DHRL-ACTF：一种新的 SDWN 智能链路故障感知自适应跨域路由算法

叶 苗[1),2)] 李锦强[1)] 何 倩[1),2)] 王晓丽[1),3)] 王宇平[3)]
王 勇[1),2)] 文 鹏[1)]

[1)]（桂林电子科技大学广西无线宽带通信与信号处理重点实验室 广西 桂林 541004）
[2)]（桂林电子科技大学计算机与信息安全学院 广西 桂林 541004）
[3)]（西安电子科技大学计算机科学与技术学院 西安 710071）

摘 要 软件定义无线网络 SDWN 多域划分管理可以将规模大的网络划分为多个子网络进行管理，但多域 SDWN 域间路由消息传递和同步以及路由建立存在全局网络状态信息获取困难、对网络状态动态变化响应慢影响域间路由收敛速度导致传输时延增加、网络链路故障发生时路由策略适应性差转发不够及时不够灵活的问题。对此，本文在层次多控制器结构下设计了一种基于 Transformer 流量预测与分层强化学习的 SDWN 跨域智能链路故障感知自适应路由方法。首先，基于层次多控制器架构设计实现的多控制域间消息传递同步机制为及时获取全局网络状态信息、设计灵活的智能跨域路由方法提供全局数据视图数据基础；其次，设计了一种 SDWN 网络测量机制以灵活获取每个网络子域中包括剩余带宽、时延、丢包率的网络状态信息，通过设计的故障链路矩阵实时感知链路状态，同时使用 Transformer 预测机制感知未来流量变化的趋势以实现对网络态势感知功能；然后，在根控制器模块设计了基于 Actor-Critic 架构的分层强化学习跨域路由机制，将跨域路由问题分解成两个子问题：域间节点选择问题和构造从该域间节点抵达目的节点的最优路径选择问题。通过将基于 Transformer 预测后的网络流量矩阵和具备故障感知能力的故障信息矩阵、域信息矩阵、子目标矩阵、智能体位置矩阵等原始网络流量矩阵作为内外控制器的状态空间；再次，为了使智能体能够适应复杂的网络环境场景并解决动作空间太大导致收敛速度慢、路由策略调整不够灵活的问题，元控制器将域中所有节点作为动作空间用于完成域间节点的选择，内部控制器则将当前节点的邻边作为动作空间，并分别设计了五种不同动作选择策略来优化路由，以减少冗余支路的产生并动态调整智能体学习轨迹；最后，采用网络链路信息和不同奖惩机制来设计内外部控制器的奖励函数，以引导智能体朝着规避故障链路具有更高奖励值的方向学习，从而高效利用网络资源生成转发策略灵活适应能力强的跨域智能路径。通过系列实验及其结果表明所提方法在不同网络状态下以及随机链路故障中有着不错的网络性能，平均带宽相较于现有 PPO、DQN、Q-learning、OSPF 和 BGP 方法分别提升了 15.51%、37.88%、37.79%、38.31%和 50.86%；平均时延在与 PPO 方法表现相近的同时，比 DQN、Q-learning、OSPF 和 BGP 方法有明显提升，而丢包率和弃包数也均优于这些现有方法，表现出了较优的跨域路由性能。本文将所做工作源代码提供至了开源平台 https://github.com/GuetYe/DHRL-ACTF。

关键词 软件定义无线网络；跨域路由；链路故障感知；深度分层强化学习；流量预测
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2025.02666

收稿日期：2024-10-31；在线发布日期：2025-08-15。本课题得到国家自然科学基金（62161006）、国家自然科学基金面上项目（62372353）、国家自然科学基金重点项目（U22A2098）、广西无线宽带通信与信号处理重点实验室基金项目（桂科 AD25069102）、认知无线电与信息处理教育部重点实验室主任基金（CRKL220103）、广西研究生教育创新计划项目（2025YCXS078）资助。**叶 苗**，博士，教授，博士生导师，主要研究领域为边缘存储与云存储、软件定义网络、无线传感器网络、强化学习、模式识别与机器学习。E-mail：ym@mail.xidian.edu.cn。**李锦强**，博士研究生，主要研究领域为软件定义网络、强化学习。**何 倩**，教授，博士，博士生导师，中国计算机学会（CCF）杰出会员，主要研究方向为云服务、数据要素管理及大数据、软件定义网络。**王晓丽（通信作者）**，博士，副教授，硕士生导师，主要研究方向为人工智能、云/雾计算、并行与分布式计算、高效任务调度。E-mail：wangxiaoli@mail.xidian.edu.cn。**王宇平**，博士，教授，博士生导师，主要研究领域为进化计算、最优化理论、数据挖掘、人工智能。**王 勇（通信作者）**，博士，教授，博士生导师，中国计算机学会（CCF）杰出会员，主要研究领域为云计算、网络流量分类、信息安全等。Email：ywang@guet.edu.cn。**文 鹏**，博士研究生，主要研究领域为软件定义网络、强化学习、随机优化与应用。

DHRL-ACTF: A New SDWN Intelligent Link Failure-Aware Adaptive Cross-Domain Routing Algorithm

YE Miao[1),2)]　LI JIn-Qiang[1)]　HE Qian[1),2)]　WANG Xiao-Li[1),3)]　WANG Yu-Ping[3)]
WANG Yong[1),2)]　WEN Ping[1)]

[1)] (Guangxi Wireless Broadband Communication and Signal Processing Key Laboratory, Guilin University of Electronic Technology, Guilin, Guangxi 541004)

[2)] (School of Computer Science and Information Security, Guilin University of Electronic Technolgoy, Guilin, Guangxi 541004)

[3)] (School of Computer Science and Technology, Xidian University, Xi'an 710071)

Abstract The multi-domain partitioning management method of software-defined wireless networks (SDWN) can divide large-scale networks into multiple subnetworks for management. However, multi-domain SDWN inter-domain routing message transmission and synchronization, as well as routing establishment, face challenges such as difficulty in obtaining global network status information changes, slow response to dynamic changes in network status affecting inter-domain routing convergence speed and increasing transmission latency, and poor adaptability of routing strategies when network link failures occur, resulting in insufficiently timely and flexible forwarding. To address these issues, this paper designs a cross-domain intelligent link failure-aware adaptive routing method for an SDWN based on transformer traffic prediction and hierarchical reinforcement learning in a hierarchical multiple-controller structure. First, a message transmission synchronization mechanism between multiple control domains is designed and implemented under a multi-level controller architecture, providing a global data view data foundation for obtaining global network status information in a timely manner and designing flexible intelligent cross-domain routing methods. Second, an SDWN network measurement mechanism is designed to flexibly acquire the state information of each network subdomain, including the remaining bandwidth, delay, packet loss rate, and wireless node distance; a link failure matrix is designed to map the link state in real time, and a transformer prediction mechanism is used to sense the trend of future traffic changes to achieve network situational awareness. Third, an actor-critic architecture-based hierarchical reinforcement learning mechanism is designed in the root controller module to decompose the cross-domain routing problem into two subproblems: the selection of an interdomain node and the construction and selection of an optimal route from this interdomain node to the destination node. The predicted network traffic matrix, the original network traffic matrix, and the failure information matrix are used as the state spaces of the internal controller, and the domain information matrix, the subgoal matrix, and the agent position matrix are used as the state spaces of the external controller. Fourth, to enable the agents to adapt to the complex network environment and to address the issue of slow convergence speed and insufficient flexibility in routing strategy adjustment caused by excessive action space, a metacontroller uses all nodes in the domain as the action space to select the interdomain nodes, while the internal controller uses the neighboring edges of the current node as the action space. Five different action selection strategies are designed to optimize routing to reduce the generation of redundant branches and dynamically adjust the learning trajectory of the agents. Finally, the network link information and different reward and punishment mechanisms are used to design the reward functions for the internal and external controllers to guide the agents to learn in the direction of higher rewards and avoid link failure, thus efficiently using network resources to generate cross-domain intelli-

gent paths with better performance. A series of experiments show that the proposed method has good network performance under different network states and random link failures. The average bandwidth of the proposed method is 15.51%, 37.88%, 37.79%, 38.31%, and 50.86% greater than those of the existing PPO, DQN, Q-learning, OSPF, and BGP methods, respectively. The average delay of the proposed method is similar to that of the PPO method and significantly better than those of the DQN, Q-learning, OSPF, and BGP methods. Additionally, the proposed method outperforms these existing methods in terms of packet loss rate and packet drop rate, demonstrating superior cross-domain routing performance. The source code of the work reported in this paper is made available on the open-source platform https://github.com/GuetYe/DHRL-ACTF.

Keywords software-defined wireless networking; cross-domain routing; link failure detection; deep hierarchical reinforcement learning; traffic prediction.

1 引 言

随着云计算、大数据和 5G 网络等各种网络技术的快速演进和发展，无线网络规模和网络流量也呈现出显著的增长，移动终端用户对无线网络的服务质量也提出了更高的需求。此外，移动无线用户的突发式流量和高速动态移动特性导致的网络拥塞对提高网络质量(Quality of Service, QoS)和用户服务体验(Quality of Experience, QoE)带来了严峻挑战[1]。传统无线网络架构在管理网络和处理动态变化的网络流量时存在容易产生传输性能瓶颈、网络链路信息测量获取受限难以管理、资源利用效率低等一系列问题[2]，严重影响了用户对通信服务质量的体验效果。研究设计一种高效适应网络高速动态变化特点的无线网络路由方法，最大限度减少链路故障对网络的影响，从而保障数据传输的可靠性和稳定性，提高无线网络流量高速动态变化情形下的数据传输效率以适应当前无线数据传输业务场景复杂多样的需求，对于合理分配和利用网络资源具有重要的理论价值和实践意义。

传统无线网络通常采用集中式的架构，其中数据转发和控制管理的高度耦合限制了网络的扩展能力[3]。此外，网络的高度异构化、网络规模不断扩张，以及底层设备种类和协议繁多等问题，给网络配置和管理都带来了严峻挑战。软件定义无线网络(Software Defined Wireless Network, SDWN)[4]对此提供了有效的解决方案，可以提高无线网络架构的可编程性和灵活性，实现高效灵活的网络配置和管理。SDWN 作为软件定义网络(Software Defined Network, SDN)的扩展性无线网络架构，通过解耦数据平面和控制平面，并利用全局网络拓扑和流抽象的网络视图动态调整网络流量以优化整体网络资源。因此 SDWN 网络架构为高效及时获取无线网络状态信息提供了可能。

近年对 SDWN 架构下大规模网络路由优化解决方案通常分为三种：集中式、分布式和混合路由方法[5]。在集中式方法中，整个网络由一个中心控制器或者中心化的实体进行管理和控制。中心控制器使用内置路由方法计算最佳路由路径，并将决策下发到无线 AP 节点，以便沿着最佳链路传高效传输数据包。例如 Zhu 等人[6]设计了基于 SDN 的集中式 QoS 路由方法，采用中心控制器获取全局网络状态信息进行路由决策。随着网络规模的扩大，基于单控制器的 SDWN 网络控制平面负载会急剧增加，从而导致控制器性能下降、处理延迟增加等问题，甚至会出现流量数据包堆积而造成网络链路故障的现象，严重影响网络的整体性能。分布式路由方法中，路由决策由网络中的各个节点自行完成，节点之间通过交换路由信息来协作完成路由。例如 Binita 等人[7]提出一种分布式的 QoS 自适应路由方法，但该方法重点考虑的是源到目的节点之间的跳数。分布式路由算法通过在邻居节点之间有限的泛洪来减少网络负载和管理压力。虽然，该方法有较好的去中心化特点，能够降低单点故障的风险，但是可能会面临路由信息同步和一致性维护的挑战。混合式方法是一种分布式多控制器的 SDWN 路由方法，将集中式和分布式方法相结合，既保留了中心化管理的优点，又具有去中心化的特性。通过将网络划分成多个子域，每个子域由一个本地控制器进行管理，并提供解除无线网络规模和路由计算复杂性耦合的潜力，为大规模网络路由优化提供了重要思路。为了

解决以上单控制器管理模式下的大规模网络出现的性能瓶颈问题,基于多控制器模式下的分域管理已经成为一种关键性技术。通过这种架构,网络被划分为多个由独立控制器管理的区域,有效提升了整体网络的处理能力和伸缩性,从而显著增强了网络的性能表现。在这样多域 SDWN 架构设计出灵活的域间路由机制就显得非常重要。

相比单域 SDWN 中路由信息转发不同,在多域 SDWN 场景下,不同域之间的路由转发与信息协作,本地控制器必须考虑邻近域的转发决策,确定转发的最佳邻近域,并选择适当的域间入口点是一个关键性问题。由于多域场景下仍旧存在网络状态信息高速动态变化的特点,设计出高效合理的跨域路由方法还需要解决域间消息传递和同步的问题。传统的边界网关协议(BGP, border gateway protocol)[8]是通过边界路由器进行相邻自治系统(AS, autonomous system)之间的消息转递,实现域间的消息同步。但基于 BGP 的方式实现消息传递与消息同步在 SDN 环境中配置较为繁琐且存在路由震荡等问题。深度强化学习(DRL)[9]是一种融合了深度学习与强化学习的技术,它赋予了智能体通过不断试错来学习最佳决策动作策略。由于其具备处理复杂问题和适应变化环境的能力,DRL 已广泛应用于包括路由优化[10-12]的各种领域。相比单域 SDWN 场景下使用强化学习方法求解路由问题不同,在多域场景下使用强化学习方法还需要解决全局网络信息的获取不及时和收敛速度缓慢的问题,同时在多域 SDWN 场景下,现有深度强化学习方法面临动作空间维度高导致收敛速度慢、路由策略调整不够灵活的问题。此外,由于网络链路负载增加导致增加发生故障的可能,经常使得原有路由路径失效或性能下降,这就要求在多域 SDWN 场景下设计的深度强化学习路由方法不仅具备适应网络高速动态变化的特点,还具备链路故障感知、能灵活调整路由策略、满足良好实时性的要求的特点。

为了解决以上提到的强化学习方法在求解 SDWN 跨域路由问题时由于全局网络信息获取不及时、动作空间维度高导致收敛速度缓慢的问题,以及网络故障的频繁发生带来的路由策略适应性差转发不够灵活的问题,本文提出了一种基于 Actor-Critic 深度分层强化学习(Deep Hierarchical Reinforcement Learning, DHRL-AC)和 Transformer 网络流量预测的 SDWN 跨域智能链路故障感知自适应路由算法 DHRL-ACTF。为解决多域 SDWN 场景下网络全局网络状态信息获取和控制器负载均衡的问题,本文采用了多级层次结构的 SDWN 网络架构,对根控制器、域内本地控制器设计了层次化通信机制组成。在这种网络架构方式下,首先,通过 Transformer 网络流量预测机制去感知网络中流量变化的趋势,并预测网络链路中隐藏的状态信息,以此实现对全局网络链路状态信息的获取与预测;其次,在根控制器模块了一种基于 AC 网络架构的分层强化学习机制,该方法将跨域路由问题分解成两个子问题,即域间节点选择问题和构造从该当前节点抵达子目的节点的最优路径选择问题。其中将预测后的网络流量矩阵(剩余带宽、时延、丢包率、弃包数、无线 AP 之间的距离)和故障信息矩阵,以及域信息矩阵、网络拓扑结构视为 DHRL 的环境,而子域边缘交换机节点作为分层学习子目标集合。并设计不同维度的内外层控制器动作机制,来调整智能体的学习轨迹。同时采用多域网络链路信息和不同奖惩机制来设计内外层控制器奖励函数,以指引智能体朝着规避故障链路具有更高奖励值的方向学习,从而确定转发的最佳邻近域,并选择适当的域间入口点,最大限度减小网络链路故障的影响,并构建转发策略出灵活的跨域智能链路故障感知的自适应路由。

本文的创新点如下:

(1)针对现有 SDWN 跨域路由方法中全局网络信息捕捉不及时和收敛速度缓慢的问题,本文基于 Transformer 流量预测机制和分层强化学习方法设计了一种高效的具备网络全局信息感知的 SDWN 自适应跨域路由方法。该方法采用了基于 Transformer 流量预测机制,能够显著加快设计的层次强化学习跨域路由方法的收敛速度,同时提升可靠性和稳定性。

(2)针对现有深度强化学习方法在解决多域 SDWN 场景下的跨域路由问题时面临动作空间维度高导致收敛速度慢、路由策略调整不够灵活的问题,本文设计了一种基于分层强化学习的 SDWN 跨域路由机制。将构造跨域路由问题分解为域间节点选择和域内路由节点选择两个子问题,并分别由元控制器智能体和内部控制器智能体来分别解决,以此可以降低求解问题的规模和维度;分别为元控制器智能体和内部控制器智能体的不同动作策略设计了相应合适的奖励机制,有效避免了奖励函数过于稀疏导致难以收敛的问题,从而提高了智能体的探索效率,并加快了路由收敛速度。

(3)相较于现有文献中使用强化学习方法解决路由优化问题时并未考虑到网络链路故障所引起的拓扑变化从而无法处理动作空间维度变化的情形,本文设计的分层强化学习的跨域路由机制具备故障链路感知功能并具备很好的泛化性。在分层强化学习框架下,通过设计故障链路矩阵来实时映射链路状态,将失效的节点视为无效动作,同时设计适当的奖惩函数形式完成指引智能体规避故障链路。设计的方法由于能对链路故障进行智能响应、灵活适应拓扑的动态变化从而具备一定的泛化性能。

本文余下内容组织如下:第 2 节相关工作介绍主流网络流量预测模型和多控制器路由算法以及故障链路感知的自适应路由方法研究现状和存在的问题;第 3 节介绍跨域路由问题与建模;第 4 节介绍分层强化学习的 SDWN 跨域链路故障感知自适应路由架构;第 5 节介绍流量预测算法和 DHRL-ACTF 算法环境的设计以及实现智能跨域路由的细节;第 6 节介绍实验环境的设置,以及通过相关实验验证本文算法的稳定性和可靠性;第 7 节为结论,同时给出面临的挑战,并提出进一步的研究方向。

