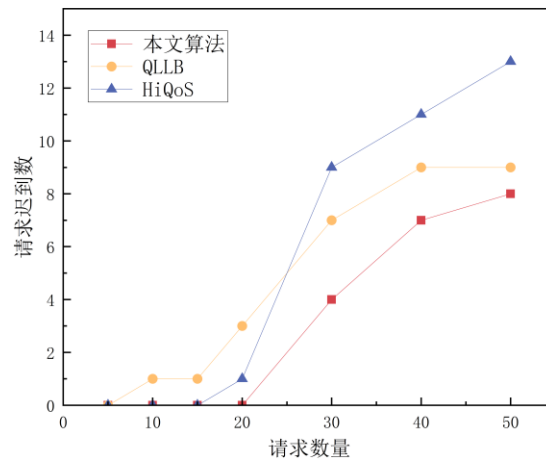


图 3-8 请求迟到数的对比

3.5 本章小结

针对 SDN 架构在多媒体流服务场景中的应用，网络链路负载不均衡会导致部分链路过载，而其他链路利用率较低，进而影响整体网络性能和 QoS 保障。在多媒体流服务中，数据流量通常具有动态性和多样化的服务质量需求，使得流量调度的复杂性增加。此外，在优化网络流量调度时，链路负载均衡和转发成本存在固有矛盾：一方面，提高负载均衡程度需要将流量更均匀地分布到不同路径，

以避免部分链路超载,但这可能导致数据包绕行,从而增加链路转发成本和时延;另一方面,降低转发成本通常倾向于选择最短路径,但这可能导致部分链路超载,使网络拥塞加剧,进一步影响 QoS 保障。因此,如何在保证服务质量的同时合理权衡链路负载均衡和转发成本,成为多媒体流量调度优化中的关键问题。

为优化这一问题,本章将问题建模为多目标多约束模型。模型以最小化链路负载均衡系数和链路转发成本为目标,同时满足各请求的带宽、时延要求和物理链路带宽资源约束,提出了基于 FAHP 和 DQN 的流转发路径探索算法。实验表明,与 QLLB 和 HiQoS 算法相比,本文算法可以更好地平衡链路转发成本和链路负载均衡程度之间的关系,在提高链路负载均衡程度的同时降低链路转发成本。

4 基于最小费用最大流的转发路径动态调整算法

4.1 引言

在多媒体流服务场景下，不同类型的流量对网络资源的需求存在较大差异。例如，高清视频流对带宽需求较高，而在线游戏对时延要求更为严格。针对这一问题，在上一章的工作中，本文提出了基于模糊层次分析的流转发路径探索算法，算法基于 FAHP 和 DQN 技术，首先利用 FAHP 量化多媒体流的 QoS 需求，计算请求的优先级，并按照优先级顺序进行路径探索。随后，结合 DQN 强化学习方法，使 SDN 控制器能够动态探索各个节点以得到最终的转发路径，在节点选择时综合考虑高链路负载均衡程度、低转发成本和资源利用率最大化，以更优地平衡两者之间的矛盾。

然而，在大规模流量并发的情况下，仅依靠逐点优化的方式可能难以从全局视角协调多个流的资源调度。因此，为了进一步提升网络资源的整体调度效率并优化转发成本开销，本章提出了一种基于最小成本最大流与深度强化学习协同优化的流转发路径动态调整算法。算法构建了全局规划引导的动态调整双阶段优化框架，通过经典图论与深度强化学习的有机融合，构建兼顾全局经济性和平衡性的路径调整机制。算法对请求路径做全局调整，由 MCMF 首先生成请求的最小费用最大流转发路径方案，再使用深度强化学习(DRL)基于链路负载均衡程度、时延等指标动态调整路径，有效降低网络拥塞和传输时延。

4.2 问题描述与建模

4.2.1 问题描述

在大规模流量并发的网络环境中，仅依赖逐点路径选择的局部优化策略，难以从全局视角有效协调多请求之间的资源分配。针对这一问题，本章在上一章工作的基础上，进一步聚焦于并发请求调度的全局协调优化。

4.2.2 系统框架

本章提出了基于 MCMF-DRL 协同的流转发路径动态调整算法，本章算法的 SDN 路由框架如图 4-1 所示：

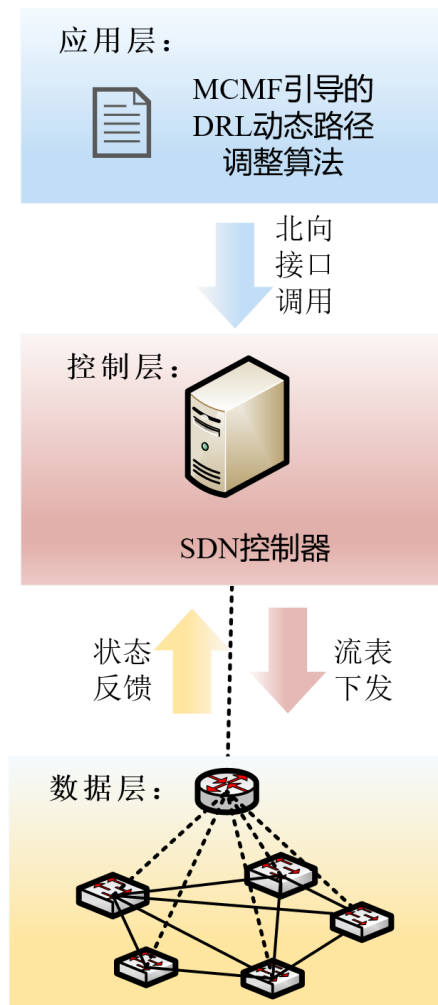


图 4-1 SDN 路由框架

本章中仍然将控制器内部划分为链路感知模块、深度强化学习模块、流表下发模块，但在深度强化学习模块中对转发路径生成策略进行了关键性的改进，如图 4-2 所示。本章延续了上一章工作中的 QoS 优先级排序，但本章中将 FAHP 算法生成的每个请求的权重带入了深度强化学习阶段，作为两个请求争用同一链路时，强化学习智能体进行决策的重要依据，从而提升了调度过程中的公平性与 QoS 保障能力。此外，将 MCMF 模块生成的请求初始路径集合，作为深度强化学习的输入，引导强化学习往成本更优方向学习负载更均衡的转发策略。这种策略下，智能体是在全局较优解的基础上，结合实时网络状态进行路径优化。这种以全局解为启发、局部自适应调整的方式，不仅能加快学习收敛速度，减少无效探索路径，还能引导强化学习在保持较低转发成本的同时，主动规避高负载链路，从而学习到更加均衡且动态适应网络变化的转发策略。

