

基于模糊层次分析和DQN的流转发路径探索算法

邱彩钰, 舒兆港⁺, 陈淑武, 谢海辉, 柳晓龙, 田佳霖

(福建农林大学 计算机与信息学院, 福建 福州 350002)

摘要: 基于软件定义网络在多媒体流服务场景下, 流请求转发过程中部分请求因优先级低导致链路拥塞, 及链路负载均衡和转发成本难以权衡的问题, 提出了基于模糊层次分析和DQN的流转发路径探索算法。算法通过模糊数计算请求的QoS权重和优先级; DQN依据优先级顺序、高负载均衡程度、低转发成本原则探索路径, 生成转发策略及相应负载均衡系数和成本。仿真实验结果表明, 该算法相较QLLB和HiQoS算法, 负载均衡系数和转发成本之和分别降低6.49%和9.74%。

关键词: 软件定义网络; 多媒体流; 链路负载均衡; 转发路径; 成本优化; 请求调度; 深度强化学习

中图法分类号: TP393 **文献标识码:** A **文章编号:** 1000-7024 (2026) 01-0105-08

doi: 10.16208/j.issn1000-7024.2026.01.013

Flow forwarding path exploration algorithm based on fuzzy analytic hierarchy process and DQN

QIU Cai-yu, SHU Zhao-gang⁺, CHEN Shu-wu, XIE Hai-hui, LIU Xiao-long, TIAN Jia-lin

(College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China)

Abstract: In multimedia streaming service scenarios based on Software-Defined Networking (SDN), the flow request forwarding process faces issues where some requests with low priority cause link congestion, and it is difficult to balance link load distribution and forwarding costs. To address these challenges, a flow forwarding path exploration algorithm based on fuzzy analytic hierarchy process (FAHP) and Deep Q-Network (DQN) is proposed. The QoS weights and priorities of requests were calculated through fuzzy numbers by the algorithm. Based on the priority order, high load balancing degree, and low forwarding cost principles, paths were explored by the DQN, and forwarding strategies along with corresponding load balancing coefficients and costs were generated. Simulation experiments show that compared to QLLB and HiQoS algorithms, the proposed algorithm reduces the sum of load balancing coefficient and forwarding cost by 6.49% and 9.74%, respectively.

Key words: software defined networking; multimedia streaming; link load balancing; forwarding path; cost optimization; request scheduling; deep reinforcement learning

0 引言

在多媒体应用中, 报文流请求转发过程存在部分请求因优先级不足而导致网络链路拥塞, 以及链路负载均衡和链路转发成本难以权衡的问题。传统路由机制中嵌入的拥

塞避免和控制机制在不同优先级的多流共存的场景中难以迅速适应^[1]。SDN虽已成为一种有效的解决方案, 但也面临着带宽分配与流量路径选择的挑战。

当前SDN负载均衡研究可总结为基于控制的负载均衡和基于数据的负载均衡, 且多为动态负载均衡。得益于人

收稿日期: 2024-09-19; **修订日期:** 2025-12-22

基金项目: 福建省高校产学研合作基金项目 (2024H6007、2024H6030); 福建省工业和信息化厅重点攻关基金项目 (闽工信函软件【2024】488号); 网络通信公司产业研究基金项目 (KH230139A、KH240131A)

作者简介: 邱彩钰 (2000-), 女, 福建莆田人, 硕士研究生, 研究方向为SDN链路负载均衡; 通讯作者: 舒兆港 (1981-), 男, 江西上饶人, 博士, 副教授, 研究方向为软件定义网络与网络功能虚拟化、机器学习与智能化网络体系结构、网络切片、云计算与边缘计算等; 陈淑武 (1976-), 男, 福建宁德人, 硕士, 教授, 研究方向为物联网技术; 谢海辉 (1991-), 男, 福建泉州人, 博士, 讲师, 研究方向为边缘计算和人工智能算法; 柳晓龙 (1989-), 男, 福建宁德人, 博士, 副教授, 研究方向为智能系统安全、深度学习中的对抗性示例和区块链; 田佳霖 (1997-), 男, 山东淄博人, 硕士研究生, 研究方向为软件定义网络 (SDN) 和DDoS防御策略。