2 相关工作

本节主要介绍常用多域 SDN 路由方法、用于提升路由性能的流量预测方法以及具有故障链路感知功能的自适应路由方法。

2.1 路由优化方法

基于多控制器管理下的路由优化问题,目前主要研究方法分为传统路由方法、启发式方法和强化学习的方法。

(1)传统路由方法:传统路由方法主要基于最短路径算法的开放式最短路径优先协议(Open Shortest Path First, OSPF)算法以及域间路由的边界网关协议(Border Gateway Protocol,BGP)。然而,在高速动态变化的无线网络中,传统路由算法无法根据实时的网络状态信息来更新路由表,存在收敛速度慢、响应时间长难以适应网络流量动态变化的问题。Caria 等人[13]提出了一种混合 SDN/OSPF 操作的新方法,通过 SDN 节点将 OSPF 域划分为子域,从而实现与完整 SDN 操作相当的流量工程能力。如 Kotronis 等人[14]为改善 BGP 收敛缓慢的问题,提出了一种改进 BGP 协议的混合 BGP-SDN 跨域路由方法,重点关注利用多域集中化来改善 BGP 收敛速度缓慢的问题。虽然在一定程度上,改进 OSPF 和 BGP 协议能够加速算法的收敛速度,但由于无线网络动态高速变化,传统路由方法无法及时获取全局网络状态信息、难以适应复杂动态变化网络环境,导致路由计算困难以及算法收敛速度慢的问题。

(2)启发式方法:Moufakir 等人[15]针对 SDN 多域网络环境中子域间缺乏协作性的问题,提出了一种创新的协作多域路由框架。该框架通过利用不同域的传入流进行路由决策,并采用贪心算法来优化路由策略,同时考虑了链路时延和带宽的影响。尽管贪心算法在寻找解决方案时效率较高,但容易陷入局部最优解。Karakus 等人[16]为了高效利用资源分配以及改进网路的性能,提出了一种基于遗传算法的 QoS 感知跨域网络流量工程架构,他们重点关注网络带宽、时延和网络可靠性方面,同时能够在网络连接或节点故障时将流量转移到满足 QoS 的路径,确保关键服务能够在网络中断中保持运行。采用遗传算法等启发式方法相比贪心算法减轻了容易陷入局部最优解的问题,但迭代计算方式导致计算开销大,很难适应网络高度动态变化的特点。Xu 等人[17]为了解决 SDN 多控制器的负载不平衡问题,使用控制器集群的负载平衡、平均延迟和交换机迁移成本作为决策因子,设计了一种改进的多目标优化遗传算法(GAIMO),通过局部搜索算子来减少模型的训练时间,用于解决遗传算法中最优解容易滑向局部最优的问题,有效提升了 SDN 多控制器的总体资源利用率,降低了网络的平均延迟和通信开销,优化了整个网络的性能。然而,该方法需要有严格的使用场景,在发生故障时网络中拓扑和链路信息发生变化都将使启发式方法出现较大波动和误差,同样存在计算开销大,很难适应网络高度动态变化的特点。

(3)强化学习方法:强化学习用于路由问题的研究成为了近几年研究的热点,特别是在单域场景下路由问题的研究。在深度强化学习出现之前,Boyan 和 Littman[18]首次尝试利用经典强化学习中 Q-learning 学习方法对数据包进行路由优化,其结果表明了在平均数据包传递时间上优于了最短路径算法。近年来 Casas-Velasco 等[19]在 SDN 中提出了一种基于 Q 学习的智能路由算法。随着深度强化学习的发展,Yao 等[20]提出的基于深度 Q 网络的能效路由算法,有效解决了软件定义数据中心网络中的能量消耗和负载均衡问题。Lu 等[21]提出了一种基于深度确定性策略梯度(DDPG,Deep determin-

istic policy gradient)的子流自适应多路径路由算法有效适应了数据中心网络中的动态流量变化,降低了延迟、提高了传输成功率并增加了吞吐量。Zhou等[22]提出了一种基于近端策略优化(PPO,Proximal policy optimization)的 QoS 感知路由优化机制(PQROM, PPO-based QoS-aware routing optimization mechanism),具有良好的收敛性和稳定性,且在训练时间、超参数调整和硬件消耗方面均得到提升。但这些研究都是在单域场景下对路由规划问题进行的讨论,而且没有考虑网络故障对路由性能的影响。

相比单域 SDWN 场景下使用强化学习方法求解路由问题不同,在多域场景下使用强化学习方法的研究工作正在逐渐受到关注,这需要解决多域网络全局网络信息获取的及时性问题,和现有深度强化学习方法在多域 SDN 场景下面临的动作空间维度高导致收敛速度慢、路由策略调整不够灵活的问题。Godfrey 等人[23]提出了一种基于强化学习的多控制器下的无线传感器路由协议,通过将权重因子分配给奖励函数的聚合目标来优化不同的目标策略,并根据传感器节点到汇聚点的跳数距离作为 Q 一路由的可执行动作,通过多控制器网络架构,用于持续监测网络监控,使得 Q-learning 代理能够根据网络条件进行动态调整网络路由。虽然 Q-learning 能够在一定程度上解决多目标优化问题,但对于多域 SDN 网络情形,该方法无法适应高维的动作空间,导致获取动作特征能力不足,所以在多域无线动态变化的网络流量中,该方法并不适用。Albert 等人[24]在多域场景中采用深度强化学习方法进行自我优化的多域服务配置的新方法,但该方法中并未使用到流量预测的机制,无法及时捕获网络流量中所隐藏的流量数据。Li 等人[25]为了解决多域网络中传统域间路由方法在应对大规模网络和网络流量方面的不足,他们提出了一种基于 Actor-Critic 网络架构的多智能体强化学习机制。该机制能够动态地满足域间路由的需求,并以端到端时延、吞吐量和平均交付率为主要优化目标。通过关注这些不同的优化目标,实验结果显示,所提方法在性能上显著优于传统的域间路由策略。Huang 等人[26]在多域网络架构中,采用了基于 Dueling DQN 网络架构的多智能体强化学习方法来解决跨域路由问题,其中为同步根控制器与本地控制器的信息,设计了一种协同通信模块,以此减少控制器之间通信的开销。在优化跨域路由问题中还采用了流量预测机制进行感知环境中的流量情况,从而构造出最优的跨域路由。然而以上方法,并未考虑不同域之间的路由协作,本地控制器邻近域的转发决策,以及如何确定转发的最佳邻近域,并选择适当的域间入口点的顺序决策问题,此外,该方法没有考虑由于故障等因素发生网络拓扑结构改变时的情形。由于网络链路负载增加会增加发生故障的可能,经常使得原有路由路径失效或性能下降,这就要求多域场景下的深度强化学习路由方法不仅具备适应网络高速动态变化的特点,还要求具备链路故障感知、灵活调整路由策略和满足良好实时性的特点。深度分层强化学习[27]是强化学习领域的一种先进方法,它主要以深度强化学习为基础,融入了分层架构,通过将复杂任务进行分解成一个由较简单任务组成的层次结构子任务,从而降低整体任务的复杂度,提高了学习效率、探索能力和泛化能力,其主要用于解决稀疏奖励、顺序决策以及弱迁移的问题,并为高维路由问题的解决提供重要的解决思路。

2.2　流量预测方法

网络流量预测本质上是对网络流量矩阵(Traffic Matrix, TM)进行预测,通过准确预测网络流量,可以提前识别潜在的网络故障问题并有效降低网络延迟。这有助于重构路由算法进行容错处理,并提高算法的稳定性和可靠性。通常,网络流量预测方法可划分为线性和非线性两种模型。

(1)线性预测模型:通常采用统计模型的方式去预测网络流量。如 Moayedi 和 Masnadi-Shirazi 等人[28]将数据分解成正常(可被预测)和异常(不可被预测)两种类型,并采用自回归滑动平均模型(ARIMA, Auto Regressive Integrated Moving Average Model)进行分析和预测出现异常的数据。Kim 等人[29]采用整数值广义自回归条件异方差来捕获或预测物联网和车载自组网的网络流量中的非线性特征。然而,上述方法通常没有考虑到无线网络节点空间依赖相关性,而无线网络中往往会涉及移动节点和信号衰落的影响,在多域网络中还会存在网络链路故障,因此基于线性预测的方法具有一定的局限性。

(2)非线性预测模型:目前非线性预测模型主要是基于机器学习和深度学习的方法。Nikravesh 等人[30]对比了机器学习中多层感知器(Multi-Layer Perceptron,MLP)、具有权重衰减的多层感知器(Multi-Layer Perceptron with Weight Decay, MLPWD)以及支持向量机(Support Vector Ma-

chines,SVM)在移动网络中多维流量数据上的预测性能。机器学习方法虽然可以挖掘一些复杂的网络流量数据,但其特征提取能力有限。于是,许多研究学者采用深度学习方法来预测网络流量,并取得了不错成果。Huang 等人[31]提出采用门控循环单元 GRU 替换 LSTM 模型的方案去预测 SDN 中 TM 信息,从中挖掘网络中所隐藏的流量信息,构造出相应的预测流量矩阵,接着使用预测后的流量矩阵去训练智能体,提升了路由算法的可靠性。虽然这些相关研究解决了线性预测精度低和泛化能力弱的缺陷,并取得了不错的预测效果,但都只考虑了网络流量的时间特征,没有考虑网络流量的空间特征,对于具有时空特性的无线网络流量而言,预测的效果并不理想。于是 Xu 等人[32]提出使用三维卷积神经网络(3D-CNN)和长短期记忆网络(LSTM)相结合的方式去预测无线网络中的流量信息;Pan 等人[33]充分考虑到网络流量之间复杂的时间和空间依赖性,提出将图卷积神经网络 GCN 与门控循环单元 GRU 结合的方式去预测网络流量,从实验结果看出在具有时空特性的流量数据上有着长期的预测能力和适用性,预测的效果也达到了预期。

2.3 基于 Transformer 流量预测和强化学习的智能路由方法

Transformer 架构自 2017 年首次提出以来已经在深度强化学习中已经得到广泛应用,包括在表征学习、建模状态转移、学习奖励函数及策略优化等关键环节。但我们通过仔细查阅发现:使用 Transformer 进行流量预测的多数研究是在交通流量预测和车辆调度领域,在计算机网络流量预测领域的文献工作非常少,最近能查阅到的这方面中文文献中有影响力的仅仅是文献[34]关于网络流量预测方面的工作,但该文献没有涉及流量预测提升路由性能方面的工作;基于 Transformer 流量预测技术用于网络路由方面的中文文献研究仅有两篇 2023 年和 2024 年关于光网络路由方面的硕士论文,英文文献的工作是 Hameed. A 在关于 IoT 物联网中对网络 QoS 优化的工作[35]。由以上对中文和英文文献的检索及结果我们可以看出,目前的 Transformer 进行流量预测的研究基本集中于车辆或者道路交通领域,在计算机通信网络中流量预测的虽然有但不是很多,在通信网络中路由的工作更少,而结合强化学习方法就没有相关研究工作了。

2.4 故障链路感知的自适应路由方法

设计具有故障链路感知的自适应路由方法,能够有效保障网络数据的稳定传输,从而提升整体网络的稳定性和可靠性。如 Malik 等人[36]在 SDN 网络中的数据平面设计了一种故障感知的新方法,它允许控制器能够提前接收到链路故障的预警信息,从而在故障发生之前避免危险路径,所提方法重点在于最大限度减少服务中断,以提高网络服务的可用性。所提方法虽然在一定程度上减少了网络链路发生故障的情形,却忽视了路由调整后网络的整体性能情况。于是 Liu 等人[37]提出了基于 Q-learning 的网络容错和拥塞感知自适应路由算法,通过 Q-learning 学习源到目的之间的拥塞和故障信息,避免数据在故障区域不必要地绕道,以此选择拥塞较少的路径,从而提升网络的整体性能。然而,随着网络规模的增大,采用 Q-learning 的强化学习方法,该方法无法适应高维的动作空间,导致获取动作特征能力不足,因此该方法并不适用。Valinataj 等人[38]设计了一种分布式、自适应和拥塞感知路由方法,所提方法能够处理网络故障引起的不规则拓扑情形,并能够在复杂的网络链路故障中,动态调整路由策略,以确保网络的可靠性和稳定性。以上方法虽然在一定程度上能够最大限度减少网络故障的影响,但对于复杂多变的大规模无线网络场景,它们无法解决跨域路由中选择最佳邻近域和合适的域间入口点选择的复杂决策问题,因此在多域网络场景下需要重新设计新的路由机制。

针对以上提到的现有 SDWN 跨域路由方法由于获取全局网络信息不及时、动作空间维度高导致收敛速度缓慢,以及网络故障的频繁发生时路由策略适应性差转发不够灵活的问题,本文提出了一种基于 Actor-Critic 深度分层强化学习(Deep Hierarchical Reinforcement Learning, DHRL-AC)和 Transformer 网络流量预测的 SDWN 跨域智能链路故障感知自适应路由算法 DHRL-ACTF。通过 Transformer 网络流量预测模型去感知网络中流量变化的趋势,并预测网络链路中隐藏的状态信息,以此实现对全局网络链路状态信息的获取与预测。在根控制器模块设计了一种基于 AC 网络架构的分层强化学习机制,该方法将跨域路由问题分解成两个子问题,即域间节点选择问题和构造从该当前节点抵达子目的节点的最优路径选择问题,其中将预测后的网络流量矩阵(剩余带宽、时延、丢包率、弃包数、无线 AP 之间的距离)和故障信息矩阵,以及域信息矩阵、网络拓扑结构视为 DHRL 的环境,而子域边缘交换机节点作为分层学习子目标集合。并设

计不同维度的内外层控制器动作机制,来调整智能体的学习轨迹。同时采用多域网络链路信息和不同奖惩机制来设计内外层控制器奖励函数,以指引智能体朝着最大奖励值的方向学习,从而确定转发的最佳邻近域,并选择适当的域间入口点,最大限度减小网络链路故障的影响,并构建较优的跨域智能链路故障感知的自适应路由。

3　问题描述与建模

本文所讨论的 SDWN 跨域路由问题主要关注如何高效处理不同网络子域之间的路由请求转发,以及两个节点之间的链路发生故障引起网络拓扑发生变化时系统如何及时响应网络变化快速重新计算和生成新路由,最大限度地减少网络故障的影响,从而确保数据传输的可靠性和稳定性。

假设用 $G=(V,E)$ 表示这样的 SDWN 多域网络,其中,V 是网络节点的集合,$n=|V|$ 表示节点的个数,E 表示网络中边的集合,$e_{i,j}\in E$ 表示从节点 i 到节点 j 的边。G 可以划分为多个控制域,这些网络子域的集合可以用$\mathcal{D}$来表示,假设每个控制器管理的其中一个子域表示为 $\mathcal{D}_i$,则每个子域就是 G 的一个子图 $\mathcal{D}_i=(V_i,E_i)$,$\mathcal{D}_i$ 包含的节点集合可以表示为 $V_i=\{v_{i,1},v_{i,2},v_{i,3},\cdots,v_{i,n_i}\}$,$n_i=|V_i|$ 表示域 $\mathcal{D}_i$ 中的节点个数,而 $E_i=\{(v_{i,j},v_{i,j'})\mid v_{i,j}\in V_i,v_{i,j'}\in V_i\}$ 表示域 $\mathcal{D}_i$ 内的边集合。同时用 $E_{i,i'}=\{(v_{i,j},v_{i',j'})\mid v_{i,j}\in V_i,v_{i',j'}\in V_{i'}\}$ 表示从域 $\mathcal{D}_i$ 到其他子域 $\mathcal{D}_{i'}$ 中的域间链路集。

给定任意一个域 $\mathcal{D}_i$,域内任意两个不同节点 i 和 j 之间的链路所已使用带宽资源和总带宽资源分别用 $b_{i,j}^{used,\mathcal{D}_i}$ 和 $b_{i,j}^{total,\mathcal{D}_i}$ 来表示。对于域 $\mathcal{D}_i$ 中的节点 j 和其他域 $\mathcal{D}_{i'}$ 中的节点 j' 之间的域间链路所使用的带宽资源和总带宽资源分别用 $b_{j,j'}^{used,\mathcal{D}_i,\mathcal{D}_{i'}}$ 和 $b_{j,j'}^{total,\mathcal{D}_i,\mathcal{D}_{i'}}$ 来表示。

为了建立模型时表述上的方便,本文用故障感知变量 $Y_{i,j}$ 表示从节点 i 到节点 j 是否发生故障的链路状态,$Y_{i,j}\in\{0,1\}$,其中 1 表示链路处于正常状态,0 表示链路处于故障状态。

区别于其他文献[39-41]通常采取单一性能指标的做法,本文这里为了尽量全面考虑网络状态,因此度量生成的路由路径质量主要考虑以下五个性能指标:链路剩余带宽 u_{bw}、时延 u_{delay}、丢包率 u_{loss}、弃包数 u_{drop}、无线 AP 之间的距离 u_{dist}。

假设 $p(src,dst)=(V_p,E_p)$ 表示生成的从源节点 src 到目的节点 dst 的跨域路由路径上的路径,其中 V_p 表示节点集合,E_p 表示边的集合。则 u_{bw} 表示跨域路径 $p(src,dst)$ 上所有链路带宽的最小值,也就是路径 $p(src,dst)$ 上的瓶颈带宽,如公式(1)所示。其余度量参数时延 u_{delay}、丢包率 u_{loss}、弃包数 u_{drop}、无线 AP 之间的距离 u_{dist} 由公式(2)~(5)表示。

$$u_{\mathrm{bw}}=\min_{\mathcal{D}_i,\mathcal{D}_{i'}\in\mathcal{D},<i,j>\in E_p}\{(b_{i,j}^{total,\mathcal{D}_i}-b_{i,j}^{used,\mathcal{D}_i}),\cdots,(b_{i,j}^{total,\mathcal{D}_i,\mathcal{D}_{i'}}-b_{i,j}^{used,\mathcal{D}_i,\mathcal{D}_{i'}})\} \tag{1}$$

$$u_{delay}=\sum_{\mathcal{D}_i,\mathcal{D}_{i'}\in D}\sum_{<i,j>\in E_p}delay_{i,j}^{\mathcal{D}_i}+delay_{i,j}^{\mathcal{D}_i,\mathcal{D}_{i'}} \tag{2}$$

$$u_{loss}=1-\prod_{<i,j>\in E_p}(1-loss_{i,j}) \tag{3}$$

$$u_{drop}=\sum_{\mathcal{D}_i,\mathcal{D}_{i'}\in D}\sum_{<i,j>\in E_p}drop_{i,j}^{\mathcal{D}_i}+drop_{i,j}^{\mathcal{D}_i,\mathcal{D}_{i'}} \tag{4}$$

$$u_{dist}=\sum_{\mathcal{D}_i,\mathcal{D}_{i'}\in D}\sum_{<i,j>\in E_p}dist_{i,j}^{\mathcal{D}_i}+dist_{i,j}^{\mathcal{D}_i,\mathcal{D}_{i'}} \tag{5}$$