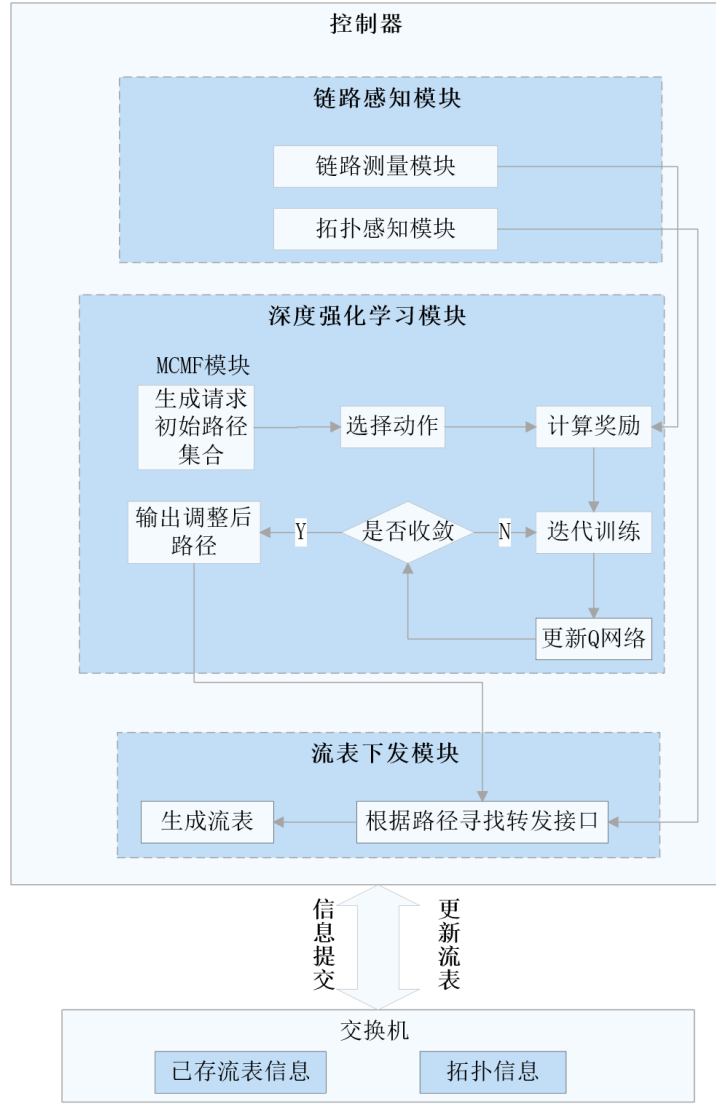


图 4-2 系统处理模型框架

4.2.3 问题建模

1、底层物理网络模型

本章底层物理网络用无向图连通图 $G(V, E)$ 表示，其中 V 表示物理节点的集合， E 表示物理链路的集合， $n=|E|$ 表示链路数量。 (i, j) 表示节点 i 、 j 间的链路， L_{ij} 、 B_{ij} 、 D_{ij} 、 C_{ij} 分别表示链路 (i, j) 上的链路编号、带宽、时延、单位带宽成本。请求流集合由 F 表示，请求 $f(f \in F)$ 由一个五元组表示 $(S_f, T_f, b_f, d_f, t_f)$ ， S_f 表示请求 f 的源节点， T_f 表示目的节点， b_f 表示请求 f 的带宽要求， d_f 表示请求 f 允许的最大时延， t_f 表示请求 f 的类型。 r_f 表示请求 f 的最终转发路径的链路集合， P_f 表示请求 f 的初始转发路径的链路集合， L_{ij}^f 表示请求 f 在链路 L_{ij} 上的需求带宽。

2、相关约束条件

问题约束和调整过程需要满足的约束如下：由于每条链路上的带宽资源是有

限的，每条链路只能支持有限个请求。因此，可以得到约束：

$$\sum_{f \in F} \sum_{L_{ij} \in r_f} L_{ij}^f \leq B_{ij}, \forall i, j \in V \quad (4-1)$$

得到请求 f 的最终转发路径 r_f 时，请求 f 的带宽和时延要求要得到满足，因此，可以得到约束：

$$\sum_{L_{ij} \in r_f} L_{ij}^f \geq b_f \quad (4-2)$$

$$\sum_{L_{ij} \in r_f} D_{ij} \leq d_f \quad (4-3)$$

3、优化目标

为了有效应对网络中的流量变化，保证网络服务的质量和畅通，路径调整至关重要。其中，如何实现在动态调整过程中，既满足用户 QoS 要求，又控制整体的转发成本，还要优化整体带宽资源的使用分布是本章算法要解决的问题，上文设定了确保用户 QoS 满足的约束条件，下面将介绍带宽资源分布的定义，构建一个综合考虑带宽资源使用分布、用户需求满足程度和转发成本效益最优化的多目标多约束优化问题模型。

带宽资源使用是否均衡是衡量整个网络是否均衡的重要指标。定义每一条链路 (i,j) 上已用带宽与总带宽之比为带宽利用率 BR_{ij} ：

$$BR_{ij} = \frac{\sum_{f \in F} \sum_{L_{ij} \in r_f} L_{ij}^f}{B_{ij}} \times 100\%, \forall i, j \in V \quad (4-4)$$

标准差是直观衡量数据分布离散程度的重要统计指标，因此本章定义各链路带宽利用率的标准差为负载均衡系数 k ，以衡量整个网络的负载均衡程度：

$$k = \sqrt{\frac{\sum_{(i,j) \in E} (BR_{ij} - \overline{BR})^2}{n - 1}} \quad (4-5)$$

转发成本指数数据流在网络中从源节点传输到目的节点时所消耗的资源，即 MCMF 问题中的费用，本章定义 t 时刻单个请求 f 转发成本为 C_f ， $L_{ij}^f(t)$ 表示请求 f 在 t 时刻对链路 (i,j) 上的带宽需求为：

$$C_f(t) = \sum_{L_{ij} \in r_f} L_{ij}^f(t) \times c_{ij} \quad (4-6)$$

则总转发成本 C_{total} 为：

$$C_{total}(t) = \sum_{f \in F} C_f(t) \quad (4-7)$$

而在动态调整的时间 T 内，可以定义长期总成本为：

$$C = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T C_{total}(t) \quad (4-8)$$

在 SDN 网络负载均衡策略中，为避免某些链路出现拥塞，流量可能被分配

到次优路径。例如，一条最低成本路径可能因负载较高而不被选用，而流量被引导至更长或更高成本的路径，从而增加整体的转发成本。而在传统的最短路径优先或最低费用路径选择策略中，流量倾向于被引导到最低代价的路径。这种方式可能导致部分链路过载，而其他链路处于空闲状态，造成网络资源利用率不均衡，进而影响整体网络性能。因此权衡负载均衡程度和转发成本之间的矛盾是 SDN 流量调度中的核心问题，本章将围绕这一问题优化，因此优化目标可以整理为：

$$\begin{aligned} & \min k + C \\ & \text{s. t.} \left\{ \begin{aligned} & \sum_{f \in F} \sum_{L_{ij} \in r_f} L_{ij}^f \leq B_{ij}, \forall i, j \in V \\ & \sum_{\forall L_{ij} \in r_f} L_{ij}^f \geq b_f \\ & \sum_{\forall L_{ij} \in r_f} D_{ij} \leq d_f \end{aligned} \right. \end{aligned} \quad (4-9)$$

其中，优化目标为最小化动态调整的长期总成本和负载均衡系数，且约束条件满足式(4-1)-(4-3)。本章涉及的主要参数及意义如表 4-1 所示。

表 4-1 主要参数及意义

类型	参数
$G(V,E)$	底层物理网络
V	物理节点的集合
E	物理链路的集合
(i,j)	节点 i 、 j 之间的链路
$B_{ij}, D_{ij}, C_{ij}, L_{ij}$	链路 (i,j) 上的带宽、时延、路由成本、编号
F	请求流的集合
S_f	请求 f 的源节点
T_f	请求 f 的目的节点
B_f	请求 f 的带宽要求
D_f	请求 f 允许的最大时延
T_f	请求 f 的类型
L_{ij}^f	请求 f 在链路 L_{ij} 上的需求带宽
r_k	请求 f 的最终转发路径的链路集合
P_f	请求 f 初始转发路径的链路集合
BR_{ij}	链路 (i,j) 的带宽利用率
k	网络的负载均衡系数