E-mail: zgshu@fafu.edu.cn

工智能领域的快速发展,在SDN中实施人工智能方法以提高网络资源利用率和整体性能正是当前研究的热点^[2]。文献[3-7]使用多智能体深度强化学习、群智能算法等优化路由策略,针对控制层面提出优化,为SDN控制器制定路由策略,但它们在共享有限资源时平等对待所有数据流,因而无法保证每个应用程序的QoS需求。文献[8-12]采用深度强化学习(deep reinforcement learning, DRL)代理、流量分类与预测等,通过研究网络流量,以实现网络资源的最佳利用,这些研究仅针对数据流的处理展开,却弱化了路由策略的考虑。因此,综合考虑控制层面与数据流层面的同步优化是一个可行的方案,本文提出基于模糊层次分析法(FAHP)和深度Q网络(DQN)的流转发路径探索算法,以应对以上挑战。本文的主要贡献有:

(1) 流转发路径探索算法综合考虑不同请求的需求,结合分类的方法对请求的优先级进行细粒度排序,为优化流量路径奠定基础;

(2) 用带宽利用率的标准差表征负载均衡系数,采用整数规划模型,同时综合了数据层面和控制层面的优化,

有效降低了请求的迟到率,缓解了多媒体流服务场景中请求因优先级不足而导致网络链路拥塞的问题;

(3) 将优化问题转化为马尔可夫决策过程(Markov decision process, MDP),提出了综合考虑负载均衡系数和转发成本的流转发路径探索算法。算法利用深度强化学习算法DQN进行节点探索,基于链路负载均衡系数和转发成本调整当前路径,以达到负载均衡系数和转发成本的平衡。

1 系统模型

本文中SDN控制器内部被划分为链路感知模块、强化学习模块和流表下发模块^[13],如图1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;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的交换机。交换机转发端口根据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。

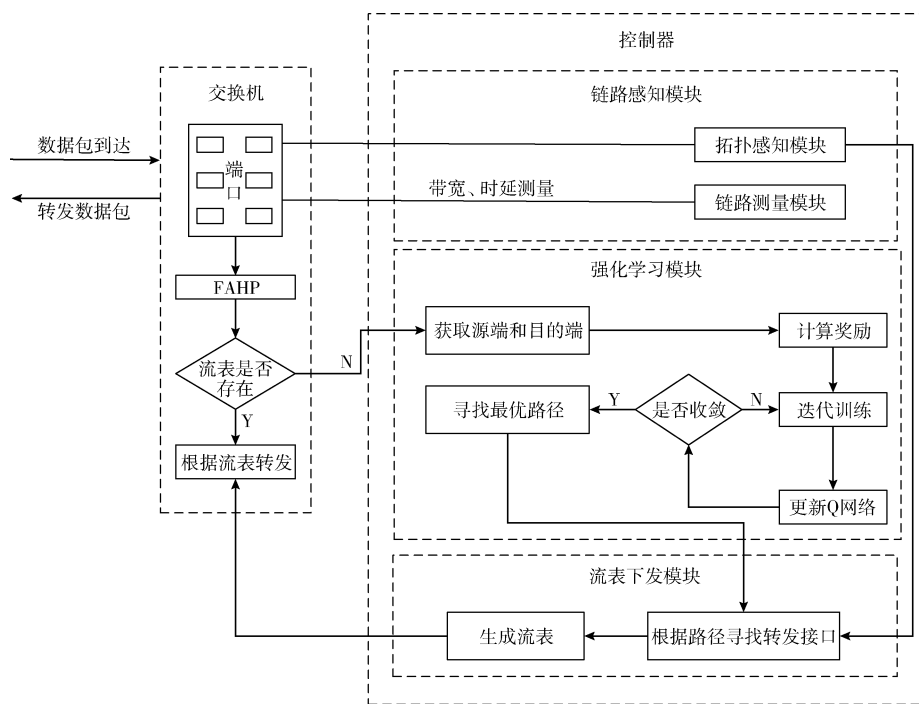


图1 系统处理模型框架

本文使用无向图 $G(V, E)$ 表示图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。本文涉及的主要参数和意义见表1。

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(1)

表 1 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}	链路 (i, j) 上的带宽
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

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

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 (2) 和条件 (3)

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

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

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 , 如式 (4) 所示, 用以衡量整个网络的负载均衡程度

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

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, 转发成本如式 (5) 所示

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

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

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

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

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

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

α 是网络负载均衡系数和转发成本的权重调节因子。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和 DQN 的流转发路径探索算法, 下一节将详细介绍该算法设计原理。

2 流转发路径探索算法

2.1 算法描述

流转发路径探索算法旨在综合考虑请求的多个 QoS 需求, 使计算请求优先级的过程更加灵活, 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。

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。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。

建立递阶层次结构。如图 2 所示为 FAHP 的层次结构^[14], 本文算法中目标层是所有请求的综合优先级, 决策层本文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。