给定源节点 src 和目的节点 dst,要生成的 SDWN 跨域路由 $p(src,dst)$ 要求最大化瓶颈带宽 u_{bw},最小化时延 u_{delay}、丢包率 u_{loss}、弃包数 u_{drop} 和无线 AP 之间的距离 u_{dist},要优化的自变量是一条从源节点到目的节点的路径 $p(src,dst)$,这些优化目标还有可能会存在相关性,比如有可能在发生拥塞的距离最短路径上时延不是最短反而更长,因此时延最短但不一定路径最短。在数学优化理论中实际上是要优化这几个目标函数其实是一个多目标优化问题,严格来讲是找这到最优支配解组成的 pareto 前沿,再结合实际问题背景来挑选合适的个别 pareto 解,这是数学上多目标优化理论中的做法。在实际工程(比如这里网络流量工程中)通常的做法是,结合问题背景特点,比如流量工程希望路径的瓶颈带宽越大越好时延越小越好,将多个目标优化问题转化为单目标优化问题,而线性加权组合的方式就是常见的一种。因此,本文考虑将以上性能指标进行线性组合方式来进行优化求解,由此可以建立以下带约束条件的优化目标形式:

$$\min_{p(src,dst)}\beta_1\frac{1}{u_{bw}}+\beta_2u_{delay}+\beta_3u_{loss}+\beta_4u_{drop}+\beta_5u_{dist}+(1-Y_{i,j})C \tag{6}$$

Subject to

$$b_{i,j}^{total,\mathcal{D}_i}-b_{i,j}^{used,\mathcal{D}_i}\geqslant QoS_{band},b_{i,j}^{total,\mathcal{D}_i,\mathcal{D}_{i'}}-b_{i,j}^{used,\mathcal{D}_i,\mathcal{D}_{i'}}\geqslant QoS_{band},\forall<i,j>\in E_p \tag{7}$$

$$delay_{i,j}^{\mathcal{D}_i}+delay_{i,j}^{\mathcal{D}_i,\mathcal{D}_{i'}}\leqslant QoS_{delay},\forall <i,j>\in E_p \tag{8}$$

$$loss_{i,j}\leqslant QoS_{loss},\forall <i,j>\in E_p \tag{9}$$

$$drop_{i,j}^{\mathcal{D}_i}+drop_{i,j}^{\mathcal{D}_i,\mathcal{D}_{i'}}\leqslant QoS_{drop},\forall <i,j>\in E_p \tag{10}$$

$$dist_{i,j}^{\mathcal{D}_i}+dist_{i,j}^{\mathcal{D}_i,\mathcal{D}_{i'}}\leqslant QoS_{dist},\forall <i,j>\in E_p \tag{11}$$

$$C\gg\beta_1\frac{1}{u_{bw}}+\beta_2u_{bw}+\beta_3u_{loss}+\beta_4u_{drop}+\beta_5u_{dist} \tag{12}$$

$$\beta_1,\beta_2,\beta_3,\beta_4,\beta_5\in[0,1] \tag{13}$$

其中，C 表示事先设置的故障链路度量成本，一般为一个极大的常数。约束条件(7)表明，所求跨域路由路径的剩余带宽资源必须满足 QoS 带宽限制 QoS_{band}，一般为一个已知的常数。约束条件(8)—(11)表示所求路由路径的时延、丢包率、弃包数、无线 AP 之间距离的总长度不应超过相应的 QoS 阈值限制。约束条件(12)表明，故障成本常数 C 远远大于其余链路构造成本值之和，从 $Y_{i,j}$ 的取值可以看出，当链路没有故障时则由前五个参数衡量跨域路由的性能；否则需要加上故障链路度量成本。约束条件(13)表明，每个度量参数的权重因子限定在[0,1]区间范围之中，它们的不同取值反映了在生成跨域路由时，不同性能指标的重要程度的大小不同。

由数学中的多目标优化理论可知，当这些权值 $\beta_1,\beta_2,\beta_3,\beta_4,\beta_5$ 取不同权值的组合时对应了不同的 pareto 支配解，所有这些 pareto 支配解就能组成 pareto 前沿，这是本文采用线性组合方式的一个依据。此外，对不同优化目标采用线性加权组合方式转化为单目标优化目标时，由于每个优化量具有不同的物理含义，不同参数单位，因此是需要进行归一化处理的，从分层强化学习奖励函数设计中可以看到，因此要求各自网络链路参数的权重 $\beta_1,\beta_2,\beta_3,\beta_4,\beta_5\in[0,1]$，而 u_{bw}，u_{delay}、u_{loss}、u_{drop} 和 u_{dist} 对应网络链路也需要进行归一化的数据处理才有可比性。

4 分层强化学习的 SDWN 跨域链路故障感知自适应路由架构

在本节中，重点介绍基于深度分层强化学习的 SDWN 跨域链路故障感知路由方法的体系结构。SDWN 跨域路由的总体架构由根控制器模块、本地控制器模块、网络链路数据采集模块、网络流量预测模块、跨域路由模块组成，每个部分的结构如图 1 所示。

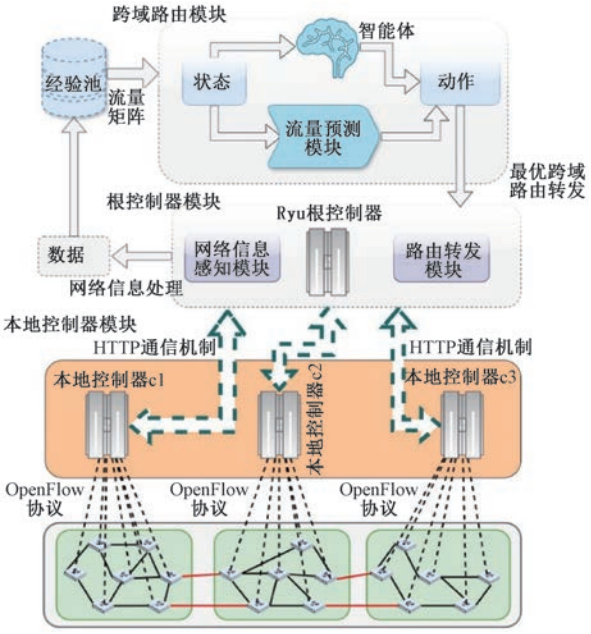


图 1 基于分层强化学习的 SDWN 跨域路由架构

4.1 根控制器

在本文中，根控制器模块通过 HTTP 协议与各个本地控制器进行高效的信息交换[42]。如图 2 所示，将本地控制器作为客户端，负责向根控制器发送域内或域间拓扑信息以及链路信息的请求；而根控制器则为服务端，负责处理这些请求并提供相应的服务。根控制器的服务端监听特定的网络端口，以便实时接收来自本地控制器的数据，数据传输格式为 JSON，其中封装在 data 的数据字段中包含网络

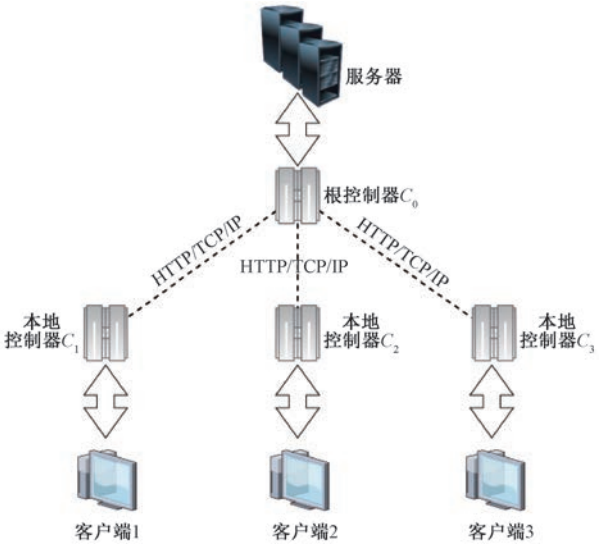


图 2 根控制器与本地控制器的通信方式

拓扑和网络链路信息。根控制器主要功能是获取全局的网络链路状态信息,并根据当前网络状态下发跨域路由模块所构造的路由。

4.2　本地控制器模块

在 SDWN 网络中,本地控制器通过链路发现协议(Link Layer Discovery Protocol, LLDP)[43]来获取网络中的拓扑信息,并根据获取的网络拓扑结构去周期性监测相关的网络链路信息。

在获取网络拓扑信息时,针对多域网络的情形,因而分为两种情况:域内网络拓扑信息和域间网络拓扑信息。在域内,本地控制器可以通过链路发现协议进行网络拓扑感知,当网络设备接收到链路发现协议时,会更新本地的邻居表或邻居关系数据库,并将相邻设备的信息进行记录,如邻居设备的 SSID、端口信息等。SDWN 控制器则通过查询网络设备的邻居表或邻居关系数据库来获取域内的网络拓扑信息。

在域间,通过 LLDP 报文是无法获取域间链路信息。因为对于不同域之间所发送的 LLDP 包,它只能被本域内的控制器识别,若 LLDP 报文来自其他子域,则本地域内控制器会默认丢弃。因此,本文在此基础上增加了一个域间链路感知模块。如图 3 所示为例,第一子域的本地控制器 C_1 通过 Packet-out 消息发送的 LLDP 报文到达 $ap2$ 边缘交换机端,$ap2$ 交换机则会向其他端口进行多播,当 $ap1$ 和 $ap3$ 收到 LLDP 报文后将被记录到边缘交换机 $ap2$ 的邻居表中。当第二子域的边缘交换机 $ap5$ 接收到 $ap2$ 的 LLDP 报文时,因为本地控制器 C_2 中没有相关的流表项与之匹配,故让 $ap5$ 自动通过 Packet-in 消息将该 LLDP 报文发送给本地控制器 C_2,本地控制器 C_2 检测到发送 LLDP 报文的交换机标识为 $ap2$,而在第二子域中并未管理该交换机,因此将 $ap2$ 到 $ap5$ 这条链路视为域间链路。

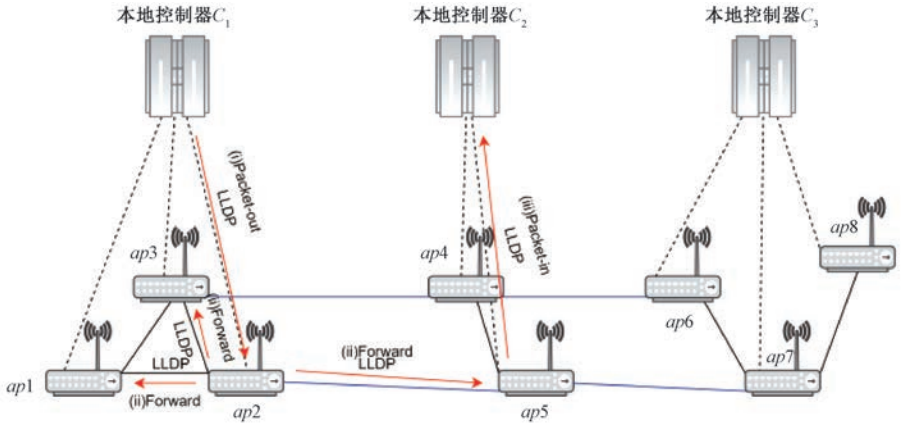


图 3　域间链路感知示意图

本地控制器主要负责获取域内和域间的网络拓扑以及相应的网络链路信息,并通过 HTTP 协议将数据发送到根控制器,同时接收来自根控制器下发的跨域路由流表。

4.3　网络链路数据采集模块

数据采集模块,主要是从 SDWN 的基础设施数据平面中周期性采集相关的网络数据,其中包括网络剩余带宽 bw、网络使用带宽 ubw、网络时延 $delay$、网络丢包率 $loss$、网络弃包数 $drop$ 以及无线 AP 交换机之间的距离 d 等参数。

与其他仅考虑单一网络状态信息不同的是,这些参数均作为本文算法的优化目标[44]。控制器通过周期性监测每个网络设备端口信息,并统计出每个端口发送字节数 tx_b、接收字节数 rx_b、发送数据包数 tx_p、接收数据包数 rx_p 以及生效时间 t_d。两次统计数据的差为已使用带宽 ubw,而剩余带宽为最大带宽 $bw_{cap\max}$ 与使用带宽差的绝对值,其相应的计算公式如下公式(14)~(15)给出。

$$bw_{ij} = | bw_{cap\max} - ubw_{ij} | \tag{14}$$

$$ubw_{ij} = \left| \frac{| (tx_{bj} + rx_{bj}) - (tx_{bi} + rx_{bi}) |}{t_{dj} - t_{di}} \right| \tag{15}$$

其中 $\langle i,j \rangle \in E_p$,$n$ 表示 AP 交换机节点的个数。

链路的往返时延 $delay$ 采用控制器自带的 Switches 模块数据来获取。首先测量 LLDP 数据发送的时间戳,具体方式是控制器通过 Packet/out 发送 LLDP 报文给交换机,然后交换机收到 LLDP 报文后将与流表信息进行匹对,接着交换机触发 Packet/in 给控制器发送 LLDP 数据包,控制器对数据包中的时间戳进行解析得到 T_{lldp_api} 和 T_{lldp_apj}。

其次控制器向交换机发送带有 echo/request 报文，然后控制器解析交换机返回的 echo/reply 报文，并用当下时间减去 data 部分解析的发送时间，从而得到控制器到无线交换机间的 echo 往返时延 T_{echo_api} 和 T_{echo_apj}，所以链路 e_{ij} 的时延计算公式如公式(16)所示。

$$delay_{ij}=\frac{(T_{lldp\_api}+T_{lldp\_apj}-T_{echo\_api}-T_{echo\_api})}{2} \tag{16}$$

公式(17)为链路丢包率 $loss_{ij}$，表示链路 $e_{(i,j)}$ 从交换机端口 i 到 j 和从交换机端口 j 到 i 两个方向计算的丢包率的最大值。公式(18)为链路包错误率 $drop_{ij}$ 表示链路 $e_{(i,j)}$ 从交换机端口 i 到 j 和从交换机端口 j 到 i 两个方向的弃包数。

$$loss_{ij}=\max\left(1-\frac{rx_{pj}}{tx_{pi}},1-\frac{rx_{pi}}{tx_{pj}}\right)\cdot 100\% \tag{17}$$

$$drop_{ij}=|\ rx_{pi}-tx_{pj}\ |\cdot 100\% \tag{18}$$

其中，tx_{bi} 表示从端口 i 的发送字节数，rx_{bi} 表示从端口 i 接收字节数。tx_{pi} 表示从端口 i 的发送数据包数，rx_{pi} 表示从端口 i 接收数据包数。

在实际物理场景中，距离对应了无线传输能量的消耗，因此在本文算法中还考虑了 AP 之间的距离。公式(19)为每个 AP 之间的距离 $dist_{ij}$，该距离表示无线交换机 AP 物理空间上的距离，由三维坐标计算公式得到。

$$dist_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2+(z_i-z_j)^2} \tag{19}$$

其中，(x_i,y_i,z_i) 表示第 i 个交换机之间的空间三维坐标。

4.4 流量预测模块

本研究构建了一个综合性的流量预测模块，它由三个关键组件构成：Transformer 流量预测子模块、Ryu 周期性流量检测子模块以及 sFlow-RT 流量监控子模块。Ryu 子模块负责在周期 t 时间点监测网络流量，并分析链路状态；Transformer 子模块则利用历史数据预测未来的流量趋势；而 sFlow-RT 子模块专注于将网络拓扑和流量信息进行可视化展示。整合这些模块的流量预测系统能够有效预测流量高峰和潜在拥塞点，从而提前采取预防措施，避免网络故障，或在故障发生时迅速作出反应，减少网络波动，确保网络的稳定运行。

4.5 跨域路由模块

对多域 SDN 网络中，由于域与域之间的数据信息是分区域相连，整个路由过程可以分为域内路由路径和域间路由路径，维度比较高的路由也可以分为域内维度和域间维度，解决这样维度比较高的问题正好是分层强化学习的优势。深度分层强化学习专门用于解决大规模问题的一种人工智能方法，通过将复杂任务进行分解成一个由较简单任务组成的层次结构子任务，从而降低整体任务的复杂度，能提高学习效率、探索能力和泛化能力。因此，设计的跨域路由模块在根控制器设计基于 AC 网络架构的分层强化学习 DHRL 机制将跨域路由问题分解成两个子问题，即域间节点选择问题和构造从该当前节点抵达子目的节点的最优路径选择问题，设计包括预测后的网络流量矩阵和故障信息矩阵在内的 DHRL 环境，设计问题相关的分层学习子目标集合和不同维度的内外层控制器动作机制来调整智能体的学习轨迹，设计问题相关的内外层控制器奖励函数确定转发的最佳邻近域和选择适当的域间入口点，规避网络故障链路的影响，构建出高效智能链路故障感知的自适应跨域路由。这样通过引入深度分层强化学习方法就能够有效应对多域网络中链路失效的问题，并求解出性能更均衡的跨域链路故障感知自适应路由，以满足网络通信质量的整体要求。

5 SDWN 跨域链路故障感知路由算法

本节介绍图 1 跨域路由模块中基于 Transformer 的网络流量预测 SDWN 跨域链路故障感知路由功能的设计。

5.1 Transformer 网络流量预测方法

设计流量预测能提前为感知未来流量趋势和模式，这对于指导多域网络中的跨域路由决策提供重要依据，不仅能够增强决策的数据驱动性，还能提前识别网络中的潜在问题，例如链路拥塞和节点故障，为跨域路由提供故障预警。当系统侦测到这些异常时，能够迅速启动智能路由机制，通过动态调整传输路径避开故障链路，确保网络的持续连通性和数据传输的可靠性。

具体描述过程如图 4 所示，首先由本地控制器通过网络链路信息采集模块周期性获取网络流量数据，这些数据包括网络剩余带宽 bw、网络时延 $delay$、网络丢包率 $loss$、网络弃包数 $drop$ 以及无线 AP 之间的距离 d。然后将带有时间序列的流量数据进行归一化处理，并放缩到 [0,1] 的范围内。