4.3 流转发路径动态调整算法

为了求解以上优化问题，本章提出了 MCMF-DRL 协同的流转发路径动态调整算法。本章所提算法主流程如图 4-3 所示，本节将详细介绍该算法设计原理。

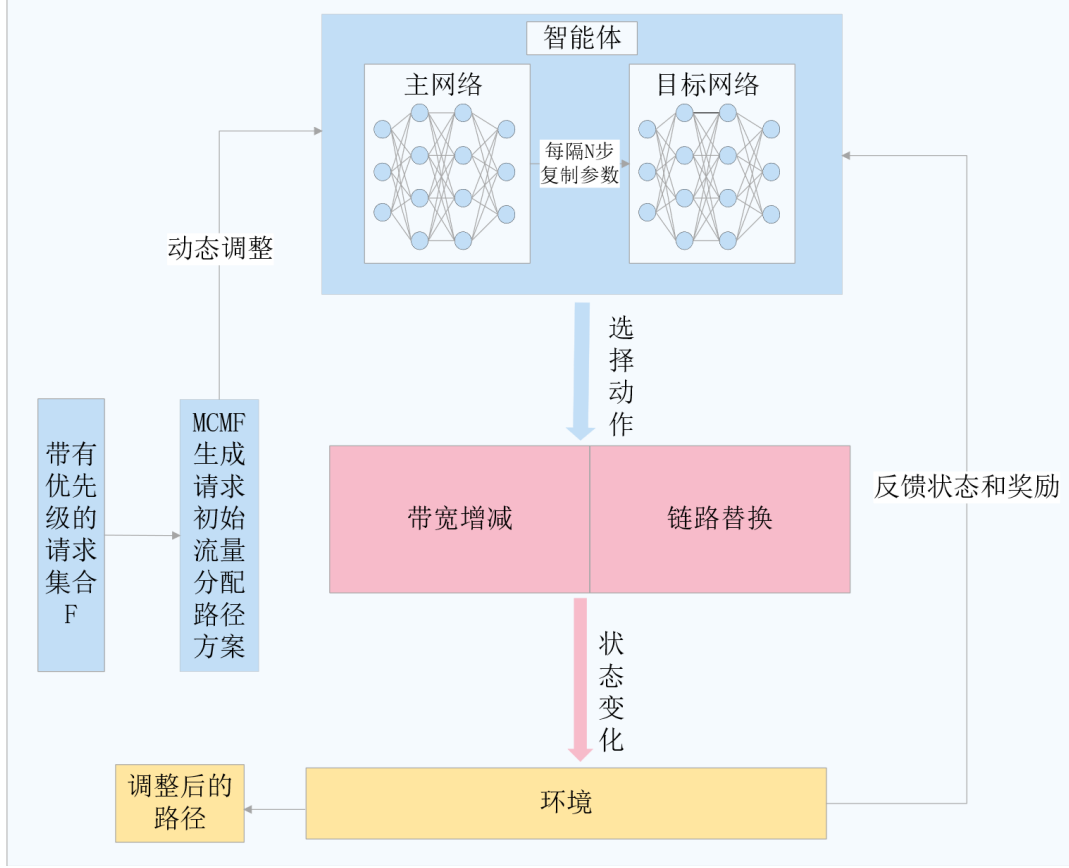


图 4-3 流转发路径动态调整算法流程图

其中，MCMF 生成初始流量分配路径方案的具体流程如图 4-4 所示，从初始化所有边流量为 0 开始，构建残量网络并反复寻找源点到汇点的最短路径，若存在路径则计算调整量并更新流量，直到无法增广为止。

调整量 θ 计算方式如式(4-10)所示，确保在满足容量约束限制的前提下，尽可能多地调整流量，以尽快逼近最大流。

$$\theta = \min(\min_{\mu^+}(c_{ij} - f_{ij}^{k-1}), \min_{\mu^-}(f_{ij}^{k-1})) \quad (4-10)$$

在 μ 上调整流量的调整策略如式(4-11)所示，通过正向边增加流量，反向边减少流量，以探索更优路径和费用最小化。

$$f_{ij}^k = \begin{cases} f_{ij}^{k-1} + \theta, & (v_i, v_j) \in \mu^+ \\ f_{ij}^{k-1} - \theta, & (v_i, v_j) \in \mu^- \\ f_{ij}^{k-1}, & \text{其他边} \end{cases} \quad (4-11)$$

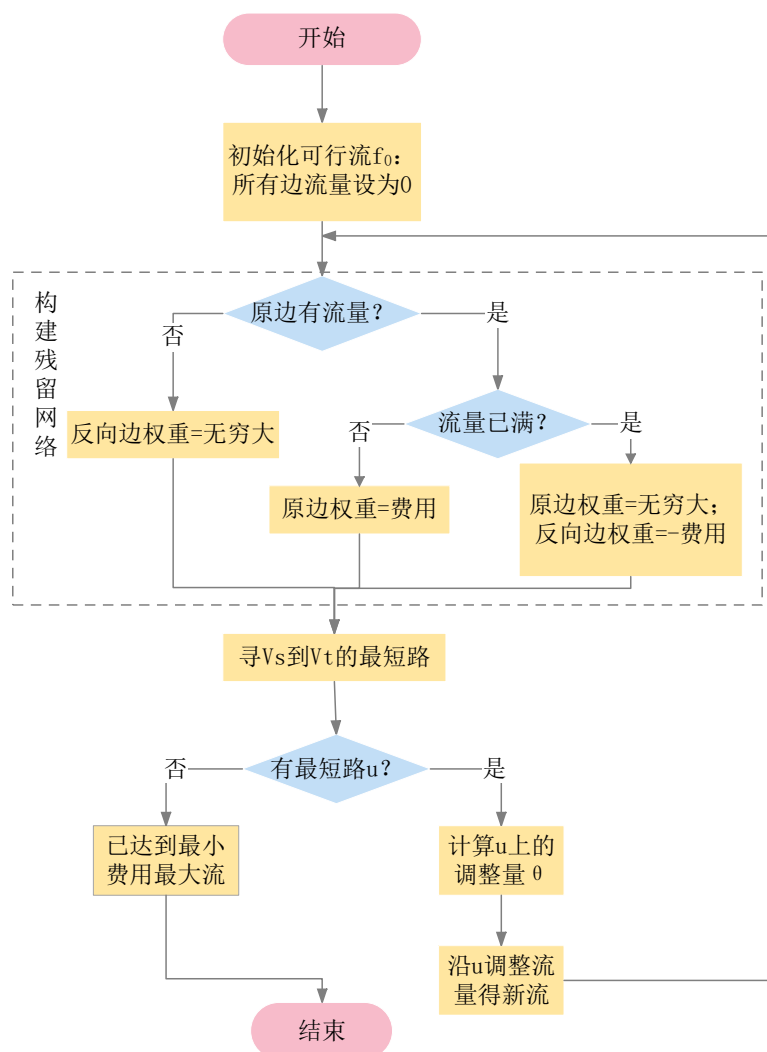


图 4-4 MCMF 流程图

在面对请求路径动态调整问题时，传统的动态规划方法在理论上能够求解最优解，然而，在实际网络环境中，动态规划的应用受到诸多限制。其主要依赖于对完整环境模型的先验知识、包括状态转移概率和决策过程的精确描述。然而，在复杂的网络环境中，链路可用带宽等资源是动态变化的，且这些变化具有随机性，因而难以提前获知完整的状态转移概率，导致问题的状态空间和动作空间变得庞大。而动态规划方法在处理大规模状态空间时，计算复杂度呈指数级增长，导致存储和计算成本急剧上升，难以高效应对复杂的流量调度问题。动态规划方法的这些局限使得其难以适应于环境模型不完整或状态空间庞大的动态环境，如请求转发路径的动态调整问题。

在此背景下，强化学习方法因其适用于大规模状态空间，无需提前构建完整环境模型的特点，成为解决该问题的一种有效方法。它通过智能体与环境的交互不断调整决策策略，并基于环境反馈的奖励优化其行为，逐步学习出最优策略。相比于传统优化方法，强化学习能够在不确定环境下进行自主决策，且能够在探

索和利用之间找到平衡，从而有效应对网络状态的动态变化。然而，经典的强化学习方法在面对高维状态空间时，存储 Q 值表的需求过大，难以适用于复杂的网络环境。因此，为了进一步提升算法的求解能力，本章采用 DRL 来学习调整策略。DRL 结合了强化学习在无模型情况下通过探索与利用学习最优策略的优点，以及深度学习在处理高维数据方面的优势，使其能够在大规模、复杂动态的网络环境中高效求解流转发路径优化问题。

4.3.1 算法描述

流转发路径动态调整算法沿用了上一章工作中对初始请求的排序结果，将排序后的请求集合作为 MCMF 模块的输入对象，按序为请求计算初始带宽分配路径。为适应 MCMF 算法求解时常用有向图模型，在调用 MCMF 算法前首先对网络拓扑的每条物理链路创建双向虚拟边，两边的带宽容量、单位成本相等，从而将无向边转换为有向边用于求解。

获得请求初始流量分配路径方案的过程如算法 1 所示。在实际的网络环境中，链路的状态总是随着时间动态变化，例如，某些链路可能由于拥塞等原因出现带宽降低、延迟增加等情况，有些链路可能经过优化或调整后性能得到提升。通过在每一步处理请求前更新网络拓扑数据，算法能够及时捕捉这些链路状态的动态变化，从而更准确地反映当前网络的实际状况，为后续的路径选择和流量分配提供可靠信息。