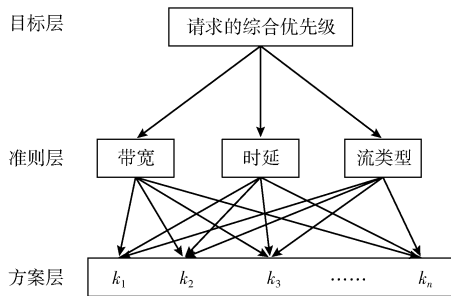


图 2 FAHP 层次结构

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

模糊一致判断矩阵 R

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 $r_{ij} + r_{ji} = 1$; 标度及其含义见表 2^[15]。

表 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_j 与元素 a_i 相比较得到的为 $r_{ji} = 1 - r_{ij}$

由判断矩阵计算该指标的权重。权重的计算公式为:

$$\omega_i = \frac{\sum_{j=1}^n a_{ij} + \frac{n}{2} - 1}{n(n-1)}, \text{ 其中 } n \text{ 为矩阵的阶数。}$$

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 3 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 3 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) ~ (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 3 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 $\text{bandwidth_fuzzy_matrix}$

(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 $\text{bandwidth_fuzzy_matrix}$ 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 $\text{delay_fuzzy_matrix}$

(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 $\text{delay_fuzzy_matrix}$ 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 $\text{index_weights}, \text{bandwidth_weights}, \text{delay_weights}, \text{type_weights}$ 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 (4) 和目标 (5) 探索下一步转发节点, 由此引入强化学习算法。在强化学习中, 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 (DRL) 来解决^[16]。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^[17]。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^[18]。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。所以本文将问题求解过程被建模为 MDP 并使用 DQN 进行求解。算法主流程如图 3 所示, 包含算法中强化学习模块的关键步骤: ① 根据当前状态 S_t 选择动作 a_t 并计算采取这个动作的奖励 r , 以及到达的下一个状态 S_{t+1} ; ② 经验回放; ③ 更新两个 Q 网络。

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 , $b(t) = [b_{00}(t), b_{01}(t), \dots, b_{ij}(t)]$, $i, j \in [0, |V| - 1]$ 表示 t 时刻网络的带宽矩阵, $b_{ij}(t)$ 表示 t 时刻

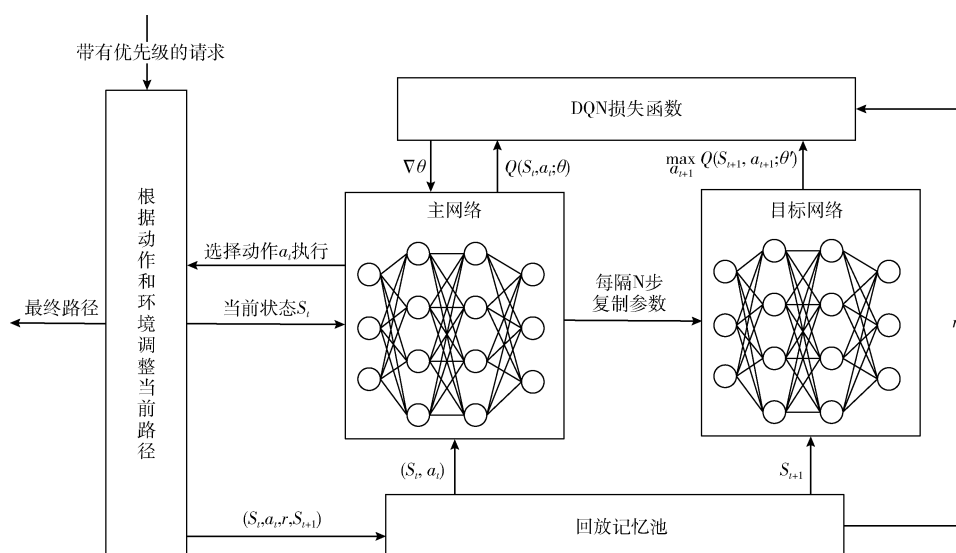


图 3 流转发路径探索算法主流程

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。本文算法使用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_t 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, 如式 (7) 所示