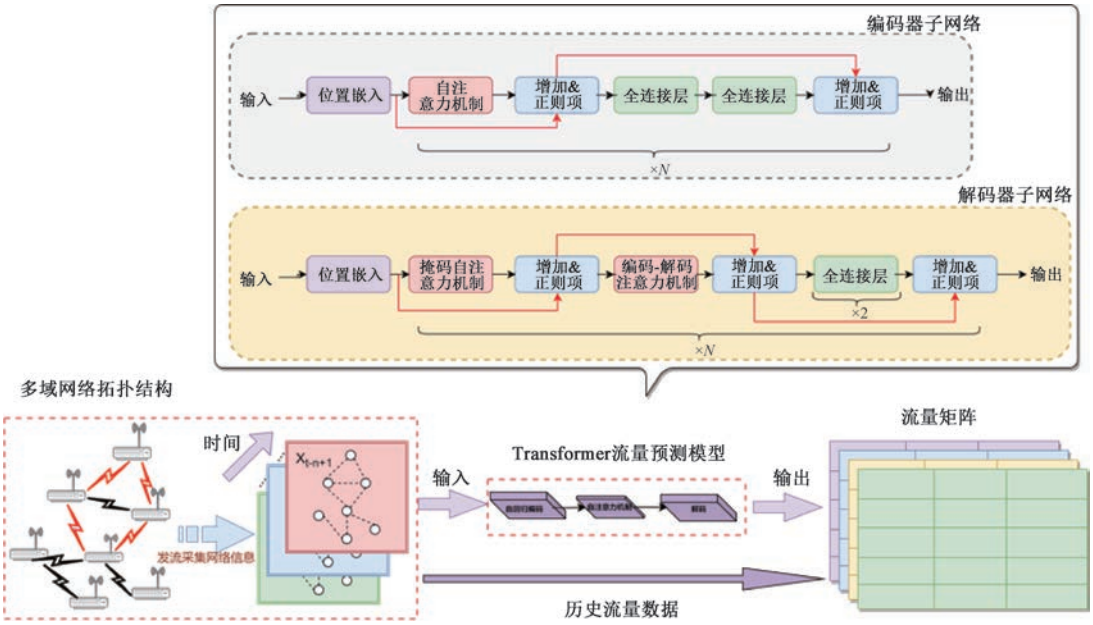


图 4　Transformer 预测网络架构

每条数据都会附上对应的链路标签,并作为 Transformer 网络架构的输入。Transformer 架构主要由输入层、编码层、解码层和输出层组成。输入层的核心是位置嵌入层,负责将数字转换为向量形式;编码层由 N 个编码层叠加而成,每个编码层包括多头注意力子层、规范化层、残差层和前馈全连接层;解码层则额外包含一个子层,用于实现掩码注意力机制和编码－解码注意力机制;输出层则由线性网络构成。Transformer 网络的最终输出,即预测的网络流量数据,连同历史流量数据一起,构成了分层强化学习中流量状态矩阵的一部分。

具体设计步骤如算法 1 所示,将学习率 lr、采样大小 $batch$、预测序列长度 ℓ、预测时间 t、训练总数 $\mathcal{M}$、历史网络数据序列等参数作为预测算法的输入,算法的输出为预测后的流量矩阵 $\mathcal{Y}=y[t+1],y[t+2],\cdots,y[t+\ell]$。第 1 行为随机初始化网络模型权重 θ,并设置位置编码、注意力机制等组件。第 2 行到第 13 行为预测模型训练的过程,其中第 2 行为算法迭代的循环次数,最大迭代次数为 $\mathcal{M}$,第 3 行和第 4 行为从 $pickle$ 数据集中每次获取 $batch$ 大小的数据参与训练,第 5 行将数据进行归一化处理,将其值放缩至 $[0,1]$ 之间,同时对数据进行维度处理,以满足预测网络训练的数据维度大小;第 6 行将数据输入预测网络得到预测值,第 7 行和第 8 行为计算损失函数并反向更新网络参数,在训练过程中以真实值和预测值的平方差来更新网络参数 θ',第 11 行为保存训练的模型权重,用于进行流量预测。

Transformer 网络流量预测算法输出的预测序列作为下一节分层强化学习算法输入中的一部分,同时可以通过预测流量得到故障链路矩阵。

算法 1. 基于 Transformer 网络流量预测算法

输入:学习率 lr;采样大小 $batch$;网络参数更新频率 $freq$;预测序列长度 ℓ;预测时间 t;训练代数 $\mathcal{M}$;输入数据的特征维度 $n\_encoder\_inputs$;数据嵌入特征维度 d_model;多头注意力模型头部数量 $nhead$;Transformer 块的个数 num_layer;历史网络数据序列 $\mathcal{X}=x[1],x[2],\cdots,x[t]$

输出:预测流量序列:$\mathcal{Y}=y[t+1],y[t+2],\cdots,y[t+l]$

1. 初始化模型参数、层数、头数等超参数 θ,同时定义位置编码、注意力机制、前馈网络等组件

2. FOR $episode = 1 \leftarrow \mathcal{M}$ DO:

3.　FOR $batch$ IN $data$:

4.　从 $graph$ 数据中获取历史流量数据 $\mathcal{X}=x[1],x[2],\cdots,x[t]$

5.　进行归一化处理:$\overline{\mathcal{X}}=\overline{x[i]}=\dfrac{(x[i]-\min(\mathcal{X}))}{\max(\mathcal{X})-\min(\mathcal{X})}$,其中,$\max(\mathcal{X})$ 和 $\min(\mathcal{X})$ 表示输入序列中最大和最小值。

6.　前向传播计算预测值 $\mathcal{Y}=transformer\_model(\overline{x[1]},\overline{x[2]},\cdots,\overline{x[t]})$

7.　计算损失函数 $loss=\sum_{i=1}^{\ell}(\mathcal{Y}_{\mathrm{pre}}[i]-\mathcal{Y}_{true}[i])^2$

8.　反向传播更新模型参数 θ'

9.　END FOR

10.　IF $episode \,\%\, freq == 0$ THEN:

11.　保存预测模型权重

12. END IF
13. END FOR

5.2 DRL-HAC 智能跨域路由算法设计

借助上一小节中设计的流量预测方法，本文设计的使用深度分层强化学习解决跨域路由问题的算法框架如图 5 所示，解决问题的关键在于设计适当的分层子目标，并确定强化学习的状态表示、内外层动作策略、内外智能体与环境的交互方式，以及相应奖励函数的设计。设计的外层与内层的控制器网络结构均为策略网络（policy network）和目标网络（target network），采用的是基于策略的 AC 网络框架以提升智能体训练的稳定性[45]。

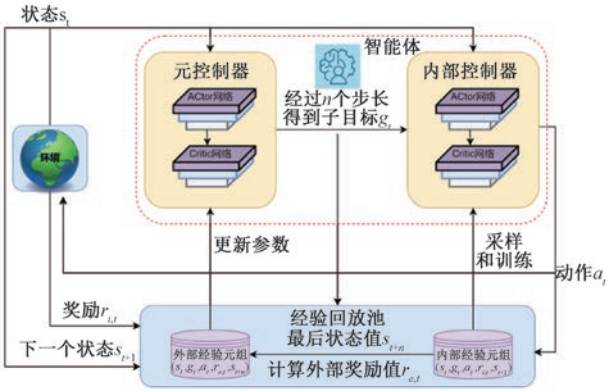


图 5 深度分层强化学习算法架构

首先外层的元控制器通过当前的网络状态 s_t，包含了每个网络子域控制器采集的网络拓扑和链路信息，其中链路信息包含当前网络的剩余带宽、网络时延、丢包率、弃包数以及无线 AP 之间的距离。然后经过 n 个训练步长判断当前的状态 s_t 是否为子目标 g_t 的状态，若为当前子目标状态则将该子目标状态作为内部控制器的临时目标状态，内部控制器根据该 s_t 和 g_t 选择下一跳节点作为动作 a_t。智能体与环境交互得到内部奖励值 $r_{i,t}$ 和下一个状态 s_{t+1}，将样本轨迹 $(s_t, g_t, r_{i,t}, s_{t+1})$ 存储到内部经验池当中，目标网络则从样本池中采集一个 batch 大小的数据进行训练，当达到 n 步长时，得到最终的状态 s_{t+n}，同时计算外部奖励值 $r_{e,t}$。将轨迹样本$(s_t, g_t, r_{e,t}, s_{t+n})$ 存入外部经验回收池，外部元控制器目标网络从中随机采样样本，以此更新外部元控制器网络中的参数，同时进行内外控制交互和迭代，直到找到所有的子目标和抵达最终目标状态，此时整个分层强化学习的智能体学习完成。

接下来详细介绍本文 SDWN 跨域智能路由算法中状态空间 $\mathcal{S}$、动作空间 $\mathcal{A}$ 和奖励函数 $\mathcal{R}$ 的设计，以及分层强化学习算法中损失函数的计算、网络更新策略和分层学习轨迹的处理方式。

5.2.1 状态空间 $\mathcal{S}$

设计的状态空间形式如公式(20)所示，其中 stack 为矩阵拼接函数，在同一维度上进行矩阵拼接。状态信息反映了智能体所感知到的环境信息，以及动作引起的变化。本文设计的状态空间 $\mathcal{S}$ 包含智能体位置状态矩阵 $\boldsymbol{M}_l$、网络流量信息矩阵 $\boldsymbol{M}_{\text{traffic}}$、链路故障矩阵 $\boldsymbol{M}_{\text{fault}}$、域信息矩阵 $\boldsymbol{M}_{\text{domain}}$。位置状态矩阵用于记录智能体所处位置和终止状态，网络流量信息矩阵用于计算智能体在环境交互过程中的奖励值，链路故障矩阵用于标识网络中出现故障的链路信息，指引智能体绕开故障链路，域信息矩阵用于统计域与域之间的边缘交换机及其连接关系，从而得到子目标集合。

$$S_t = stack(\boldsymbol{M}_l, \boldsymbol{M}_{\text{traffic}}, \boldsymbol{M}_{\text{fault}}, \boldsymbol{M}_{\text{domain}}) \quad (20)$$

位置状态矩阵 $\boldsymbol{M}_l$ 由一个大小为 $n \times n$ 的二维矩阵构成，可以由公式(21)所示，其中 m_{ij} 表示矩阵中第 i 行第 j 列的值，n 为无线 AP 节点个数，这里我们设置 a_{ij} 的值为-1、0、1、2 中的一个，由智能体当前状态所决定。

如图 6 所示，给定时刻的当前状态的位置状态矩阵 $\boldsymbol{M}_l$，位置矩阵 $\boldsymbol{M}_l$ 对角线上元素表示节点的对应编号信息，非对角线上元素表示边的信息。在对角线上，蓝色区域 src 表示源节点，对应矩阵对角线元素标识为 1，橙色区域 dst 表示目的节点，对应矩阵对角线元素标识为-1，绿色区域表示智能体走到经过的节点位置，对应矩阵元素标识为 1，即用 1 替换原有标识值，粉红色区域表示智能体抵达的节点位置为子目标的位置，对应矩阵元素标识为 2，子目标的选择及表示方式是在域信息矩阵中规定的，其余的矩阵对角元素标识为 0；对于非对角元素值，对应拓扑中对应给的边，本文初始化所有边的标识为 0，黄色区域则表示智能体经过的边，对应矩阵元素标识为 1，其他灰色区域对应矩阵元素为 0 表示智能体未经过相应的边。由以上的设计方式，位置矩阵 $\mathrm{M_l}$ 在智能体与环境交互过程中会一直依据所选动作和子目标而进行更新，直到抵达子目标和抵达最终目标状态后且算法达到收敛状态时，智能体完成整个路由的学习，对最终位置矩阵标识的路径进行解析就得到最终构造的跨域路由。

$$\boldsymbol{M}_l = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix}, m_{ij} \in \{-1, 0, 1, 2\} \quad (21)$$

图 6　$\boldsymbol{M}_t$ 位置状态矩阵

网络流量信息矩阵 $\boldsymbol{M}_{\text{traffic}}$ 则由网络剩余带宽 bw_{ij}、网络时延 $delay_{ij}$、网络丢包率 $loss_{ij}$、网络弃包数 $drop_{ij}$、无线 AP 之间的距离 $dist_{ij}$ 组成的多通道大小为 $n \times n$ 的二维矩阵。在矩阵中节点和边的关系函数由公式(22)给出,其中矩阵的行编号 i 和列编号 j 表示无线 AP 的 SSID 标识的编号,当 $i=j$ 时矩阵对角线的元素设置为空值,表示无效的数值,行号与列号则为对应网络中的节点;当 $i \neq j$ 时,则对应的元素表示为每条链路上的权重,其行号与列号则为对应的链路。如图 7 中的 $\boldsymbol{M}_{\text{traffic}}$ 矩阵所示,为了解决在 SDWN 中获取的网络链路数据对分层强化学习的神经网络模型训练产生不利影响,本文对流量矩阵进行了归一化处理,此处归一化方式与流量预测的处理方式一致,以限制数值范围,防止神经网络梯度爆炸现象的发生,提高算法模型的收敛速度,并减小对智能体搜索轨迹的影响。本文采用了 Min-Max 方法[46],将矩阵中的元素大小限制在指定范围 $[a,b]$ 内。具体地,归一化后的流量矩阵 $\overline{\boldsymbol{M}_{\text{traffic}}}$ 由公式(23)所示。

$$f(e_{ij},n_{ij})=\begin{cases}e_{ij}, i \neq j\\ n_{ij}, i=j\end{cases} \tag{22}$$

其中,$i,j \leqslant n$;$e_{ij}=e_{ji} \in E$

$$\overline{\boldsymbol{M}_{\text{traffic}\,ij}}=a+\frac{m_{ij}-\min(\boldsymbol{M}_{\text{traffic}})\cdot(b-a)}{\max(\boldsymbol{M}_{\text{traffic}})-\min(\boldsymbol{M}_{\text{traffic}})+\varepsilon} \tag{23}$$

其中,$a,b \in [0,1]$,$\varepsilon=1e^{-6}$,m_{ij} 表示流量矩阵中的元素,$\min(\boldsymbol{M}_{\text{traffic}})$ 和 $\max(\boldsymbol{M}_{\text{traffic}})$ 分别表示矩阵中的最小和最大值,ε 是一个很小的数值,用于避免分母为零的计算错误。a 和 b 为归一化后的上下边界阈值。

如图 8 所示,网络链路故障矩阵 $\boldsymbol{M}_{\text{fault}}$ 是在网络拓扑的邻接矩阵基础上进行设计的,当故障检测到故障链路时,会不断更新故障链路矩阵。若存在故障边,则故障链路矩阵中的 1 将更新为 −1,即橙色区域均为故障链路。在分层强化学习过程中,智能体会学习故障链路矩阵,避免选到故障链路,在奖励函数中则会对选到故障链路的动作进行惩罚处理。

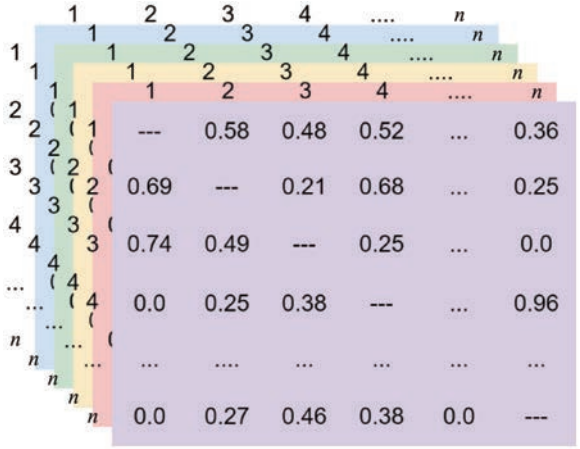


图 7　$\boldsymbol{M}_{\text{traffic}}$ 网络信息状态矩阵

图 8　$\boldsymbol{M}_{\text{fault}}$ 网络链路故障矩阵

域信息矩阵 $\boldsymbol{M}_{\text{domain}}$ 用于设定分层强化学习的子目标,指引智能体朝着完成子目标的方向学习。如图 9 所示,其中 D_i^k 的上标表示为第 k 子域,下标 i 表示边缘交换机的编号,D 表示为边缘交换机节点。相应的,$e_{i,n}^{1,k}$ 上标表示从第一子域到第 k 子域的域间链路,而下标则表示哪一条域间链路,e 为链路标识。通过域信息矩阵,可以得到每个域的边缘交换机节点信息和域间链路的连接关系。本文将每个域的边缘交换机所在的节点状态作为子目标,并将得到的子目标集合用于上层策略中进行训练。其中 t 时刻的子目标 g_t 与子域边缘交换机节点 D 的关系如公式(24)所示。

$$g_t=\{D_1^1,D_2^1,D_3^2,\cdots,D_t^k\},1 \leqslant k,t \leqslant n \tag{24}$$

5.2.2　动作空间 $\mathcal{A}$

在分层强化学习中,动作空间的设计与环境紧密相关,用于指引智能体所学习的方向。其中,涉及

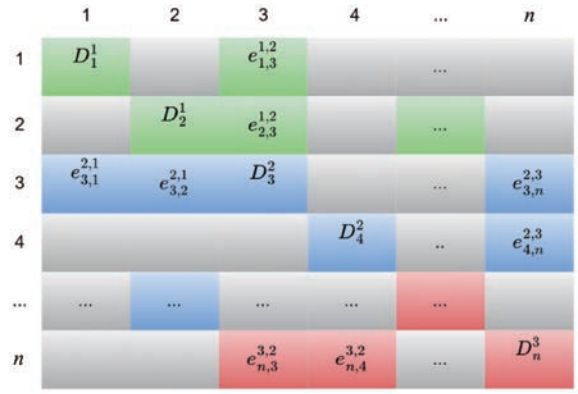


图 9 $\boldsymbol{M}_{\text{domain}}$ 域信息矩阵

外层的元控制器动作空间设计和内层控制器的动作空间设计。外层动作空间为所有节点的集合 $\mathcal{A}_{\text{ex}}=V=\{a_1,a_2,\cdots,a_i,a_{n-m},a_n\}\in\{N_1,N_2,\cdots,N_i,N_{n-m},N_n\}$，其中 m 表示无效动作个数，n 为图 G 中的节点个数，即无线 AP 交换机的个数。智能体每次决策时，将初始状态 s_t 输入策略网络中得到相应的动作 a_t，并根据动作去判断智能体是否抵达子目标状态，若为子目标状态则将其传递到内部控制器作为预设目标。外层动作空间的设计主要涉及两种情况以及相应的处理方案。