因此本算法在处理每个请求前首先更新网络拓扑数据，再获取当前请求的源点与终点，初始化可行流、成本以及记录链路带宽占用的数组（第 1-6 行）。在存在增广路径的情况下，调用 SPF 算法寻找成本最短路径，若找到则计算路径上的最小剩余容量，即构建残留网络，并更新链路带宽占用、可行流和成本，重复此过程直至找不到增广路径（第 7-16 行），最后输出每个请求的最大流、最小费用、初始转发路径及路径中每段链路占用的带宽，循环处理所有请求后返回整体的最大流、最小费用及相关路径和带宽信息。算法 1 给出了请求的最小成本最大流初始流量分配方案，为后续智能体调整路径提供方向指引，使得智能体在调整路径时能够基于一个较为优化的基础进行优化，减少了盲目搜索所带来的时间和资源浪费。

算法 1 最小费用最大流（MCMF）

输入： 转换后的网络拓扑 G^* , 排序后的请求集合 F

输出： 每个请求的初始转发路径及路径中每段链路占用的带宽、及该方案的最大流、最小费用

1. for $i=1$ to $|F|$
 2. 更新网络拓扑数据
-

```

3.   获取当前请求  $f_i$  的源点  $S_i$  与终点  $T_i$ 
4.   初始化可行流  $F_i=0$ , 成本  $C_i=0$ 
5.   初始化所有链路  $L_{ij}$  占用带宽  $f(L_{ij})=0$ 
6.   初始化链路带宽占用记录  $links\_flow=[]$ 
7.   while 存在从  $S_i$  到  $T_i$  的增广路径
8.       调用  $SPF(G^*, S_i, T_i)$  寻找成本最短路径  $P$ 
9.       if 无法找到  $P$ 
10.          break
11.       计算路径  $P$  上的最小剩余容量  $\theta$ 
12.       for  $(i,j) \in P$ 
13.          更新  $f(L_{ij})=f(L_{ij})+\theta$ , 反向边  $f(L_{ji})=f(L_{ji})-\theta$ 
14.          记录链路带宽占用  $links\_flow[L_{ij}]=f(L_{ij})$ 
15.       更新可行流  $F_i$  和成本  $C_i$ 
16.   输出请求  $f_i$  的最大流  $F_i$ 、最小费用  $C_i$ 、初始转发路径及路径中每段链路占用的带宽
17. end for
18. return 所有请求的最大流、最小费用、初始转发路径及路径中每段链路占用的带宽
    
```

接着流转发路径调整算法需要提供合理的调整策略,以适应网络环境动态变化的需求。为了将现有的优化问题适配到 DRL 方法中,需要将优化问题转化为一个马尔可夫决策过程,其核心包括状态空间、动作空间、状态转移概率、奖励函数和折扣因子,可以由五元组对应表示为 (S, A, P, R, γ) 。在本章问题中,马尔可夫决策过程的建模如下:

状态空间: 表示网络在某一时刻的特定状态。在路径动态调整问题中状态空间包含三类信息。定义如式(4-12)所示:

$$S(t) = \{D(t), l(t), BR(t)\} \quad (4-12)$$

其中, $D(t)$ 表示 t 时刻网络中各请求的转发时延向量,其公式表达为:

$D(t) = [D_{f_1}(t), D_{f_2}(t), \dots, D_{f_{|F|}}(t)], f_i \in F$, $|F|$ 为请求数量,其中 $D_{f_x}(t) = \sum_{L_{ij} \in \pi_{f_x}} D_{ij}$, π_{f_x} 表示智能体对流请求 f_x 得到的最终路径。 $l(t)$ 则表示 t 时刻各链路上的剩余带宽资源的向量,表达式为 $l(t) = [l_{00}(t), l_{01}(t), \dots, l_{ij}(t)], i, j \in [0, |V|-1]$ 。 $BR(t)$ 表示 t 时刻各链路上的带宽利用率向量,表达式为 $BR(t) = [BR_{L_{ij}}(t)], L_{ij} \in E$, 则 $BR(t)$ 向量中各元素的标准差即为 t 时刻网络的负载均衡系数 $k(t)$, 体现当前时刻整体网络的负载均衡程度。 $|V|$ 为节点数量, $|E|$ 为链路数量, 因此, 状态空间大小为 $|E| + |F| + |V| \times |V|$ 。这些信息的整合, 能够更全面地刻画当前网络状态, 对于资源分配决策和请求转发路径调整具有重要作用。

动作空间: 本章算法针对请求的初始转发路径得到调整策略, 智能体根据当前状态 $S(t)$ 对带宽需求不同的请求集合, 选择最佳的路径调整策略, 因此动作空间是 $|F|$ 维向量, 即:

$$A(t) = \{a_1(t), a_2(t), \dots, a_{|F|}(t)\} \quad (4-13)$$

其中, $a_f(t)$ 表示在 t 时刻请求 f 选择的动态调整策略。

状态转移概率: 在马尔可夫决策过程中, 状态转移概率指在状态 $S(t)$ 下执行动作 a 后转移到新状态 $S(t+1)$ 的概率分布。在多媒体流量调度问题中, 网络状态是动态变化的, 链路负载、可用带宽、请求分布等因素具有随机性, 使得状态转移概率无法提前建模。深度强化学习能够通过智能体与环境的交互, 在不依赖显式状态转移概率的情况下, 逐步优化流量调度策略。

奖励函数: 对于每一次状态转移, 环境都会反馈一个奖励给智能体, 通过奖励函数对所选的路径调整方案进行评价, 促使智能体选择更优的流量调度策略。强化学习的目标是长期优化, 因此, 奖励函数应当引导智能体学习如何在即时奖励和长期效益之间权衡。对于即时奖励, 由于在本章动态调整问题中, 优化目标为满足时延要求同时最小化转发成本和负载均衡系数, 而智能体的目标是最大化长期奖励, 因此, 本章中将奖励设置为转发成本、负载均衡系数和时延的负值之和, 对于调整失败的惩罚设置为 $-\sigma$, σ 设置为远大于转发成本、负载均衡系数和时延之和的数, 如式(4-14)所示:

$$r(t) = \begin{cases} -\sigma, & \text{调整失败} \\ -C_{total}(t) - k(t) - D(t), & \text{调整成功} \end{cases} \quad (4-14)$$

折扣因子: 由第二章强化学习部分介绍的未来奖励可知, 强化学习利用折扣因子 γ , 表示对未来奖励的考虑程度, 未来奖励可表示为式(4-15):

$$R(t) = r(t) + \gamma \cdot r(t+1) + \gamma^2 \cdot r(t+2) + \dots = \sum_{n=0}^{\infty} \gamma^n \cdot r(t+n) \quad (4-15)$$

其中, n 为迭代次数。

接下来本文将围绕流转发路径动态调整算法展开进一步探讨。如算法 2 所示, 描述了流转发路径动态调整算法的具体过程。算法开始时通过调用算法 1 得到请求的初始流量分配方案 (第 1 行)。第 2 至 18 行是算法训练的过程, 首先初始化环境和一个经验池 (第 2 至 3 行), 以生成状态空间和存储经验 (S_t, a_t, S_{t+1}, r)。同时初始化主网络和目标网络的参数 (第 4 至 5 行)。接着对调整算法进行 $num_episode$ 次训练, 针对每一次训练, 初始化状态并对请求路径进行以下步骤:

- 1、在主网络中通过 ϵ 贪婪策略选择动作 (第 9 行);
- 2、根据选择的相应动作, 调用相应的子算法对请求路径进行对应的调整 (第 10-13 行);
- 3、根据选择的动作调整后获得奖励和下一状态, 并存入经验池 (第 14 行), 其中, 奖励通过式(4-14)计算;

- 4、当经验池中有足够的经验时，从中随机采样一批经验样本，使用梯度下降法更新主网络参数（第 15 行）；
- 5、在达到更新步长后，将主网络参数复制给目标网络（第 16 行）。