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

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 $path[-1] \neq end$

(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 在选择动作后执行, 选择的动作即下一节点。

由于DQN根据Q值选择动作, 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, 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) 使用式 (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  $path$ 
```

2.2 算法复杂度分析

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 仿真与性能分析

本节对所提出算法进行仿真和性能分析, 并从负载均衡系数、转发成本角度与已有的QLLB算法^[13]和HiQoS算法^[19]进行比较分析。

3.1 实验环境和参数设置

本文采用由ROCKETFUEL收集的拓扑Ebone^[9], 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[9]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[20]设置, 同时根据文献[20]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, 其余参数见表3。

表3 实验主要参数和值

参数	值
请求类型	视频、游戏、文件、VoIP
请求带宽要求	10 Mbps~120 Mbps
请求时延要求	40 ms~400 ms
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
目标网络更新周期	200

在调用FAHP算法时, 对于同一请求中的3个QoS指标, 本文考虑的模糊矩阵标度从高到低为: 时延、流类型、带宽, 计算得到3个指标的权重向量 $\omega = (0.383, 0.334, 0.283)^T$; 对于不同请求在同一指标上的比较, 本文考虑的模糊矩阵标度从高到低为: 游戏、视频、文件、VoIP。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见表4。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。

表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; $S_4:5; T_4:14; d_4:223;$ $b_4:61; t_4:文件$

3.2 实验结果与分析

为了评估本文算法性能以及和已有的QLLB算法和HiQoS算法进行对比, 我们对负载均衡系数和转发成本数据进行了归一化处理, 且数据值在 $[0, 1]$ 之间。主要对比的数据包括请求迟到数、负载均衡系数、转发成本等。

请求迟到数对比。如图4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不足而导致网络拥堵的问题。

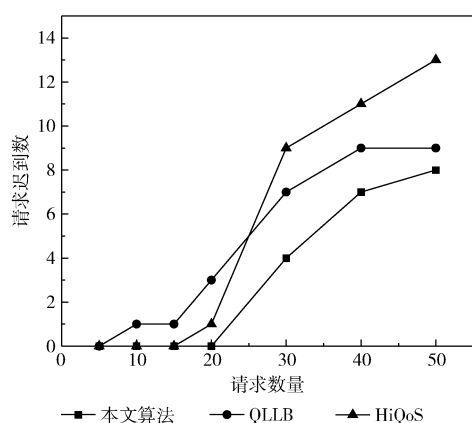


图4 请求迟到数对比

负载均衡系数 k (式 (4)) 对比。如图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至节点9) 或包含这个源目的节点对 (如: 节点2-节点3至节点9-节点10) 的请求将重复使用节点3至节点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它两种算法。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。

转发成本对比。链路的转发成本 (式 (5)) 表征使用该链路的所有代价总和。如图6所示, 本文算法的平均转发成本比 QLLB 算法低 10.2%。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。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。

总目标对比。本文优化总目标是最小化负载均衡系数和转发成本之和。如图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

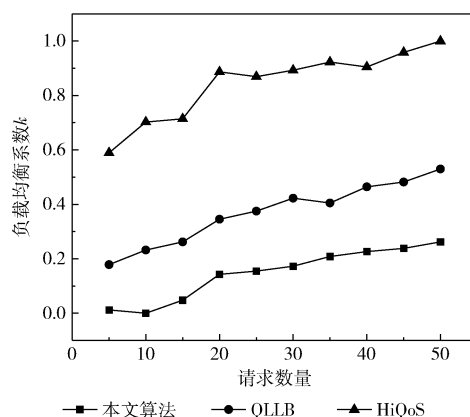


图5 负载均衡系数k对比

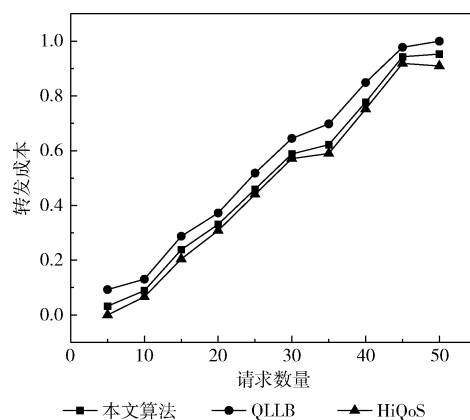


图6 转发成本对比

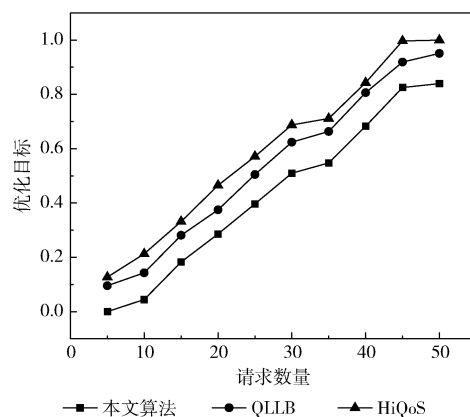


图7 总目标对比

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。

4 结束语

针对 SDN 架构在多媒体流服务场景的应用中存在的网络链路负载不均衡问题, 本文将问题建模为多目标和多约束模型。模型以最小化链路负载均衡系数和链路转发成本

为目标,同时满足每条请求的带宽、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。仿真结果表明,与QLLB和HiQoS算法相比,本文的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。在未来的工作中,将进一步优化调度规则以及研究路径选择中的方法。调度方面将考虑根据多媒体流的特性自动识别并调整相应的资源分配策略,例如优先为时延敏感的实时视频流提供较低延迟的网络路径,而为大容量的文件传输流提供高带宽的链路。路径选择方面将考虑分布式路径选择算法,使各SDN控制器能够在局部环境下独立进行路径选择决策,同时保持全局网络优化。以提高系统的扩展性,适应大规模复杂网络环境。

参考文献:

- [1] Belgaum MR, Musa S, Alam M, et al. A systematic review of load balancing techniques in software defined networking [J]. IEEE Access, 2020, 8: 98612-98636.
- [2] Alhilali A H, Montazerolghaem A. Artificial intelligence based load balancing in SDN: A comprehensive survey [J]. Internet of Things, 2023, 22: 100814.
- [3] Liu Chenyi, Xu Mingwei, Yang Yuan, et al. DRL-OR: Deep reinforcement learning-based online routing for multi-type service requirements [C]//IEEE INFOCOM-IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2021: 1-10.