首先对当前状态进行检测，以确定是否为子目标状态：

(1)如果当前状态是子目标状态，则将状态空间中位置状态矩阵 M_l 中相应的节点位置元素标识为 2，表示该节点为子目标。同时，将该状态矩阵 $\boldsymbol{s}_t^{\boldsymbol{g}}$ 和子目标 $\boldsymbol{g}_t$ 传递到内部控制器中进行迭代训练。

(2)如果当前状态不是子目标状态，则保持初始状态不变，也不给予任何惩罚或奖励。然后，通过更新 n 个步长，选择新的动作，继续寻找子目标状态。在此过程中，仅进行外部控制器的迭代。

内部动作空间 $\mathcal{A}_{\text{in}}=EN_i=\{a_{ij_1},a_{ij_2},\cdots,a_{ij_l}\}$，其中 EN_i 为当前节点 N_i 的邻接边集合，$N_i\in V$，l 为图 G 中最大的度，a_{ij_l} 为可选的动作。在图 10 中，通过一个实例来介绍内部控制器如何确定动作空间的大小。

如图 10 所示，图中最大的度 l 为 4，即内部动作最大的维数为 4。对源节点 1 而言，它的动作集合为 $\mathcal{A}_{\text{in}}^1=\{a_1,a_2,a_3,a_4\}=\{None,None,2,None\}$，其中 $None$ 为无效动作集。对于节点 3 而言，它的动作集合为 $\mathcal{A}_{\text{in}}^3=\{a_1,a_2,a_3,a_4\}=\{2,4,5,6\}$。内部空间相比于 DRL-PPONSA 动作空间的设计，本文采用图 G 中最大的度来确定动作空间的维度，从而大大减少了因为网络拓扑规模太大从而导致动作空间维数太大的问题。接下来详细介绍内部控制器与环境交互后动作选择的情况，主要分为以下五种情况进行讨论，如图 11 所示。

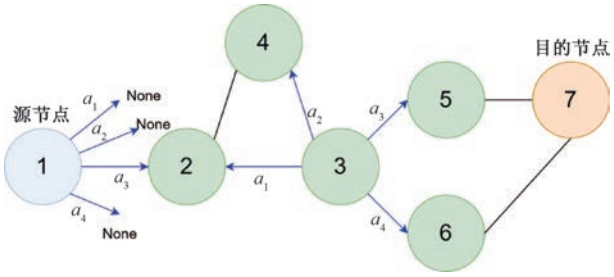


图 10 内部动作集合维数示意图

(1)智能体选择的动作属于邻接节点，且为未选过的节点，则更新当前状态 S_{t+1}，给予正向奖励。

(2)智能体选择的动作属于邻接节点，但之前已选过该节点，则进行环路标记，给予环路惩罚。

(3)当所选的节点不属于邻接节点，即视为无效动作 $None$，则不改变当前状态，并给予无效惩罚。

(4)当所选节点为邻接节点，且未形成环路，但该节点为故障边的节点，则不改变当前状态，并给予故障惩罚。

(5)智能体选到目的节点，则将当前状态设置为 $None$，并给予终点奖励。

5.2.3 奖励函数 $\mathcal{R}$

奖励函数用于激励智能体朝着目标方向进行学习，在本文算法设计中，智能体始终朝着最大奖励值的方向学习，且算法迭代到一定次数后能够达到收敛的状态，以此寻找合适的跨域路由，并满足多目标优化路由的需求。同样地，为了更好指引智能体学习，奖励函数也分成外部奖励 $\mathcal{R}_{\text{ex}}$ 和内部奖励 $\mathcal{R}_{\text{in}}$。

外部奖励函数 $\mathcal{R}_{\text{ex}}$ 主要是指引外部的元控制器能够快速学习到最优的子目标状态，同时将累计奖励传递到内部控制器，激励智能体选出较优的跨域路由。外部奖励设置主要分为两种情况进行分析：

(1)若选到的动作节点为子目标节点，则给予正向奖励值 R_g，如公式(25)所示。

$$R_g=\zeta_g R_s \tag{25}$$

其中，ζ_g 为子目标正向激励因子，R_s 为标准奖励值。

(2)若选到的动作不是子目标节点，则不给予奖励值，此时的奖励值 $R_g=0$。

内部奖励函数 $\mathcal{R}_{\text{in}}$ 则根据智能体与环境交互后产生的累计奖励，指引智能体选择到较好的动作，同时避开网络故障和环路的情况，从而顺利抵达最终的目的。本文将内部奖励设置为以下五种情况，与每个

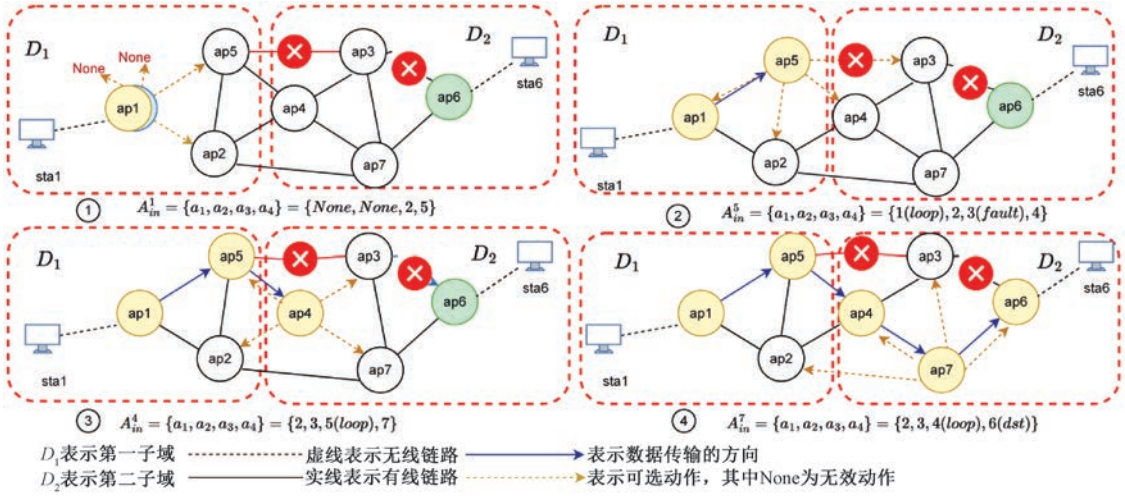


图 11　内部动作选择示意图

内部动作的情形一一对应,奖励函数设置的好坏,很大程度上决定了算法的收敛性能,如公式(26)所示。

(1)当智能体所选择的动作属于邻接节点,且从未选过该动作,也不是故障链路时,此时给予的正向奖励为 R_{link},即为链路奖励激励,其中 $\beta_1,\beta_2,\beta_3,\beta_4,\beta_5\in[0,1]$,是各自网络链路参数的权重,而 $\overline{bw}$,$\overline{delay},\cdots,\overline{dist}$ 为对应网络链路进行归一化后的值。

(2)当智能体选择的动作属于邻接边,且在之前路由矩阵中已标记该动作,视为环路情形,此时给予环路惩罚 R_{loop},其中 ζ_{loop} 为环路惩罚因子,R_s 为标准奖励值。

(3)当选择的动作属于邻居节点且在路由矩阵中并未标记该链路,但在故障矩阵中显示该链路为故障链路,故对该行为做出惩罚,故障惩罚奖励为 R_{fault},其中 ζ_{fault} 为故障惩罚因子。

(4)所选动作不属于邻居边时,则将该动作视为无效动作,并给予无效动作惩罚,其奖励函数为 $R_{\text{none}},\zeta_{\text{fnone}}$ 为无效惩罚因子。

(5)当所选动作属于目的节点时,将给予智能体一个终点正向奖励,其奖励函数为 $R_{\text{finish}},\zeta_{\text{finish}}$ 为终点奖励因子,C 为常数。

$$R_{\text{in}}=\begin{cases}R_{\text{link}}=\beta_1\overline{bw}(e_{ij})-\beta_2\overline{delay}(e_{ij})-\\\qquad\beta_3\overline{loss(e_{ij})}-\beta_4\overline{drop}(e_{ij})-\\\qquad\beta_5\overline{\text{dist}(e_{ij})}\\R_{\text{loop}}=-\zeta_{\text{loop}}R_s,\zeta_{\text{loop}}\in[0,1],R_s=1\\R_{\text{fault}}=-\zeta_{\text{fault}}R_s,\zeta_{\text{fault}}\in[0,1],R_s=1\\R_{\text{none}}=-\zeta_{\text{none}}R_{\text{link}},\zeta_{\text{none}}\in[0,1]\\R_{\text{finish}}=\zeta_{\text{finish}}R_{\text{link}},\zeta_{\text{finish}}=C\end{cases}\tag{26}$$

外部奖励和内部奖励的关系,当外部的元控制器选到子目标后,会将累计的外部奖励值传递到内部奖励中进行累加,从而得到总奖励值,智能体根据总奖励值函数去学习最优的跨域路径。总奖励函数如公式(27)所示,其中 P 表示路径,e_{ij} 则表示路径中的边。

$$R_{\text{total}}=\sum_{e_{ij}\in P}R_g(\text{e}_{ij})+R_{\text{in}}(\text{e}_{ij})\tag{27}$$

5.2.4　损失函数

深度分层强化学习中的策略网络参数 θ 和目标网络的参数 φ 的更新是通过最小化损失函数实现的,设计损失函数如公式(28)所示。

$$L(\theta,\varphi)=E[\min(w_t^{IS}*AF_t,clip(w_t^{IS},1-\varepsilon,1+\varepsilon)*AF_t)]+\mu*E[(V(s_t)-V(s_{t+1}))^2]\tag{28}$$

其中,AF_t 为优势函数,由公式(31)所示,R_t 为 t 时刻的奖励函数,γ 为给定的状态价值权重因子,$\boldsymbol{V}(s_t)$ 为当前状态价值函数,$\boldsymbol{V}(s_{t+1})$ 为下一状态价值函数。w_t^{IS} 为重要性采样的权重比率,由公式(30)所示,$\pi_\theta((a_t\mid s_t))$ 为策略网络更新后策略分布,$\pi_{\theta_{old}}((a_t\mid s_t))$ 为更新前的策略概率分布。$\text{clip}(w_t^{IS},1-\in,1+\in)$ 为梯度裁剪函数,由公式(29)所示,ε 为裁剪因子。重要性采样和梯度裁剪主要用于评估新旧策略的偏差程度,以此限制策略的改进,从而使算法收敛性更稳定。

$$clip_t(w_t^{IS},1-\varepsilon,1+\varepsilon)=\begin{cases}1-\varepsilon,w_t^{IS}\leqslant 1-\varepsilon\\1+\varepsilon,w_t^{IS}\geqslant 1-\varepsilon\\w_t^{IS},1-\varepsilon<w_t^{IS}<1+\varepsilon\end{cases}\tag{29}$$

$$w_t^{IS}=\frac{\pi_\theta(a_t\mid s_t)}{\pi_{\theta_{old}}(a_t\mid s_t)} \tag{30}$$

$$AF_t=R_t+\gamma V(s_{t+1})-V(s_t) \tag{31}$$

5.2.5 HAC 更新策略

该算法的策略更新公式通过最大化期望累积回报的上限,同时限制了策略更新的幅度,以此保证算法的收敛性和稳定性。HAC 算法的策略更新公式如(32)所示:

$$\theta_{t+1}=\mathrm{argmax}_\theta\mathbb{E}_t[\min(w_t^{IS}*AF^{\theta^k}(s_t,a_t),clip(w_t^{IS},1-\varepsilon,1+\varepsilon)*AF^{\theta^k}(s_t,a_t))] \tag{32}$$

其中,$\mathbb{E}_t$ 表示对 t 时刻的期望值,w_t^{IS} 为在 t 时刻新策略和旧策略的比率,而 $A^{\theta^k}(s_t,a_t)$ 为迭代 k 次后,执行动作 a_t 的优势评估函数,用于度量所执行策略的优劣程度,$clip(w_t^{IS},1-\in,1+\in)$ 则确保更新不会偏离旧策略太远。

5.2.6 分层强化学习轨迹处理(transitions)

为解决本文跨域路由使用分层策略遇到上层和下层策略不平稳的问题,同时对分层强化学习中奖励过于稀疏导致无法收敛的问题,本文提出了两种解决方案。

对于非平稳问题,首先将原样本 transitions 中的动作,即下层目标修正为智能体实际可以抵达的状态。此时的样本为下层策略达到最优状态所得到的样本数据,而最优的策略是独立于不断更新的下层策略。

设定指定的子目标集合作为上层策略对应 episode 最终实现的目标,同时应对不同情况设置不同的奖励方式,保证正负奖励达到均衡的状态,以此解决奖励函数稀疏的问题。

5.3 DHRL-ACTF 核心算法的描述

DHRL-ACTF 核心代码实现如算法 2 所示,在多域网络拓扑 G 中找到从 N_s 到 N_d 的最优的跨域路由。其中,将算法 1 输出的预测矩阵构成 $\boldsymbol{M}_{\text{traffic}}$ 网络信息状态矩阵,通过预测网络流量矩阵和图 G 得到网络故障矩阵 $\boldsymbol{M}_{\text{fault}}$,以此作为算法 2 输入的组成部分,此外算法的输入还包括当前智能体位置矩阵 $\boldsymbol{M}_l$、学习率 α、域信息矩阵 $\boldsymbol{M}_{\text{domain}}$、强化学习层数 k、最大子目标界限 H ;子目标测试频率 λ,最后是训练的代数 $\mathcal{M}$。

第 1 行初始化内外层控制器 Actor 及 Critic 的网络参数 θ_1,θ_2,同时把经验池进行置空处理,并设置源节点 N_s 和目的节点 N_d。第 2 行到第 24 行为一个迭代周期的训练,即智能体从源节点运动到目的节点,并输出智能体所决策的跨域路由作为一次迭代的周期。

第 4 行到第 5 行,获取强化学习初始状态 s_t,其中初始状态由位置信息矩阵 $\boldsymbol{M}_l$、网络链路流量矩阵 $\boldsymbol{M}_{\text{traffic}}$(矩阵信息包含剩余带宽、网络时延、丢包率、错包率、无线 AP 距离等链路参数)、网络故障矩阵 $\boldsymbol{M}_{\text{fault}}$、域信息矩阵 $\boldsymbol{M}_{\text{domain}}$ 组成。第 6 行,根据域信息矩阵设置分层强化学习的子目标 g_t,以引导智能体朝着阶段性子目标进行学习,并加快算法收敛速度。

第 7 到第 26 行为内外控制器的训练过程,其中第 8 行通过给定 H 期限,用于判断当前动作是否为子目标。第 9 行到第 10 行为外层控制器通过当前状态 s_t 和子目标 g_t 去获取行动的动作 a_t,如果判断 a_t 是子目标,同时进行子目标测试,进一步确认该动作为当前子目标,则进入内部控制器中进行训练,而外层控制器则进行累计外层奖励 $\mathcal{R}_{\text{ex}}$。

第 11 到第 16 行为内部控制器训练过程,其中第十二行为通过外层元控制器传递的子目标 g_t 与当前的状态 s_t 进行维度拼接得到内部控制器的初始状态 s_t',智能体在该阶段主要任务为从源节点寻找当前子目标最优跨域路由,即在多域场景来说,目的是寻找第一子域最优跨域路由。第 13 到 17 行为内部控制器根据 Actor 网络选择动作 a_t',并参与环境交互后得到内部累计奖励 $\mathcal{R}_{\text{in}}$,同时得到下一个状态 s_{t+1}',第 15 到第 16 行则将学习到的轨迹 $transition(s_t',a_t',R,s_{t+1}')$ 进行储存,用于更新内部网络参数。

第 18 到 20 行,当内部控制器中智能体抵达子目标状态后,退出内控制器循环,并将当前子目标状态作为外层元控制器的下一个状态,外层控制器根据该状态继续寻找下一个子目标。第 19 到 20 行,将外层控制器学到的轨迹存入经验池 E_1 中,用于计算外层元控制器的损失函数,以此更新外部的网络参数。第 21 到 26 行为判断智能体是否抵达最终的目的状态,以及结束外层元控制器循环的条件,最终使得智能体朝着最高累计奖励的方向学习,从而构造出更优的跨域路由。

算法 2. 基于 DHRL－ACTF 的 SDWN 跨域链路故障感知自适应路由算法

输入:智能体位置矩阵 $\boldsymbol{M}_l$;链路故障矩阵 $\boldsymbol{M}_{\text{fault}}$;域信息矩阵

$\boldsymbol{M}_{\text{domain}}$;层数 k;最大子目标界限 H;子目标测试频率 λ;学习率 α;训练代数 $\mathcal{M}$

输出:源到目的节点多域跨域路由 $P\_(s \rightarrow d)$

1.初始化内部控制器和元控制器的 Actor 及 Critic 网络权重 θ_1,θ_2;并清空经验缓存区 $\mathcal{B}$;设置源节点 N_s,目的节点 N_d

2. FOR $episode = 1 \leftarrow \mathcal{M}$ DO:

3.　FOR $\boldsymbol{M}_{\text{traffic}}$,$\boldsymbol{M}_{\text{fault}}$,$\boldsymbol{M}_{\text{domain}}$ IN 网络信息存储池 DO:

4.　　根据源节点-目的节点(N_s,N_d)重置网络环境得到智能体初始位置矩阵 $\boldsymbol{M}_t$

5.　　将矩阵 $\boldsymbol{M}_t$,$\boldsymbol{M}_{\text{traffic}}$,$\boldsymbol{M}_{\text{fault}}$,$\boldsymbol{M}_{\text{domain}}$ 进行同维度拼接得到强化学习的初始状态 s_t