训练结束后，保存训练的网络模型，将请求集合的初始流量分配方案输入算法，可得到每个请求调整后的路径。

算法 2 流转发路径动态调整算法

输入：训练次数 $num_episode$ ，更新步长 $step$ ，折扣因子 γ ，带宽利用率阈值 α

输出：请求的最终转发路径 $path$ ，转发成本 $cost$ ，负载均衡系数 k

- 1.调用算法 1，得到请求集合 F 的最小费用最大流带宽分配方案
 - 2.初始化拓扑环境
 - 3.初始化经验池 $ReplayBuffer$
 - 4.初始化主网络参数 θ
 - 5.初始化目标网络参数 $\theta' = \theta$
 - 6.for episode= 1 to num_episode
 7. 初始化状态空间
 8. for i=1 to $|F|$
 9. 在主网络中采用 ε -Greedy 策略选择动作 A
 10. if $A=1$ then
 11. 调用算法 3
 12. else if $A=2$ then
 13. 调用算法 4
 14. 获得奖励 r 和下一状态 S_{t+1} ，将 (S_t, a_t, S_{t+1}, r) 放入经验池
 15. 从经验池中随机采样一批经验，更新主网络参数
 16. 若达到更新步长，则将主网络参数 θ 复制到目标网络 θ'
 17. end for
 18. end for
 - 19.通过训练好的网络得到所有请求调整后的路径 $\pi = \{\pi_1, \pi_2, \dots, \pi_{|F|}\}$ ， $cost, k$
-

对于动作的执行，若选取的动作是为请求路径在当前链路上进行带宽资源调整，即在不改变转发路径结构的前提下，尝试优化带宽分配策略，则将会调用算法 3 执行相应的调整操作。首先判断请求的初始路径方案中，在该链路上所需的带宽是否小于或等于当前链路的可用带宽资源。若是，说明链路尚有足够的容量容纳该请求的带宽需求。接着判断，若完成此次带宽分配调整后，该路径上的带宽利用率是否不会高于阈值 α ，若是，则在当前链路上调整带宽分配，返回调整后的状态、奖励、和调整后的路径方案；否则，调整失败。

算法 3 带宽增减算法

输入：请求 f_m 的原路径方案 P_m

输出：调整后的路径方案 π_m

- 1.for L_{ij} in P_m
 2. if $L_{ij}^f \leq l_{ij}(t)$ and $BR_{ij}(t) \leq \alpha$
 3. 在当前链路上增加或减少带宽分配
-

```

4. else
5.     return 奖励, 调整失败, 新的状态,  $\pi_m$ 
6. end if
7. end for
8. return 奖励, 调整成功, 新的状态,  $\pi_m$ 
    
```

若选取的动作是替换请求转发路径上的链路, 如算法 4 所示, 首先查找原路径方案中每条链路的备选子路径集合, 备选子路径是指在当前网络拓扑结构下, 连接相同起止节点且具备可行性的其他路径选项。接下来, 从这些备选子路径中选择剩余带宽资源最多且所需带宽占用后该链路带宽利用不会超过阈值、时延不会超过请求时延要求的新路径替换, 若在所有备选子路径中均无法找到满足条件的路径, 则替换失败。

算法 4 链路替换算法

输入: 请求 f_m 的原路径方案 P_m

输出: 调整后的路径方案 π_m

```

1. for  $L_{ij}$  in  $P_m$ 
2.     查找节点  $i, j$  之间满足带宽需求的子路径, 并将所有满足条件的备选子路径根据路径上的剩余带宽升序排序得到备选集合  $optional\_link$ 
3.     if  $optional\_link$  为空
4.         break
5.     else
6.         for  $link$  in  $optional\_link$ 
7.             if 在  $link$  上分配原链路的流量需求,  $link$  的带宽利用率满足且不超时 then
8.                 替换原来节点  $i, j$  间的链路为  $link$ 
9.             else
10.                从备选集合中移除该条子路径
11.            end for
12.        end if
13.    end for
14. if 所有  $L_{ij}$  in  $P_m$  的备选子路径集合都为空
15.     return 奖励, 替换失败, 新的状态,  $\pi_m$ 
16. else
17.     return 奖励, 替换成功, 新的状态,  $\pi_m$ 
    
```

4.3.2 复杂性分析

算法 1 的时间复杂度为 $O(|F| \cdot |V| \cdot |E| \cdot f_{max})$, f_{max} 为请求的平均最大流, 算法 3 的时间复杂度为 $O(|V|)$, 算法 4 的时间复杂度为 $O(p \cdot |V| \cdot \log V)$, p 为路径的平均链路数, 通过上述 3 个算法时间复杂度分析, 得出本章所提的基于最小费用最大流的流转发路径动态调整算法的时间复杂度为 $O(|F| \cdot |V| \cdot |E| \cdot f_{max}) + O(num_episode \cdot |V|^2 \cdot p \cdot \log V)$, 其中 $num_episode$ 为训练次数。

4.4 仿真设计与性能分析

本节将详细阐述使用 Pytorch 框架进行的仿真实验，以评估本章所提出的基于 MCMF-DRL 协同的流转发路径调整算法的性能。首先介绍了本章用于评估算法的仿真设置，包含网络拓扑结构、请求类型以及请求的带宽、时延设置。随后，通过对比两种不同算法，分析本章算法在负载均衡系数、转发成本等方面的性能优势，直观地展示本章所提算法在负载均衡方面的有效性。

4.4.1 实验环境与参数设置

本章实验中的物理网络来自由 ROCKETFUEL 收集的另一个 ISP 网络拓扑 Sprintlink^[29]，该拓扑共有 44 个节点，166 条链路。对于 ROCKETUEL 拓扑，链路的传输成本信息是明确给出的，但并未提供链路的带宽容量数据。为合理设定链路容量，本文参考了思科路由器的默认配置方式，即链路成本与其容量呈反比关系，据此将链路容量设置为其成本的倒数。这种设置方式已被多项前沿研究采用，具有一定的现实依据与可比性^[29]。本章中的请求由第三章工作中设计的请求生成器给出，带宽利用率阈值根据文献[29]设置。本章算法中使用 Pytorch 框架构建深度强化学习中的策略网络，网络结构为三层全连接网络。对于激活函数的选择，相较于 Sigmoid 函数，线性整流单元（Rectified Linear Unit, ReLU）的梯度计算更加稳定，可有效避免梯度消失问题，从而提升模型的收敛速度和训练效率。此外，ReLU 能更高效地提取数据中的非线性特征，具备更强的表达能力。因此，本章选择 ReLU 作为神经元激活函数。针对优化器的选择，Adam 优化器在处理复杂损失函数时表现出更快的收敛速度和较少的超参数调整需求，因此本章算法选择 Adam 作为模型优化器。相关参数设置如表 4-2 实验参数设置所示：

表 4-2 实验参数设置

参数	值
请求类型	视频、游戏、文件、VoIP
请求带宽要求	10Mbps~120Mbps
请求时延要求	40ms~400ms
惩罚参数 σ	10000
初始学习率	0.0001
初始探索率	0.9
带宽利用率阈值	0.95
更新步长	200

4.4.2 对比算法介绍

为了全面评估本章算法的性能,验证其在解决请求路径动态调整问题中的优越性,本文与以下两种方法进行比较:

RILNET (ReInforcement Learning NETworking): 该方法是一种 SDN 架构下基于强化学习的负载均衡方法,采用深度确定性策略梯度算法,根据控制器收集的流量数据、网络配置和路由控制信息来学习隐藏的拥塞模式从而优化路由策略,在最大链路数据丢失、最大链路延迟方面优于传统的 ECMP 和 DRILL 方法^[61]。

ALBLP (Adaptive Load-Balancing Architecture Based on Link-State Prediction): 该方法是基于链路状态预测的自适应负载均衡方法。使用 LSTM 模型预测链路状态,并将预测的链路状态值作为 Dijkstra 算法的权重来计算路径,并主动下发流表,一定程度上缓解了控制器和交换机之间传输时延对负载均衡的影响^[62]。

4.4.3 评价指标

本章中,将从多个维度对算法性能进行评估,包括负载均衡程度,转发成本,和时延。

4.4.4 实验结果与分析

负载均衡系数 k 的比较:如图 4-5 所示,本章算法的负载均衡系数平均低于 RILNET 算法和 ALBLP 算法 34.2%和 43.3%。由于 ALBLP 算法中的路径计算采用的是 Dijkstra 算法,对链路负载的考虑不足,容易导致多个路径共用的最短路径通路的负载增加,而其他通路上的负载较少,导致链路利用率不均衡。而在 RILNET 算法中,虽然在强化学习的奖励设置中定义了拥塞感知的奖励,智能体在学习时考虑了链路利用率,但该方法在离散动作空间下效果受限。