- [4] Moin M M, Narayan D G, Patil S. A hybrid bio-inspired algorithm for routing in software defined networks [C]//12th International Conference on Computing Communication and Networking Technologies. IEEE, 2021: 1-7.
- [5] Aloboud E, Kurdi H. Cuckoo-inspired job scheduling algorithm for cloud computing [J]. Procedia Computer Science, 2019, 151: 1078-1083.
- [6] 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.
- [7] He Qiang, Wang Yu, Wang Xingwei, et al. Routing optimization with deep reinforcement learning in knowledge defined networking [J]. IEEE Transactions on Mobile Computing, 2024, 23(2): 1444-1455.
- [8] Aguirre Sanchez L P, Shen Yao, GUO Minyi. DQS: A QoS-driven routing optimization approach in SDN using deep reinforcement learning [J]. Journal of Parallel and Distributed Computing, 2024, 188: 104851.
- [9] Zhang Junjie, Ye Minghao, Guo Zehua, et al. CFR-RL: Traffic engineering with reinforcement learning in SDN [J]. IEEE Journal on Selected Areas in Communications, 2020, 38(10): 2249-2259.
- [10] Rezaei S, Liu Xin. Multitask learning for network traffic classification [C]//29th International Conference on Computer Communications and Networks. Honolulu: IEEE Press, 2020: 1-9.
- [11] Raikar M M, S M 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.
- [12] Ke Yu, Wang Junli, Yan Chungang, et al. Routing strategy for SDN large flow based on deep reinforcement learning [C]//20th IEEE Int Symposium on Parallel and Distributed Processing with Applicat/15th IEEE Int Conf on Social Comp and Networking/12th IEEE Int Conf on Big Data and Cloud Comp/12th IEEE Int Conf on Sustainable Comp and Commun. Piscataway: IEEE Press, 2022: 523-530.
- [13] WANG Weifa, ZHANG Daming, LIU Kunqian, et al. Q-learning load balancing algorithm in software-defined network [J]. Telecommunication Engineering, 2021, 61 (9): 1066-1072 (in Chinese). [王伟发, 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 (9): 1066-1072.]
- [14] Huang Feilong, Zhang Jianzhong, Xu Jingdong, et al. An SDN-based QoS guaranteed mechanism for geospatial flows [C]//IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking. Piscataway: IEEE Press, 2019: 1394-1401.
- [15] Jain S, Jain V K, Mishra S. Fuzzy-AHP model for intelligent network selection in VANET for data offloading [C]//IEEE 7th Conference on Information and Communication Technology. IEEE, 2023: 1-5.
- [16] LIU Chuyi, WAN Jianxiong, LI Wenjing. Single cluster cloud resource management algorithm based on deep reinforcement learning [J]. Journal of Inner Mongolia University of Technology (Natural Science Edition), 2023, 42 (5): 459-466 (in Chinese). [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(5): 459-466.]
- [17] Ye Junhong, Zhang Yingjun. DRAG: Deep reinforcement learning based base station activation in heterogeneous networks [J]. IEEE Transactions on Mobile Computing, 2019, 19(9): 2076-2087.
- [18] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23 (6): 4909-4926.
- [19] 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.
- [20] 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.