6.　　根据域信息矩阵 $\boldsymbol{M}_{\text{domain}}$ 设置子目标 g_t

7.　　WHILE True DO:　　//　元控制器循环

8.　　FOR i IN $range(H)$:

9.　　$a_t \leftarrow \pi_a(s_t, g_t)$ // Actor 网络获取动作 a_t

10.　　IF 子目标 g_t 有效 THEN:

11.　　WHILE True DO: //　内部控制器循环

12.　　$s_t' \leftarrow train-stack(s_t, g_t)$

13.　　$a_t' \leftarrow train-\pi_a'(s_t', a_t)$

14.　　与环境交互得到内部奖励函数 $\mathcal{R}_{\text{in}}$ 和下一个状态 s_{t+1}'

15.　　内部控制器将内层强化学习轨迹 $transition(s_t', a_t', R, s_{t+1}')$ 存储到经验缓存区 E_2 中

16.　　从经验缓存区 E_2 中进行数据采样,从而计算损失函数,如公式(28)所示,并以此更新内部网络参数 $\theta_1' \leftarrow \theta_1$,$\theta_2' \leftarrow \theta_2$

17.　END FOR //终止寻找子目标状态循环

18.　当内部控制器智能体抵达子目标后的状态作为外层控制器的状态,即 $s_t \leftarrow s_{t+1} \leftarrow s_t'$

19.　$E_1 \leftarrow transition(s_t, a_t, \mathcal{R}_{\text{ex}}, s_{t+1}, g_t) \leftarrow learn$

20.　从经验池 E_1 中采样数据更新外部网络参数 $\tilde{\theta}_1 \leftarrow \theta_1$,$\tilde{\theta}_2 \leftarrow \theta_2$

21.　IF s_t' 为抵达子目标状态 THEN:

BREAK　//　跳出内部控制器循环

22.　ELSE　// 子目标无效

23.　继续寻找子目标状态 s_t^g,在 H 个界限内未找到则退出外层元控制器的循环

24.　IF s_t' 抵达最终目标 THEN:

BREAK // 终止所有循环

25.　END FOR

26.　END FOR

5.4　DHRL-ACTF 算法的复杂度分析

对设计的 DHRL-ACTF 算法复杂度,我们分别从流量预测和分层强化学习两个方面的算法复杂度分别进行分析。

(1)流量预测算法复杂度

流量预测时间复杂度主要由两个部分的时间开销组成,分别是输入嵌入层和 Transformer 编码器部分,具体的详情如下所示。输入嵌入层(Embedding Layer),将输入特征维度从 $n\_encoder\_inputs$ 映射到 d_model,单个时间步的复杂度:$O(n\_encoder\_inputs \times d\_model)$,加上 $batch$ 样本,序列长度 t 的时间开销后的时间复杂度为 $O(batch \times t \times n\_encoder\_inputs \times d\_model)$。另外一个是 Transformer 编码器的时间复杂度计算,每层包含自注意力和前馈网络,自注意力计算开销为 $O(t^2 \times d\_model)$,前馈网络时间开销为 $O(t \times d\_model^2)$,单层时间计算总复杂度为:$O(t^2 \times d\_model + t \times d\_model^2)$,同样考虑 batch 样本为 $O(batch \times num\_layer \times (t^2 \times d\_model + t \times d\_model^2))$,整体时间复杂度为 $O(batch \times [t \times n\_encoder\_inputs \times d\_model + num\_layer \times (t^2 \times d\_model + t \times d\_model^2)])$。

对于流量预测算法空间复杂度,模型参数所占用嵌入层参数:$O(n\_encoder\_inputs \times d\_model)$,每个 Transformer 层的参数:自注意力投影矩阵:$O(4 \times d\_model^2)$,前馈网络参数:$O(2 \times d\_model^2)$,num_layer 层总参数:$O(num\_layer \times d\_model^2)$,同时加上中间激活层之后的总的空间复杂度为:$O(batch \times t \times d\_model + batch \times num\_layer \times t \times (nhead \times t + d\_model))$。

(2) 分层强化学习算法复杂度

分层强化学习时间复杂度:外层循环(训练代数 M),每次训练需遍历网络信息存储池中的所有($M\_traffic$,M_fault,M_domain)组合,假设组合数量为 N。外层循环总时间复杂度为 $O(M \cdot N)$。在元控制器循环(子目标搜索)中,每次训练元控制器尝试最多 H 个子目标(最大子目标界限),每个子目标需执行内部控制器循环,假设内部控制器平均执行 T 步到达子目标(或失败),那么元控制器循环的时间复杂度为 $O(M \cdot N \cdot H \cdot T)$,其次计算内部控制器(子目标内路径探索)循环的时间复杂度,每步执行前向传播(*Actor* 网络)、环境交互、经验存储、采样更新。前向传播:假设 Actor 网络为 L 层全连接网络,每层维度为 D,复杂度为:$O(L \cdot D^2)$,经验采样与更新:从经验池 $E2$ 中采样 B 个样本,每次更新复杂度为 $O(B \cdot L \cdot D^2)$,单步内部循环复杂度:$O(L \cdot D^2 + B \cdot L \cdot D^2)$,总内部循环复杂度:$O(M \cdot N \cdot H \cdot T \cdot (L \cdot D^2 + B \cdot L \cdot D^2))$,外层

网络参数更新计算的时间复杂度与内部类似，但频率由子目标达成决定，假设每个子目标更新一次，复杂度为 $O(M \cdot N \cdot H \cdot B \cdot L \cdot D^2)$，那么整体的时间复杂度为 $O(M \cdot N \cdot H \cdot T \cdot (L \cdot D^2 + B \cdot L \cdot D^2))$，可进一步简化为 $O(M \cdot N \cdot H \cdot T \cdot B \cdot L \cdot D^2)$。

分层强化学习空间复杂度：网络参数存储 Actor-Critic 网络：每层参数量为 D^2，L 层网络总参数量为 $O(L \cdot D^2)$，本文所述算法是一种分层结构，即包含了元控制器和内部控制器各含 Actor/Critic 网络，总参数量为 $O(4L \cdot D^2)$，同时分层强化学习还包含了两个经验缓存区，内部经验池 E_2：存储 $T \cdot H$ 条轨迹，每条轨迹含状态、动作、奖励等，假设状态维度为 S，动作维度为 A，则单条轨迹空间：$O(S+A+1) \approx O(S)$，总空间：$O(H \cdot T \cdot S)$；外部经验池 E_1：存储子目标级经验，数量为 H，总空间：$O(H \cdot S)$，总经验池空间为 $O(H \cdot T \cdot S + H \cdot S) = O(H \cdot T \cdot S)$；而对于状态拼接与输入矩阵所占用的空间复杂度，拼接矩阵 $\boldsymbol{M}_l$，$\boldsymbol{M}_{\text{traffic}}$，$\boldsymbol{M}_{\text{fault}}$，$\boldsymbol{M}_{\text{domain}}$，假设每个矩阵维度为 $K \cdot K$，则状态空间：$O(K^2)$，因此整个分层强化学习的总空间复杂度为 $O(L \cdot D^2 + H \cdot T \cdot S + K^2)$。

6 实验设置与性能评估

本章节主要介绍仿真环境的参数设置、网络链路发流的方式以及网络故障节点的设置问题，然后分析预测结果和 DHRL 超参数的设置，最终给出本文算法 DHRL-ACTF 与 PPO、Q-learning、Dueling DQN、OSPF 和 BGP 的对比结果。

6.1 仿真环境设置

实验将多域网络搭建在操作系统为 Ubuntu 20.04.3 的服务器上，配置 GeForce RTX 3090 显卡用于加速神经网络计算。数据平面采用仿真平台 Mininet-Wi-Fi[47] 搭建模拟多域网络场景，同时采用 sFlow-RT 3.0 监测链路端口的流量状况，最后通过 iperf3 工具发流，并采集了 1000 个时刻的时间序列网络流量数据，采样时间间隔是 10s，并且将每个时刻流量数据存储为 pickle 文件。其中，模拟的数据流中 40%为域内流量，60%为域间流量数据，最后使用 python 3.8 和 pytorch 1.11.0 实现 SDWN 与强化学习的交互，使用的相关硬件和软件信息仿真如表 1 所示。

表 1 仿真工具

工具	版本号	功能
Mininet-Wi-Fi	2.3.1b	创建网络拓扑并设置网络参数
Ryu 控制器	4.3.4	流量监测和下发流表
GPU 显卡	GeForce RTX 3090	加速计算
Ubuntu 系统	18.04.6	实验系统环境
sFlow-RT 监控软件	3.0	流量监测和分析

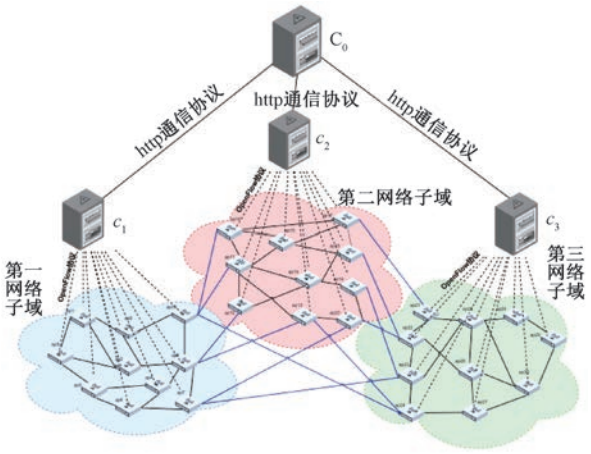


图 12 实验拓扑结构图

本实验通过随机增大域内或域间一部分链路的发流大小来模拟故障链路，在 t 时间内采集到网络链路带宽利用率为 0，并且出现较高的延迟时，视为故障链路。实验故障链路为随机设定，但设置最多不能同时出现 12 条故障链路，即将网络整体故障率控制在 10%左右。实验拓扑如图 12 所示，为模拟的无线物联网数据中心拓扑结构，共设置了三个网络子域，蓝色线表示域间链路，黑色则为域内链路。其中，蓝色区域为第一个网络子域，由控制器 c_1 进行控制，AP 的 SSID 标识从 1 到 10；红色区域为第二网络子域，由控制器 c_2 进行控制，AP 的 SSID 标识从 11 到 20；绿色区域为第三网络子域，由控制器 c_3 进行控制，AP 的 SSID 标识从 21 到 30。该多域网络共包含 30 个节点，118 条链路，每个无线接入点均有一个 STA 相连接，图中未对 STA 进行标注。为模拟真实的无线网络流量使用情况，我们使用 IPerf3[48] 工具编写生成流量信息的 Python 脚本。脚本通过客户端给服务端发送用户数据报协议(UDP)流量请求，采用随机选取客户端和服务端进行多目标发流的方式，并结合重力模型调整发流的大小，其中网络环境仿真参数如表 2 所示。将发流大小控制在 0～50Mbit/s 的范围内，保证流量信息在当天 10:00－15:00 时段达到峰值，其他时段缓慢变小。流量信息采集间隔为每 5s 采集一次，最后由

Ryu[49]控制器监控模块脚本生成 1000 个 30 个节点间的 40 个流量矩阵的图数据,然后将图数据写入 Pickle 文件中,矩阵中的元素包含剩余带宽、使用带宽、时延、丢包率、距离等链路信息,并使用 sFlow-RT[50]工具监控每秒的平均发流比特流量信息,结果如图 13 所示。

表 2　仿真参数

参数	值	参数	值
MAC 协议	IEEE 802.11 g	Frequency band	2.4 GHz
AP 数量	30(节点)	无线移动用户	30(节点)
CCA 阈值	−62 dBm	信道	1,6,12
传播损耗	Log-Distance	发射功率	21 dBm
接收增益	5 dBm	发射增益	5 dBm
信号范围	38～140 m	仿真区域面积	300 * 300 m^2
调制技术	OFDM	仿真时间	360 min
无线用户移动速度	2.5～4m/s	无线用户移动范围	38～140 m

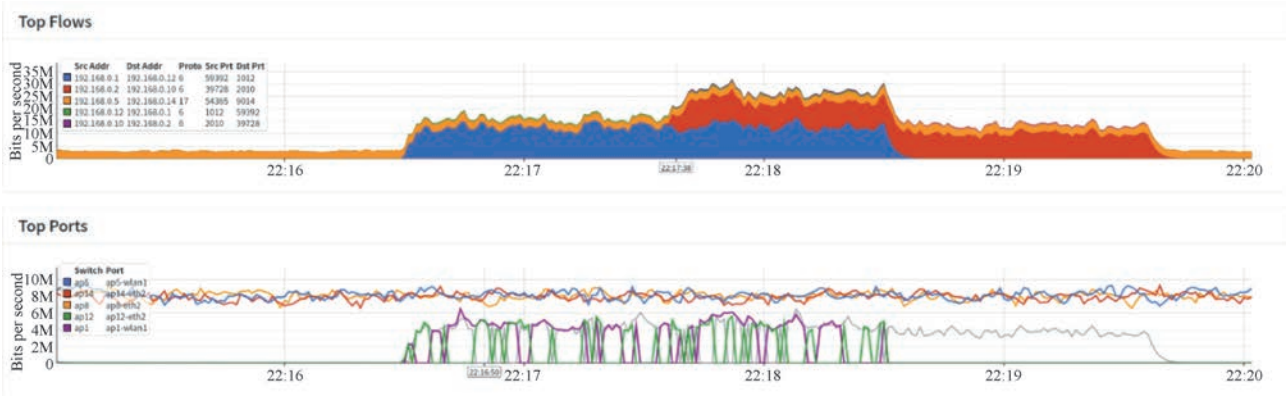


图 13　流量随时间变化的曲线

6.2　Transformer 流量预测性能对比及消融实验性能对比分析

采用 Transformer 网络模型去预测网络流量情况,能够在一定程度上提升路由算法的性能,同时在预测网络链路故障也能提供一定的数据支持。虽然本文主要工作不是研究流量预测问题,但分别进行了主要流量预测方法的性能对比,以及本文设计跨域路由方法是否使用流量预测方法的消融实验。

我们将采取的 1000 个采样点的网络流量数据对应的网络流量矩阵值进行了 0 到 1 之间的归一化处理,按时间顺序将这些网络流量数据划分为训练集和测试集,训练集包括 780 个采样点,用于模型学习流量变化规律,测试集包括 220 个采样点,通过对比测试集时段内的真实值与预测值来评估模型对未来网络流量动态变化(如突增、周期性波动)的预测能力,使用平均绝对误差(Mean Absolute Error, MAE)、均方误差(Mean Squared Error, MSE)和平均绝对百分比误差(Mean Absolute Percentage Error, MAPE)作为衡量模型预测准确度的指标,这三项指标都是越小越能体现模型的预测准确度越高。图 14(a)给出了主流网络流量预测模型的预测值与真实值的变化曲线,蓝色虚线表示原始网络链路的流量数据,所有值均采用归一化后的值进行对比,紫色线表示采用 Transformer 模型预测的结果,而黄色线则表示采用 GCN-GRU 模型预测的结果,蓝色实线表示采用 LSTM 预测的结果。可以看出,采用基于 Transformer 预测方法能更好反映出真实数据的变化趋势,这主要因为 Transformer 中多头注意力机制的加入,能够更准确地处理每个元素,有效捕捉时间序列中的长期依赖关系。图 14(b)给出了 MAE、MSE 和 MAPE 性能指标的对比结果,可以发现采用 Transformer 预测模型的均方误差 MSE 要比 GCN-GRU 和 LSTM 都要小,说明了采用 Transformer 模型预测的效果要比 GRU 和 LSTM 好。通过这样的预测机制可以对故障链路矩阵表示的流量趋势进行预测,和实际流量数据的趋势相同,如果真实的链路性能比较差则预测的链路性能也会比较差,从而可以保证设计的强化学习动作选择策略方法会避开链路性能比较差的链路。

接下来通过消融实验来分析设计的预测机制模型在分层强化学习中的具体表现,采用预测模型去提前感知网络状态和流量信息分布,能够帮助跨域路由算法做出更优化的决策。实验结果如图 15(a)和图 15(b)所示,可以看出增加了预测模型的分层强化学习方法,其奖励值要比未使用预测模型的要高,同时在寻找子目标和终点时所需的步长更短,这是由于一方面仅仅通过 SDN 网络测量机制一般很难获取到未知隐藏的网络流量状态,智能体通过预测模型能够获取更多未知的网络流量状态数据,另一方面使用预测机制能预测未来一段时间内网络流量状态的变化趋势,提前做出精准判断,能够让强化

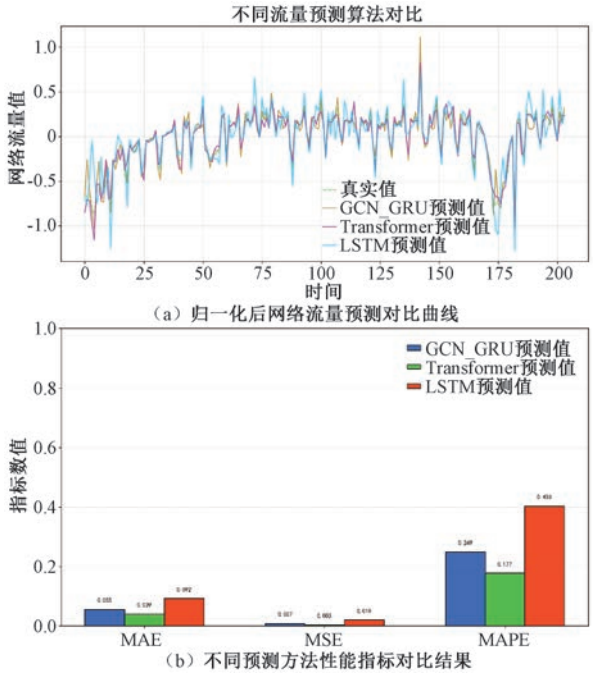


图 14 使用不同流量预测机制的性能对比分析

学习智能体探索到更大的状态空间,学习到更优的行为动作,从而获得更高的奖励值,不断优化所选择的路由,并绕开相关拥塞路径,从而降低网络发生故障的风险。由此可以验证了网络流量状态预测算法能够提升 DHRL-ACTF 智能路由算法的性能。