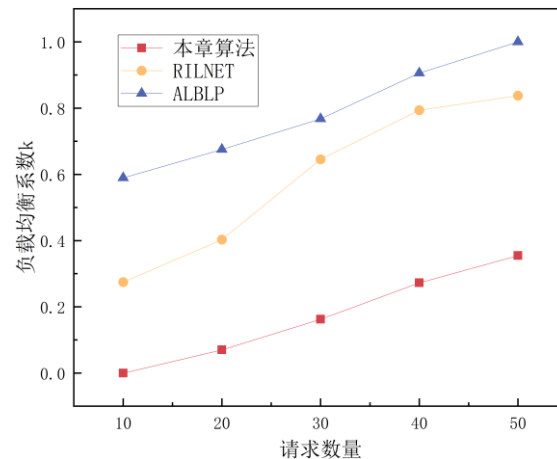


图 4-5 负载均衡系数 k 的对比

转发成本的比较：如图 4-6 所示，本章算法在转发成本上略高于 ALBLP 算法 5.29%，这表明在优化链路带宽利用的均衡性时，需要在一定程度上牺牲路径成本。即为了实现更好的负载均衡，系统会选择绕行路径方案，而非始终采用最短路径，这种权衡关系也反映了负载均衡和转发成本之间的矛盾。

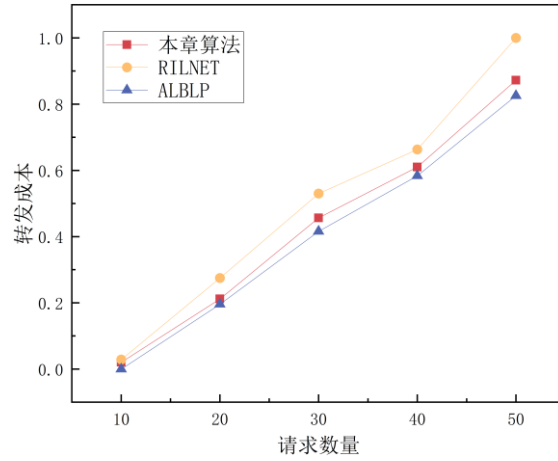


图 4-6 转发成本的对比

优化目标对比：本章优化目标即最小化负载均衡系数和转发成本之和，如图 4-7 所示，本章算法能够在带宽限制、时延约束、流请求动态变化的多约束条件下实现更优的优化目标，负载均衡系数和转发成本之和分别低于 RILNET 算法和 ALBLP 算法 44.4%和 49.1%。表明本章算法能够根据网络的动态变化自适应调整路径，在不同规模请求的场景下保持稳定的优化能力，虽然在成本上略有牺牲，但可以得到更好的链路均衡效果，实现负载均衡与转发成本之间的良好权衡。

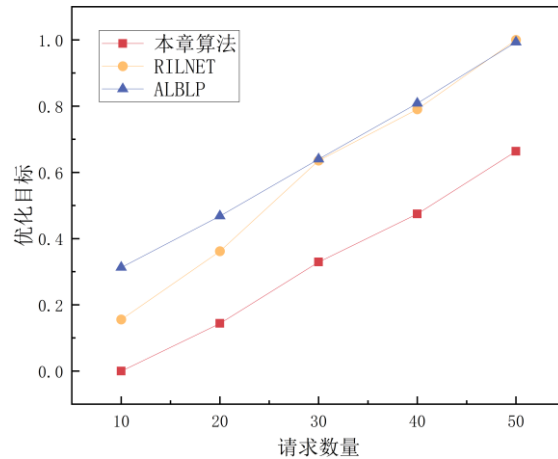


图 4-7 优化总目标的对比

时延的对比：本章实验中的时延指的是学习请求路径的时间，即算法在确定合适的转发路径时所需的计算和决策时间。如图 4-8 所示，本章算法的学习请求路径时间小于 RILNET，但高于 ALBLP。ALBLP 通过预测链路状态提前下发流表的策略，使得在处理流量请求时无需过多计算路径，从而缩短了路径学习时间。相比之下，本章算法虽然在计算路径时仍然需要一定的学习过程，但通过 MCMF

为强化学习提供了合理的学习方向，有效减少了搜索空间，提高了路径选择的效率。因此相较于 RILNET,本章算能更快地收敛到较优路径，减少学习请求路径的时间。

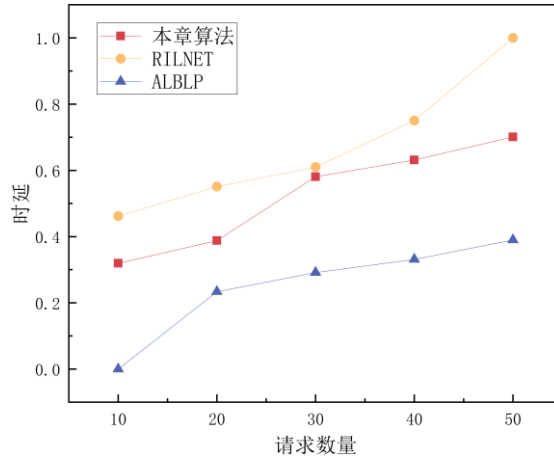


图 4-8 时延的对比

4.5 本章小结

面对大规模流量并发的网络环境时，逐点路径探索策略在调度效率与全局协调能力方面存在明显瓶颈。为解决该问题，本章围绕多请求并发调度场景，提出了一种基于最小成本最大流与深度强化学习协同优化的路径动态调整算法，以实现更加高效且全局感知的资源分配机制。

本章方法通过构建全局路径规划与局部策略调整相结合的双阶段优化框架，充分发挥 MCMF 在转发成本控制方面的优势，并利用 DRL 的在线学习能力动态适应网络状态变化，进一步提升链路利用率与负载均衡水平。算法在路径生成阶段融入多媒体请求的 QoS 权重，实现差异化调度；在路径调整阶段综合考虑链路负载与时延，提升全局调度性能。通过与两个不同算法对比，本章算法在最小化负载均衡和转发成本方面效果更优。

5 总结和展望

5.1 总结

随着现代网络应用的迅猛发展，网络流量需求持续增长，使得传统网络架构在负载均衡、路径优化和资源调度方面面临严峻挑战。SDN 作为一种新型的网络架构，以其集中控制和灵活管理的特点，在提升网络性能方面展现出了巨大潜力。然而，随着网络规模的不断扩大，SDN 在带宽分配、流量调度和路径选择方面仍然存在诸多问题，其中尤为突出的是负载不均衡导致的链路拥塞问题。针对这一挑战，本文研究了 SDN 架构下基于深度强化学习的网络负载均衡算法，以有效提升网络的资源利用率、降低链路负载不均衡程度，并改善整体用户体验。

1、提出了一种基于模糊层次分析的流转发路径探索算法，以优化 SDN 在多媒体服务场景中，报文流请求转发过程存在的部分请求因优先级划分不合理而导致网络链路拥塞，以及链路负载均衡和转发成本难以权衡的问题。首先，利用模糊层次分析法计算不同请求的 QoS 需求权重，确定请求的优先级。其次，按照优先级顺序进行路径探索，并引入深度 Q 学习进行节点选择，以权衡链路负载均衡和转发成本。实验结果表明，与 QLLB 和 HiQoS 算法相比，所提算法在负载均衡系数和转发成本的最小化目标上具有更好效果。

2、提出了一种基于最小成本最大流与深度强化学习协同优化的流转发路径动态调整算法，以优化多请求并发调度场景下，路径逐点探索策略难以实现全局资源协调的问题，进一步提升链路利用率与调度效率。首先，通过最小成本最大流模型从全局角度为所有请求生成初始转发路径，以最小化转发成本。随后，将路径结果作为输入，引导深度强化学习根据实时网络状态动态调整路径，提升整体负载均衡水平。

5.2 展望

本文以深度强化学习作为研究出发点，致力于优化 SDN 架构下网络负载均衡问题，并通过实验验证了本文所提的方法在流量调度和负载均衡方面取得了一定的优化效果，但仍然存在一些值得进一步研究的方向。未来的研究可以在以下几个方面展开，以进一步提升 SDN 在复杂网络环境中的适应性和性能：