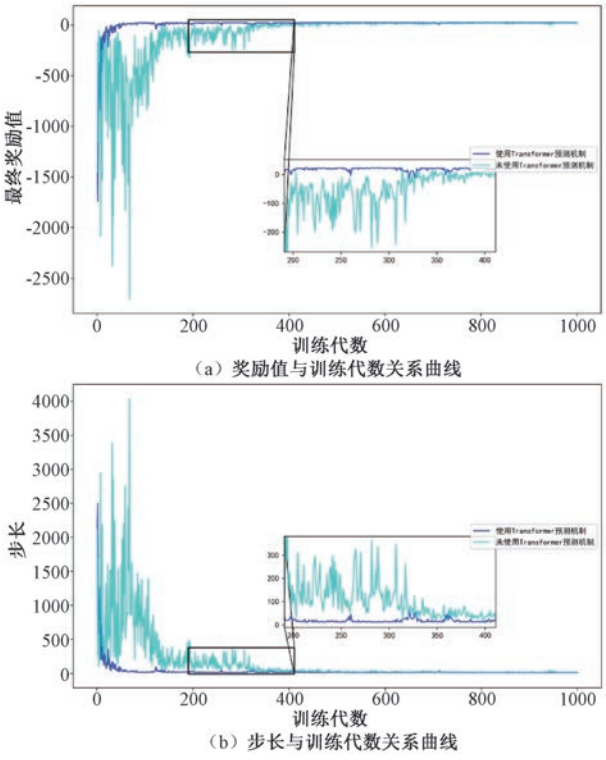


图 15 使用流量预测与未使用流量预测的消融实验性能对比分析

6.3 DHRL 参数设置与分析

本文设计的智能跨域路由算法是基于 AC 架构的分层强化学习方法,它是一种在线学习模式的策略优化算法,其中内外控制器由 Actor 和 Critic 网络构成,每个网络由两层卷积层和三层全连接层搭建,共同影响智能体的决策。智能体在进行训练过程中所设置的基本参数由表 3 给出。PPO 强化学习采用四层的网络结构,其中包含两层 Actor Network ,两层 Critic Network 。Actor Network 用于采样当前动作的概率以及下一个状态 S_{t+1} 的动作概率,Critic Network 则用于评价当前状态 S_t 的累计回报 R_t 和下一个状态 S_{t+1} 的累计回报 R_{t+1}。优化器使用自适应矩阵估计(Adam),它能够自适应学习率的更新,同时可以减少计算机内存的消耗,使智能路由算法的网络模型在训练过程中能够快速收敛。接下来,主要介绍内部控制器中网络参数调整对算法收敛性的影响。首先在第一子域设定一个随机的源节点,并在第二或第三子域设定一个随机的目的节点,用于确定智能体的起始位置和需要抵达的终点目标。其次,将每个子域的域边缘交换机作为子目标,引导智能体进行分层学习目标策略。高层元控制器负责将域边缘交换机抽象成子目标状态,而低层控制器则根据子目标状态执行具体的动作完成子任务。简而言之,高层智能体负责制定全局策略和目标,低层智能体负责执行局部任务和动作。

表 3 DHRL 参数设置

参数	符号	值
学习率	α_1,α_2	$1e^{-3},3e^{-3}$
批量大小	k	32
裁剪因子	ε	0.2
更新频率	f	10
经验池大小	B	3000
折扣率	γ	0.99
折扣因子	ξ_1,ξ_2	0.5,0.8
训练代数	M	1000

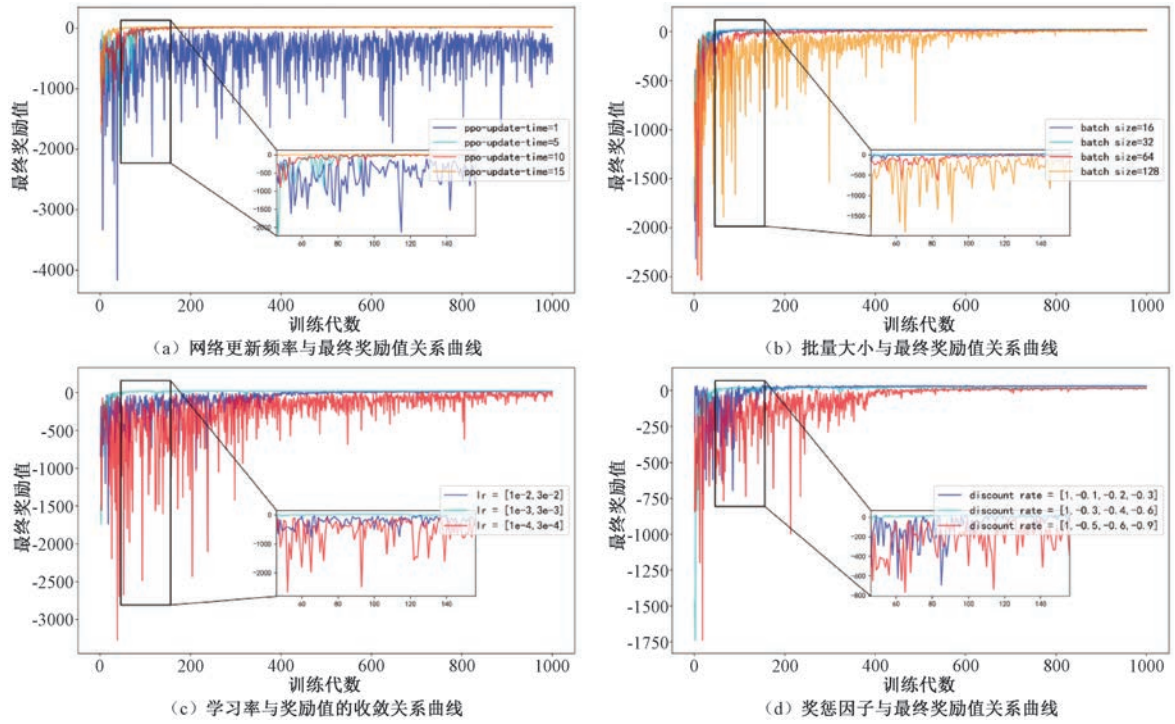
在分层强化学习过程中,设计合适的网络更新频率 f、批量大小 k、学习率 α、奖励函数奖惩因子 ζ 等参数对于智能体学习效果至关重要。在进行实验之前,需要对网络中的参数进行单一变量控制,即只更改需要变化的参数,其他参数保持不变。

如图 16(a)所示,设计不同的网络更新频率,探讨其对算法收敛性的影响。其中,横坐标表示训练所需的代数,纵坐标表示最终的奖励值。当内部网络更新频率 $f=15$ 时,智能体能够获得奖励值最大且收敛效果最好,从局部放大的图中可以看出,算法

在 50 多代左右就能够稳定收敛;当 $f=1$ 时,智能体决策路径中出现了大量环路,从而导致奖励值不稳定,算法也无法收敛到最大值。数据批量大小 k 一直是神经网络中的关键因素,对神经网络的训练过程有着重要的影响。

如图 16 (b)中,实验设置了四个不同的批量大小,分别为 16、32、64、128。其中,当 $k=32$ 时,算法收敛效果比较稳定,且收敛的速度更快。随着 k 的取值越大,奖励值波动越明显,当 $k=128$ 时智能体需要训练到 600 代左右才趋向于收敛。如图 16 (c)中给出了学习率对算法的影响,内部控制器中 Actor 网络的学习率 α 分别设置为 10^{-2}、10^{-3}、10^{-4},Critic 网络则分别设置为 3×10^{-2}、3×10^{-3}、3×10^{-4},当奖励值设置为 10^{-3} 和 3×10^{-3} 的组合时,算法能够快速收敛到最大值。另外,过大或较小的奖励值都会影响算法收敛的好坏。

通过大量实验测试后,选出三个具有代表性的参数进行分析,强化学习中奖励值设置的好坏,直接决定了智能体能否收敛到最大奖励值。本文中采用了组合奖励的方式,分为稀疏奖励、稠密奖励以及负奖励组成。将几种奖励模式组合在一起,综合考虑智能体行为和任务目标,以更准确地引导智能体学习的过程。在图 16 (d)中给出不同的奖惩因子 ζ,当 $\zeta_{\text{finish}}=1$,$\zeta_{\text{loop}}=-0.3$,$\zeta_{\text{none}}=-0.4$,$\zeta_{\text{fault}}=-0.6$ 时,算法收敛的效果更好,智能体能够在最快的时间内寻找到较优的跨域路由。对比 DRL-PPONSA 算法,设计的 DHRL-ACTF 算法平均收敛速度提升了 66.5%。



(a) 网络更新频率与最终奖励值关系曲线　(b) 批量大小与最终奖励值关系曲线

(c) 学习率与奖励值的收敛关系曲线　(d) 奖惩因子与最终奖励值关系曲线

图 16　DHRL 网络参数设置对比曲线

6.4　对比实验

接下来,将 DHRL-ACTF 跨域路由算法与 PPO、DQN、Q-learning 和两种经典路由方法 OSPF、BGP 的性能对比。本文主要使用网络平均瓶颈带宽 $\overline{bw}_{ij}$、网络平均时延 $\overline{delay}_{ij}$、网络平均丢包率 $\overline{loss}_{ij}$、网络平均弃包数 $\overline{drop}_{ij}$ 和无线接入点平均距离 $\overline{d}_{ij}$ 作为评价算法性能的重要指标。

公式(33)则给出对应网络平均瓶颈带宽、网络平均时延等指标的相关表示,其中 n_p 表示使用数据 pickle 的个数,一个 pickle 的数据量表示当前时刻 t 内所获得的全局网络流量。

$$\begin{aligned}
\overline{bw}_{ij}&=\frac{1}{n_p}\sum_{k=1}^{n_p}u_{bw}^{k},\\
\overline{delay}_{ij}&=\frac{1}{n_p}\sum_{k=1}^{n_p}u_{delay}^{k},\\
\overline{loss}_{ij}&=\frac{1}{n_p}\sum_{k=1}^{n_p}u_{loss}^{k},\\
\overline{drop}_{ij}&=\frac{1}{n_p}\sum_{k=1}^{n_p}u_{drop}^{k},\\
\overline{d}_{ij}&=\frac{1}{n_p}\sum_{k=1}^{n_p}u_{dist}^{k}
\end{aligned}\tag{33}$$

本文设计的实验中，统计一天的流量情况，每三个小时取一次平均值作为统计结果。在进行实验时需要设定优化函数每个权重的取值，其中 $\beta_1=0.7$、$\beta_2=0.3$、$\beta_3=0.3$、$\beta_4=0.1$、$\beta_5=0.1$。如图 17(a)是衡量跨域路由的平均链路剩余带宽，DHRL-ACTF 智能体所构建跨域路由的平均剩余带宽在大多数时刻下比其他算法要好，其中平均比 PPO、DQN、Q-learning、OSPF、BGP 算法分别高了 15.51%、37.88%、37.79%、38.31%、50.86%。该结果表明，DHRL-ACTF 在网络出现故障后进行重新路由能够决策出剩余带宽较大的路径，从而表明该算法能够将拥塞流量转移到更大的链路中。图 17(b)中衡量指标为从源节点到目的节点跨域路由后链路最小剩余带宽的平均值，即为平均瓶颈带宽。DHRL-ACTF 算法平均瓶颈带宽要平均比 PPO、DQN、Q-learning、OSPF、BGP 算法高 9.92%、20.47%、20.56%、19.43%、23.14%，这是由于 DHRL-ACTF 引入了基于流量预测的路径选择机制，利用 Transformer 架构实现 t-step 链路负载时序预测能有效构建预知的网络状态；而分层强化学习中的元控制器能基于预测结果动态规避潜在的拥塞链路，在策略网络训练中采用带宽加权奖励机制(奖励权重多出了约 40%)，能有效引导智能体选择高带宽路径。相比之下，对比的方法 DQN 和 Q-learning 方法因缺乏动态链路感知能力容易陷入局部最优，而且 OSPF 和 BGP 方法区别于本文设计方法 DHRL-ACTF，它们采用静态链路权重策略对动态适应能力不足，因而设计方法 DHRL-ACTF 在链路带宽性能上能获得更好的表现。

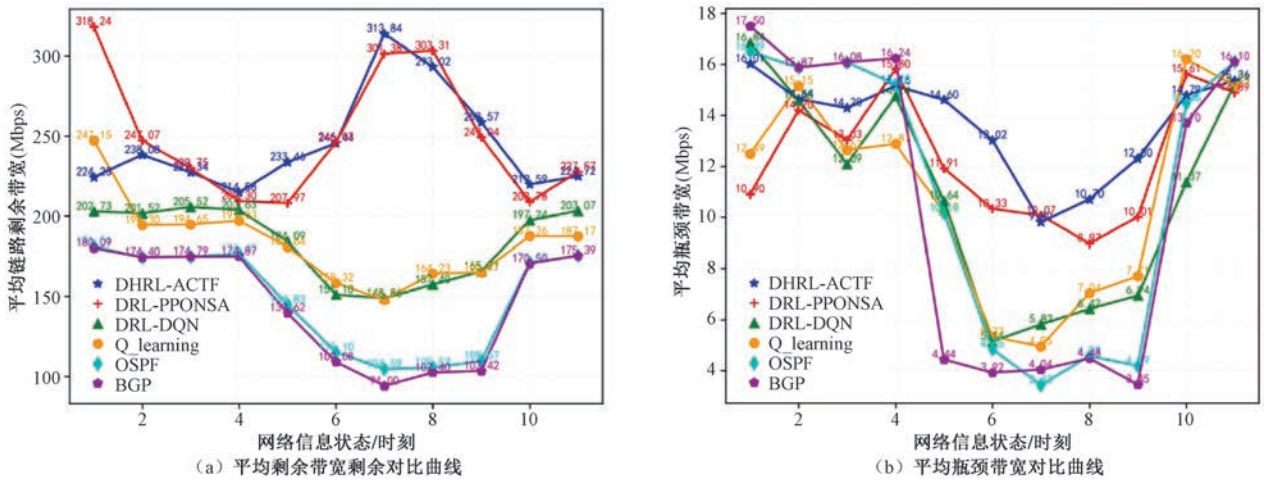


图 17 平均剩余带宽与瓶颈带宽结果分析

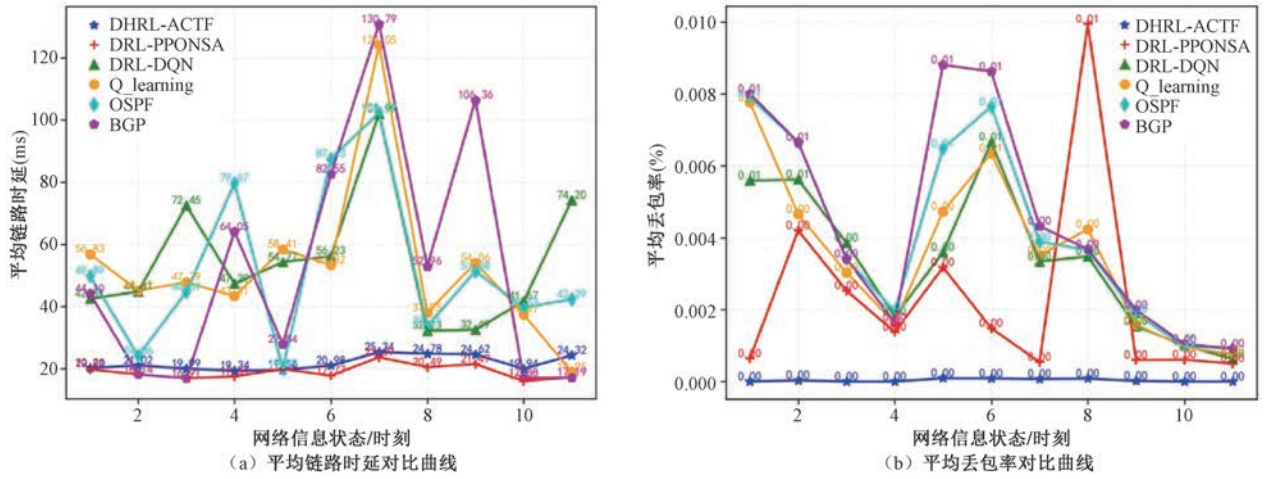


图 18 平均链路时延和丢包率结果分析

图 18(a)中衡量的是网络链路的平均时延。DHRL-ACTF 算法的平均时延要平均比 DQN、Q-learning、OSPF、BGP 分别低 58.02%、58.41%、66.56%、68.02%，但比 PPO 高 1.73%。从数据中

可以看出本文设计的跨域路由方法,有着不错的低时延性能。图 18(b)中衡量的是网络链路的平均丢包率情况。DHRL-ACTF 算法平均比 PPO、DQN、Q-learning、OSPF、BGP 分别低 46.64%、63.27%、65.22%、70.29%、72.26%,这是由于设计方法 DHRL-ACTF 构建了层次化策略的分解框架,可以将时延权重 β_2 从 0.1 动态调整至 0.3 实现动态协同优化;而且,元控制器能基于预测结果动态规避潜在拥塞链路,可以有效避免拥塞路径,进而选择到低时延、高可靠链路进行传输,从而可以降低链路的时延和丢包率。这些实验数据及结果表明,所提算法也有着较低丢包率性能。

图 19(a)中衡量的是网络链路平均丢包数,弃包数反映了网络链路因为发生故障或拥塞而造成数据包丢失的情况。DHRL-ACTF 算法平均比 PPO、DQN、Q-learning、OSPF、BGP 低 70.72%、92.26%、91.81%、91.64%、94.67%。由此可知,BGP 在跨域路由后,网络数据包丢失更加严重,而 DHRL-ACTF 则丢失的数据包更少。图 19(b)衡量的是跨域路由的链路长度情况。DHRL-ACTF 构造的链路长度平均要比 PPO、DQN 分别小 16.52%、13.63%,比 Q-learning、OSPF、BGP 算法要分别长 1.74%、11.67%、10.48%,通过前面关于时延、丢包率和弃包率等指标的实验结果可以发现,10%的路径长度增加可以换来 65%的带宽增益及 72.3%的故障切换代价降低,路径震荡频次较 OSPF 减少了 83.6%,显著提升了网络稳定性,这其实是由于构建带宽权重优于时延与丢包率、次优于弃包率及链路长度的设计机制带来的优势。链路长度在权重设置时,作为最后一个约束条件,其权重 β_5 仅为 0.1,而剩余带宽的权重 β_1 为 0.7。从数据中得出,虽然所提算法在链路长度上比传统最短距离算法要长一些,但 DHRL-ACTF 算法却有着不错的剩余带宽、低时延以及低丢包率等网络性能,所构造的跨域路径性能更加均衡。通过这样的实验结果也可以判断,在不同网络应用场景下(比如交互实时性优先场景,或者视频传输质量相关的可靠性优先的场景),只需要调整设计优化模型的不同优化目标对应的权值取值即可。