1、增强泛化能力与实时适应性：本研究的强化学习模型主要基于固定的网络拓扑进行训练，但在实际应用中，网络拓扑往往是动态变化的。因此，当前模

型可能无法很好地适应不同的网络环境，且在面对未知或变化的网络拓扑时，性能可能会显著下降。为了提高模型的泛化能力，未来可以引入自适应强化学习方法，使得模型能够在面对未知网络环境时快速进行调整和优化。例如，利用在线学习或增量学习技术，使模型能够在实时环境中不断更新和学习，从而在不同的网络拓扑和流量模式下保持较好的性能。此外，迁移学习技术也可作为重要的研究方向，将在特定网络拓扑中训练得到的模型迁移到其他相似的网络环境中。这不仅可以有效减少对大规模网络环境的重新训练时间，还能显著降低计算成本和资源消耗，提高模型的实用性和部署效率。通过迁移学习，模型能够借鉴已有的网络拓扑经验，并加速在新环境中的适应过程，从而在多样化和动态变化的网络环境中表现得更加稳健与高效。

2、分布式学习架构的引入：目前的强化学习模型采用集中式训练方式，虽通过在单一控制器上进行大规模的训练，优化网络的流量调度和管理。然而，这种方法在面对大规模网络环境时，可能会遇到计算资源瓶颈，导致训练效率低下，且难以满足网络规模不断扩展的需求。为了克服这一限制，未来可以探索基于联邦学习或分布式强化学习的方案，这些方法允许多个控制器共同协作，通过在不同节点上并行训练流量调度模型，从而提高计算效率、加快模型训练速度，并有效提高系统的可扩展性。通过分布式架构，网络中的每个控制器可以在本地处理和优化流量调度任务，并定期共享或合并训练结果，以获得全局最优解。这不仅能够减少对集中式计算资源的依赖，还能在面对网络拓扑变化时，更快速地进行局部调整和优化。

此外，基于边缘计算的强化学习方法也是未来的一个重要发展方向。在传统的集中式训练方法中，所有计算和决策都依赖于核心控制器，这不仅增加了延迟，也可能导致计算瓶颈。通过将计算任务分布到网络边缘设备，可以在靠近数据源的地方进行决策和优化，减少数据传输的延时，并有效降低核心控制器的负载。这种边缘计算模式下的强化学习不仅能够实现更加实时的网络决策，还能提升系统的自适应能力，快速响应网络环境中的变化，提高网络的鲁棒性和灵活性。通过结合边缘计算和分布式学习架构，未来的网络管理系统将能够更加高效地处理大规模流量调度问题，且能在复杂多变的网络环境中快速适应和优化。

参考文献

- [1] Neghabi A A, Navimipour N J, Hosseinzadeh M, et al. Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature[J]. IEEE access, 2018, 6: 14159-14178.
- [2] Mousa A K, Abdullah M N. A Survey on Load Balancing, Routing, and Congestion in SDN[J]. Engineering and Technology Journal, 2022, 40(10): 1284-1294.
- [3] Chen Y T, Li C Y, Wang K. A fast converging mechanism for load balancing among sdn multiple controllers[C]//2018 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2018: 00682-00687.
- [4] Attarha S, Hosseiny K H, Mirjalily G, et al. A load balanced congestion aware routing mechanism for Software Defined Networks[C]//2017 Iranian Conference on Electrical Engineering (ICEE). IEEE, 2017: 2206-2210.
- [5] Pietrabissa A, Celsi L R, Cimorelli F, et al. Lyapunov-based design of a distributed wardrop load-balancing algorithm with application to software-defined networking[J]. IEEE Transactions on Control Systems Technology, 2018, 27(5): 1924-1936.
- [6] Song B, Chang Y, Zhang X, et al. Mixed-Flow Load-Balanced Scheduling for Software-Defined Networks in Intelligent Video Surveillance Cloud Data Center[J]. Applied Sciences, 2022, 12(13): 6475.
- [7] Song P, Liu Y, Liu T, et al. Flow Stealer: lightweight load balancing by stealing flows in distributed SDN controllers[J]. Science China. Information Sciences, 2017, 60(3): 032202.
- [8] Kumar P, Kumar R. Issues and challenges of load balancing techniques in cloud computing: A survey[J]. ACM computing surveys (CSUR), 2019, 51(6): 1-35.
- [9] 陈中卿, 李丹丹, 闪德胜, 等. 基于流特征的数据中心非对称流负载均衡方法[J]. 软件学报, 2022, 34(8): 3924-3937.
- [10] Joshi S, Kumari U. Load balancing in cloud computing: Challenges & issues[C]//2016 2nd international conference on contemporary computing and informatics (IC3I). IEEE, 2016: 120-125.
- [11] Mendelson G. A lower bound on the stability region of redundancy-d with FIFO service discipline[J]. Operations Research Letters, 2021, 49(1): 113-120.
- [12] Rathi R, Narkhede N, Hasamnis M A. Design and Implementation of First-In-First-Out (FIFO) Buffer for Distributed Load Balancing Systems[C]//International Conference on Smart Computing and Communication. Singapore: Springer Nature Singapore, 2024: 11-21.

- [13] Omer Y A H, Mustafa A B A, Abdalla A G. Performance analysis of round robin load balancing in sdn[C]//2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCEEE). IEEE, 2021: 1-5.
- [14] Garcia-Carballeira F, Calderon A, Carretero J. Enhancing the power of two choices load balancing algorithm using round robin policy[J]. Cluster Computing, 2021, 24(2): 611-624.
- [15] Harjanti T W, Setiyani H, Trianto J. Load balancing analysis using round-robin and least-connection algorithms for Server Service Response Time[J]. Applied Technology and Computing Science Journal, 2022, 5(2): 119-128.
- [16] Ranjithkumar S, Sellamuthu K, Rajkumar R S, et al. Certain Investigation on Load Balancing using Cloudlet Assignment and minmax Algorithm[J]. Annals of the Romanian Society for Cell Biology, 2021, 25(3): 2223-2229.
- [17] Kim M H, Lee J Y, Raza Shah S A, et al. Min-max exclusive virtual machine placement in cloud computing for scientific data environment[J]. Journal of Cloud Computing, 2021, 10: 1-17.
- [18] Chauhan S, Soni S, Kumar A, et al. Load Balancing Algorithms for Cloud Computing Performance: A Review[C]//International Conference on Cognitive Computing and Cyber Physical Systems. Singapore: Springer Nature Singapore, 2023: 159-176.
- [19] Kaur S, Sharma R, Kalsi P, et al. Load Balancing Algorithms for Cloud Computing Performance: A Review[C]//The fifth International Conference on Computing, Communications, and Cyber-Security: IC4S'05 Volume 1. Springer Nature, 2024, 1: 159.
- [20] 胡逸飞, 包梓群, 包晓安. 一种基于改进遗传算法的动静态负载均衡算法[J]. Electronic Science & Technology, 2023, 36(9).
- [21] Ahmad R W, Gani A, Hamid S H A, et al. A survey on virtual machine migration and server consolidation frameworks for cloud data centers[J]. Journal of network and computer applications, 2015, 52: 11-25.
- [22] Bibal Benifa J V, Dejeu. Performance improvement of Mapreduce for heterogeneous clusters based on efficient locality and replica aware scheduling (ELRAS) strategy[J]. Wireless Personal Communications, 2017, 95: 2709-2733.
- [23] Shivakeshi C, Sreepathi B. Software defined network based enhanced energy-aware load balancing routing protocol[J]. Electric Power Components and Systems, 2024: 1-17.
- [24] Ragmani A, Elomri A, Abghour N, et al. An improved hybrid fuzzy-ant colony algorithm applied to load balancing in cloud computing environment[J]. Procedia Computer Science, 2019, 151: 519-526.
- [25] Aloboud E. Cuckoo-inspired Job Scheduling Algorithm for Cloud Computing International Workshop on Peer-to-Peer Architectures[J]. Networks, and Systems, 2019.

-
- [26] Pradhan A, Bisoy S K, Kautish S, et al. Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment[J]. IEEE Access, 2022, 10: 76939-76952.
- [27] Gures E, Shayea I, Ergen M, et al. Machine learning-based load balancing algorithms in future heterogeneous networks: A survey[J]. IEEE Access, 2022, 10: 37689-37717.
- [28] Sanchez L P A, Shen Y, Guo M. DQS: A QoS-driven routing optimization approach in SDN using deep reinforcement learning[J]. Journal of Parallel and Distributed Computing, 2024, 188: 104851.
- [29] 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.
- [30] Rezaei S, Liu X. Multitask learning for network traffic classification[C]//2020 29th International Conference on Computer Communications and Networks (ICCCN). IEEE, 2020: 1-9.
- [31] Raikar M M, Meena S M, Mulla M M, et al. Data traffic classification in software defined networks (SDN) using supervised-learning[J]. Procedia Computer Science, 2020, 171: 2750-2759.
- [32] Samal T, Kabat M R. A prioritized traffic scheduling with load balancing in wireless body area networks[J]. Journal of King Saud university-computer and information sciences, 2022, 34(8): 5448-5455.
- [33] Ke Y, Wang J, Yan C, et al. Routing strategy for SDN large flow based on deep reinforcement learning[C]//2022 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom). IEEE, 2022: 523-530.
- [34] Liu C, Xu M, Yang Y, et al. DRL-OR: Deep reinforcement learning-based online routing for multi-type service requirements[C]//IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE, 2021: 1-10.
- [35] Ruelas A M R, Rothenberg C E. A load balancing method based on artificial neural networks for knowledge-defined data center networking[C]//The 10th Latin America Networking Conference. 2018: 106-109.
- [36] Ejaz S, Iqbal Z, Shah P A, et al. Traffic load balancing using software defined networking (SDN) controller as virtualized network function[J]. IEEE Access, 2019, 7: 46646-46658.
- [37] Ghomi E J, Rahmani A M, Qader N N. Load-balancing algorithms in cloud computing: A survey[J]. Journal of Network and Computer Applications, 2017, 88: 50-71.
- [38] Oliveira A T, Martins B J C A, Moreno M F, et al. SDN - based architecture for providing quality of service to high - performance distributed applications[J]. International Journal of

- Network Management, 2021, 31(5): e2078.
- [39] Mehraban S, Yadav R K. Traffic engineering and quality of service in hybrid software defined networks[J]. China Communications, 2024, 21(2): 96-121.
- [40] Beshley M, Kryvinska N, Beshley H, et al. Traffic engineering and QoS/QoE supporting techniques for emerging service-oriented software-defined network[J]. Journal of Communications and Networks, 2024, 26(1): 99-114.
- [41] Pei X, Sun P, Hu Y, et al. Enabling efficient routing for traffic engineering in SDN with Deep Reinforcement Learning[J]. Computer Networks, 2024, 241: 110220.
- [42] Alam I, Sharif K, Li F, et al. A survey of network virtualization techniques for Internet of Things using SDN and NFV[J]. ACM Computing Surveys (CSUR), 2020, 53(2): 1-40.
- [43] Kim Y, Gil J M, Kim D. A location - aware network virtualization and reconfiguration for 5G core network based on SDN and NFV[J]. International Journal of Communication Systems, 2021, 34(2): e4160.
- [44] Ali T E, Chong Y W, Manickam S. Machine learning techniques to detect a DDoS attack in SDN: A systematic review[J]. Applied Sciences, 2023, 13(5): 3183.
- [45] Hnamte V, Najar A A, Nhung-Nguyen H, et al. DDoS attack detection and mitigation using deep neural network in SDN environment[J]. Computers & Security, 2024, 138: 103661.
- [46] Nguyen T G, Phan T V, Hoang D T, et al. Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks[J]. IEEE Transactions on Cognitive Communications and Networking, 2021, 7(4): 1048-1065.
- [47] Karakus M, Durresi A. Quality of service (QoS) in software defined networking (SDN): A survey[J]. Journal of Network and Computer Applications, 2017, 80: 200-218.
- [48] Keshari S K, Kansal V, Kumar S. A systematic review of quality of services (QoS) in software defined networking (SDN)[J]. Wireless Personal Communications, 2021, 116(3): 2593-2614.
- [49] Sharifani K, Amini M. Machine learning and deep learning: A review of methods and applications[J]. World Information Technology and Engineering Journal, 2023, 10(07): 3897-3904.
- [50] Taye M M. Understanding of machine learning with deep learning: architectures, workflow, applications and future directions[J]. Computers, 2023, 12(5): 91.
- [51] Vignon C, Rabault J, Vinuesa R. Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions[J]. Physics of fluids, 2023, 35(3).
- [52] 甘应爱. 运筹学[M]. 清华大学出版社有限公司, 2005.
- [53] 孟萍. 软件定义网络中的负载均衡路由算法研究[D]. 南京邮电大学, 2022.
- [54] 王炜发, 张大明, 刘堃铃, 等. 软件定义网络中基于Q-学习的负载均衡算法[J]. 电讯技术, 2021, 61(09):1066-1072.

- [55] Huang F, Zhang J, Xu J, et al. An SDN-based QoS guaranteed mechanism for geospatial flows[C]//2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, 2019: 1394-1401.
- [56] Jain S, Jain V K, Mishra S. Fuzzy-AHP model for Intelligent network selection in VANET for data offloading[C]//2023 IEEE 7th Conference on Information and Communication Technology (CICT). IEEE, 2023: 1-5.
- [57] 刘楚仪, 万剑雄, 李文静. 基于深度强化学习的单集群云资源管理算[J]. 内蒙古工业大学学报(自然科学版), 2023, 42(05):459-466.
- [58] Ye Junhong, Zhang Yingjun Angela, Deep reinforcement learning based base station activation in heterogeneous networks[J]. IEEE Transactions on Mobile Computing:2019, 19(9): 2076-2087.
- [59] Padhy S, Chou J. Reconfiguration aware orchestration for network function virtualization with time-varied workload in virtualized datacenters[J]. IEEE Access:2021, 9: 48413-48428.
- [60] Kamboj P, Pal S, Bera S, et al. QoS-aware multipath routing in software-defined networks[J]. IEEE Transactions on Network Science and Engineering:2022, 10(2): 723-732.
- [61] Lin Q, Gong Z, Wang Q, et al. Rilnet: A reinforcement learning based load balancing approach for datacenter networks[C]//Machine Learning for Networking: First International Conference, Springer International Publishing, 2019: 44-55.
- [62] Chen J, Wang Y, Huang X, et al. ALBLP: Adaptive Load - Balancing Architecture Based on Link - State Prediction in Software - Defined Networking[J]. Wireless Communications and Mobile Computing, 2022, 2022(1): 8354150.

攻读学位期间的学术论文与研究成果

- [1] 邱彩钰, 舒兆港, 陈淑武, 等. 基于模糊层次分析和 DQN 的流转发路径探索算法[J]. 计算机工程与设计, 2025. (第一作者, 已录用)
- [2] Tian J, Shu Z, Chen S, et al. DDoS Defense Strategy Based on Blockchain and Unsupervised Learning Techniques in SDN[J]. IEEE Transactions on Network and Service Management, 2024. (第六作者, 在投)
- [3] Tian J, Shu Z, Chen S, et al. Enhanced DDoS Defense in SDN: Double-Layered Strategy with Blockchain Integration[C]//2024 13th International Conference on Communications, Circuits and Systems (ICCCAS). IEEE, 2024: 380-384. (第六作者)
- [4] 王媛滔, 舒兆港, 钟一文, 等. 基于 VNF 实例共享的服务功能链部署算法[J]. 计算机应用研究, 2023, 40(6). (第四作者)

致谢

转眼走到了，学生时代最终章。回望这段充满挑战与成长的旅程，我要向所有给予我支持和帮助的人表达最诚挚的谢意。

首先，衷心感谢我的导师舒兆港副教授。在这段求知之路上，不仅给予了我悉心的指导，还教会了如何独立思考、勇敢探索，也为我提供了良好的学习环境和研究资源。

我还要感谢我的家人和朋友们。你们一直是我坚强的后盾，无论何时，总能给予我温暖和力量，让我勇敢面对挑战。

其次，感谢实验室的师兄师姐、同门、师弟们。研究的过程或许艰辛，但有你们的陪伴，这段时光也变得弥足珍贵。

最后，感谢自己没有放弃。

毕业不是终点，而是新的起点。未来的路或许充满未知，但我会带着这段旅程的收获，继续勇敢前行。期待新的旅程精彩万分。