为了进一步验证 DHRL-ACTF 算法的有效性和稳定性,本文还进行了如下五种不同场景下性能测试实验以进行闭环验证,五种不同传输场景分别设置如下:

(1)正常流量场景:我们采用常用的一种重要分布——泊松分布来模拟分析正常网络流量场景,泊松分布通常可用于网络带宽预测、网站流量预测、网络流量优化和网络安全分析,这里我们先将网络流量设置了链路故障率为 0 时服从 $\lambda=50$ 的泊松分布 $P(\lambda=50)$ 来生成流量矩阵中 $\boldsymbol{M}_{\text{traffic}}$ 的正常流量场景,此时不考虑有链路故障的发生;

(2)突发流量场景:突发流量是指在某个比如在节假日等特殊时间或发生网络攻击等特定事件的时间段内,网络流量超过了平常水平突然增加,此时网络流量在短时间内急剧增加。对此,我们通过在 10%比例训练步长内注入高斯脉冲型 $N(\mu=150,\sigma=30)$ 的流量来进行模拟,峰值带宽占比设为 80%;

(3)动态链路故障场景:网络故障包括了多种,由于本文标题和研究内容都限定在链路故障的场景下(节点故障属于另一种网络故障,需要用另一个方法来表示和研究,我们将节点故障的研究放在了今后的研究课题中),我们通过随机断开 10%链路(即将故障链路矩阵中的 1 更新为−1)的方式进行链路

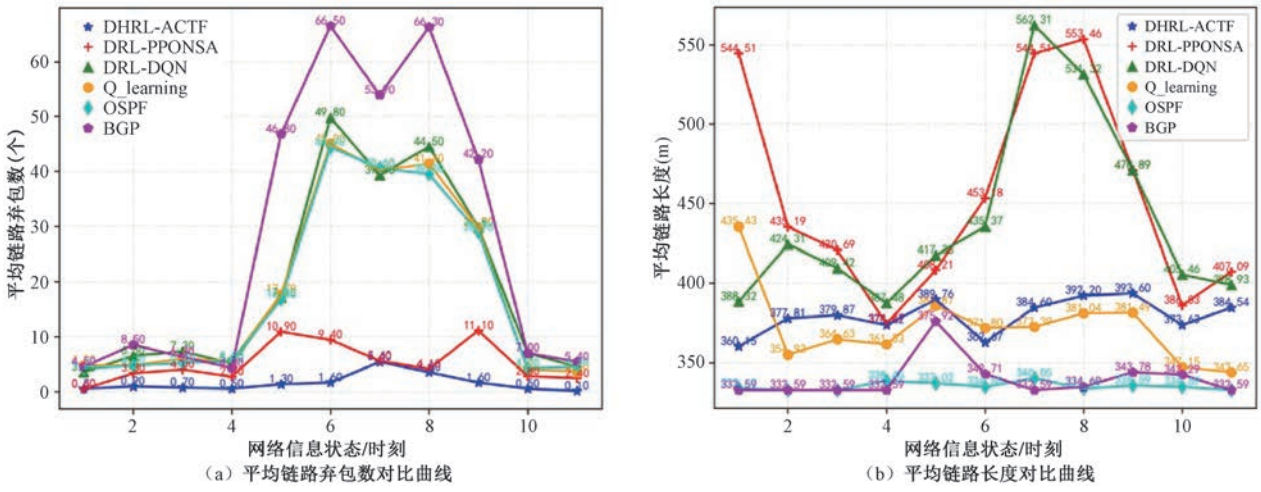


图 19　平均链路弃包数和链路长度结果分析

故障产生的模拟，其中故障间隔时间服从指数分布 $\exp(\beta=50)$；

(4)跨域路由流量存在限制的场景：考虑到域间链路带宽小于域内带宽的情形，考虑通过 $\boldsymbol{M}_{\text{domain}}$ 划分了一定数量的自治域后，可以通过设定域间链路带宽为域内带宽的 50%来进行模拟；

(5)高负载场景：流量高负载场景是指网络流量超过实际系统处理能力，对此可以通过设定背景流量恒定占用 80%链路容量并叠加组合使用(2)中突发流量方式来模拟测试此时极限负载时的性能。

以上每种具体传输场景参数及其设置方式如下表 4 所示：

表 4 不同传输场景相关参数及设置

场景名称	$\boldsymbol{M}_{\text{fault}}$	$\boldsymbol{M}_{\text{traffic}}$	$\boldsymbol{M}_{\text{domain}}$	其余相关参数
正常流量	全 0 矩阵	泊松分布 P(λ = 50)	多域	λ = 50
突发流量	1%故障链路矩阵元素为 1(指数分布间隔)	高斯脉冲 $\mathcal{N}(150,30)$，峰值带宽占比设为 80%	多域	峰值时长占比 10%
动态故障	10%故障链路矩阵元素为 1(指数分布间隔)	泊松分布 $P(\lambda=50)$	多域	$\beta=50$
跨域路由流量限制	1%故障链路矩阵元素为 1(指数分布间隔)	泊松分布 $P(\lambda=50)$	多域，域间带宽限制为原来的 50%	域间延迟惩罚系数 0.5
高负载场景	5%故障链路矩阵元素为 1(指数分布间隔)	正常流量场景下增加所有流量的 80%＋突发高斯脉冲	多域	叠加的突发峰值大小为 3 倍的正常流量值

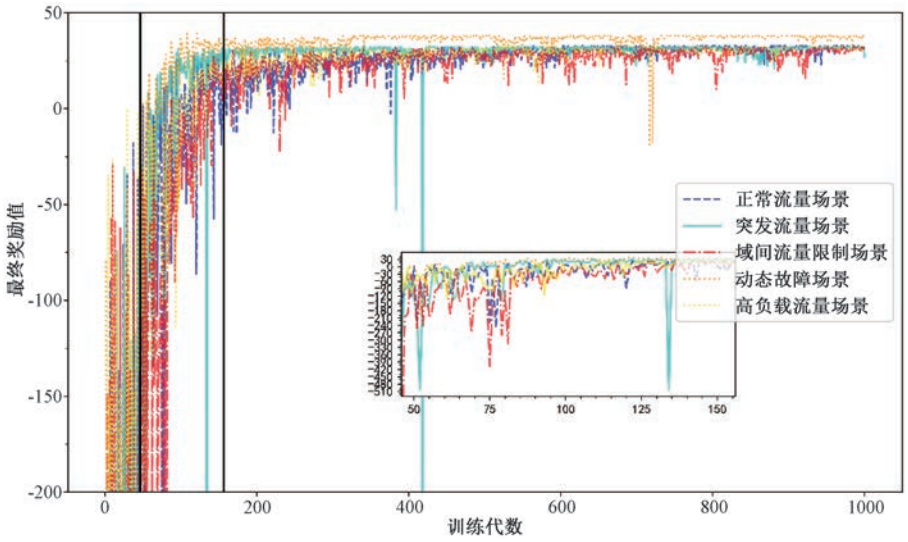


图 20 不同传输场景对算法有效性和稳定性的影响

为保证和前面实验结果的一致性，我们统一将这些不同实验场景下的网络剩余带宽、时延、丢包率、弃包数、无线距离对应权重因子统一设置为 $\beta_1=0.7$、$\beta_2=0.3$、$\beta_3=0.3$、$\beta_4=0.1$、$\beta_5=0.1$[51]，图 20 为在以上五种场景下得到的实验结果。从图中可以看出在各个传输场景下的奖励值变化具备如下特点：

(1)初始阶段：所有传输场景下的最终奖励值都比较低，且波动较大。这表明在训练初期，模型的表现不稳定，需要更多的训练来优化。随着训练代数的增加，最终奖励值逐渐上升，并趋于稳定。在大约 200 代之后，所有场景的奖励值都达到了较高的水平，并且波动逐渐减小。这表明模型在经过一定数量的训练后，性能得到了显著提升。

(2)中间阶段：随着训练代数的进一步增加，最终奖励值继续上升，并在大约 400 代时达到稳定状态。此时，所有场景的奖励值都维持在较高的水平，且波动较小。这表明模型在经过充分训练后，已经达到了较好的稳定状态。

(3)稳定阶段：所有场景的最终奖励值都维持在较高的水平，并且波动较小。这表明模型在经过充分训练后，能够在不同的流量场景下保持较高的稳定性和有效性。

(4)在突发流量场景和高负载流量场景中，虽然在大约 50 代和 400 代时出现了明显的奖励值下降，但经过一段时间的调整后，奖励值又恢复到了较高的水平。这表明所提方法在面对突发流量和高负载流量时，虽然会受到一定程度的影响，但能够迅速恢

复到较高的稳定状态。

这些特点表明随着训练轮次的增加,在各个传输场景下的模型累积奖励值均呈现显著上升趋势,且在经历充分训练后最终收敛于稳定状态。这样的结果表明设计的路由规划方法在不同网络环境下都表现出良好的收敛特性,能够通过迭代优化逐步逼近最优策略。

7 总　结

本文在 SDWN 框架上提出了一种基于分层强化学习算法所构造智能路由的方法 DHRL-ACTF,同时加入了 Transformer 预测模型去探索网络中未知的流量趋势等模式信息,使控制器能够实时感知网络态势,实现了对 SDWN 网络的智能路由决策控制。在动态拓扑变化的 SDWN 网络环境中,DHRL-ACTF 使用基于策略梯度更新的 PPO 分层强化学习方法,将高维的网络路由问题分解成两个子问题,即域间节点选择和域内节点选择的子问题。并且在网络参数变化时智能体仍然可以智能化地调整路由进行流表下发。由实验结果可知 DHRL-ACTF 算法在不同网络状态下以及随机设定的链路故障中有着不错的网络性能,平均指标大多数比其他算法的更优。相比于单一优化目标算法,DHRL-ACTF 跨域路由算法所构造的路径性能更加均衡。SDWN 架构的特点和分层强化学习的机制能保证设计的路由规划方法对网络状态高速动态变化的适应和无线网络规模的适用。此外,本文算法虽然解决了网络链路中出现节点失效(节点减少)或链路失效的难题,即将节点失效视为无效动作,但对于节点增加的情况目前还没有对应的解决方案,未来打算采用图卷积神经网络能够适应不同网络拓扑结构的特点,以此作为设计图深度强化学习状态空间的基础,并根据邻接矩阵来设计动作空间,以减少无效动作的选择。为了提高设计方法在不同数据场景下的泛化性能,可以进行自动参数调参的方式以达到预期收敛效果,对奖励函数设计中的参数采用基于课程学习或域自适应的奖励设计方式,引入自动化参数优化方法(如贝叶斯优化、元学习或自适应奖励机制),还有些前沿强化学习的方法,比如持续强化学习、增量强化学习、元强化学习等方法,可以让强化学习提高其不同流量数据分布时的泛化性能。除了在网络仿真软件中的仿真工作,课题组未来计划进行后续一些实际性落地工作,我们尝试过用红米路由器做一些小规模的实验,后来基于硬件联发科技 MT7621 交换机芯片、软件 OpenWRT 路由器操作系统和 OpenFlow 协议的 OVS 交换机,自行设计了边缘端 SDN 实物板,如果大规模验证的话还需要批量焊接和定制这样 SDN 实物板,相比目前的 P4 交换机或者至少几万一个成品 SDN 交换机比较昂贵的方式,可以缩减很多成本。

此外,需要说明网络故障其实包括了节点故障和链路故障,本文这里主要解决的是链路故障,对节点失效故障处理是进行间接地转化成链路故障。本文期待下一步要解决的问题是节点新加入的路由问题,这是目前强化学习设计方法解决路由问题的难点,原因是新加入节点会引起网络拓扑结构改变,从而彻底改变以矩阵形式表示的状态空间形式,这不仅仅是网络故障问题讨论的范畴了,但这是和故障相对应的另一个非常有趣而又挑战性的研究问题,也将是我们下一步计划开展的非常有意义的研究工作。

致　谢　在此向对本文的工作给予支持和建议的审稿人和编辑、桂林电子科技大学广西无线宽带通信与信号处理重点实验室、认知无线电与信息处理教育部重点实验室通信研究所团队和西安电子科技大学计算机学院王宇平教授网络智能优化团队的老师和同学表示衷心的感谢。

参　考　文　献

[1] Karakus M, Durresi A. Quality of service (QoS) in software defined networking (SDN): A survey. Journal of Network and Computer Applications, 2017, 80: 200-218

[2] Hakiri A, Sellami B, Patil P, et al. Managing wireless fog networks using software-defined networking//Proceedings of the 2017 ieee/acs 14th international conference on computer systems and applications (aiccsa). Hammamet, Tunisia, 2017: 1149-1156

[3] Wang Y, Shang F, Lei J. Energy-efficient and delay-guaranteed routing algorithm for software-defined wireless sensor networks: A cooperative deep reinforcement learning approach. Journal of Network and Computer Applications, 2023, 217: 1-17

[4] Cicioğlu M, Çalhan A. MLaR: machine-learning-assisted centralized link-state routing in software-defined-based wireless networks. Neural Computing and Applications, 2023, 35(7): 5409-5420

[5] Li Y, Zhang Q, Yao H, et al. Stigmergy and hierarchical learning for routing optimization in multi-domain collaborative

satellite networks. IEEE Journal on Selected Areas in Communications, 2024, 42(5): 1188-1203

[6] Zhu S, Sun Z, Lu Y, et al. Centralized QoS routing using network calculus for SDN-based streaming media networks. IEEE Access, 2019, 7: 146566-146576

[7] Dhamala B K, Dawadi B R, Manzoni P, et al. QoS-oriented adaptive routing in distributed SDN using SARSA reinforcement learning//Proceedings of the 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). Kirtipur, Nepal, 2023: 301-306

[8] Zhao X, Band S S, Elnaffar S, et al. The implementation of border gateway protocol using software-defined networks: A systematic literature review. IEEE Access, 2021, 9: 112596-112606

[9] Mazyavkina N, Sviridov S, Ivanov S, et al. Reinforcement learning for combinatorial optimization: A survey. Computers & Operations Research, 2021, 134: 1-24

[10] Botvinick M, Ritter S, Wang J X, et al. Reinforcement learning, fast and slow. Trends in Cognitive Sciences, 2019, 23(5): 408-422

[11] Ge L, Li Y, Li Y, et al. Smart distribution network situation awareness for high-quality operation and maintenance: a brief review. Energies, 2022, 15(3): 1-24

[12] Casas-Velasco D M, Rendon O M C, da Fonseca N L S. DRSIR: A deep reinforcement learning approach for routing in software-defined networking. IEEE Transactions on Network and Service Management, 2021, 19(4): 4807-4820

[13] Caria M, Das T, Jukan A, et al. Divide and conquer: Partitioning OSPF networks with SDN//2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). 2015: 467-474

[14] Kotronis V, Gämperli A, Dimitropoulos X. Routing centralization across domains via SDN: A model and emulation framework for BGP evolution. Computer Networks, 2015, 92: 227-239

[15] Moufakir T, Zhani M F, Gherbi A, et al. Collaborative multi-domain routing in SDN environments. Journal of Network and Systems Management, 2022, 30(1): 1-23

[16] Karakus M. Gate-bc: Genetic algorithm-powered qos-aware cross-network traffic engineering in blockchain-enabled sdn. IEEE Access, 2024, 12: 36523-36545

[17] Xu A, Sun S, Wang Z, et al. Multi-controller load balancing mechanism based on improved genetic algorithm// Proceedings of the 2022 International Conference on Computer Communications and Networks (ICCCN). Honolulu, USA, 2022: 1-8

[18] Boyan J, Littman M. Packet routing in dynamically changing networks: A reinforcement learning approach. Advances in Neural Information Processing Systems, 1993, 6: 671-678

[19] Casas-Velasco D M, Rendon O M C, da Fonseca N L S. Intelligent routing based on reinforcement learning for software-defined networking. IEEE Transactions on Network and Service Management, 2020, 18(1): 870-881

[20] Yao Z, Wang Y, Qiu X. DQN-based energy-efficient routing algorithm in software-defined data centers. International Journal of Distributed Sensor Networks, 2020, 16(6): 1-12

[21] Lu Y, Chen Y, Xu X, et al. A sub-flow adaptive multipath routing algorithm for data centre network. International Journal of Computational Intelligence Systems, 2023, 16(1): 1-11

[22] Zhou W, Jiang X, Guo B, et al. PQROM: To optimize software defined network QoS-aware routing with proximal policy optimization. Journal of Intelligent & Fuzzy Systems, 2022, 42(4): 3605-3614

[23] Godfrey D, Jang J, Kim K I. Weight adjustment scheme based on hop count in Q-routing for software defined networks-enabled wireless sensor networks. Journal of Information & Communication Convergence Engineering, 2022, 20(1): 22-30

[24] Pagès A, Agraz F, Caro J B, et al. Machine learning-based multi-domain actuation orchestration in support of end-to-end service quality-assurance. IEEE Transactions on Network and Service Management, 2022, 20(3): 2575-2586

[25] Li Z, Zhou X, De Turck F, et al. Feudal multiagent reinforcement learning for interdomain collaborative routing optimization. Wireless Communications and Mobile Computing, 2022, 2022(1): 1-11

[26] Ye M, Huang L Q, Wang X L, et al. A new intelligent cross-domain routing method in SDN based on a proposed multiagent reinforcement learning algorithm. International Journal of Intelligent Computing and Cybernetics, 2024, 17(2): 330-362

[27] Guo L, Zhang J, Chen T, et al. Reinforcement learning-enhanced shared-account cross-domain sequential recommendation. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(7): 7397-7411

[28] Moayedi H Z, Masnadi-Shirazi M A. Arima model for network traffic prediction and anomaly detection//Proceedings of the 2008 international symposium on information technology. Kuala Lumpur, Malaysia, 2008, 4: 1-6

[29] Kim M. Network traffic prediction based on INGARCH model. Wireless Networks, 2020, 26(8): 6189-6202

[30] Nikravesh A Y, Ajila S A, Lung C H, et al. Mobile network traffic prediction using MLP, MLPWD, and SVM// Proceedings of the 2016 IEEE international congress on big data (BigData Congress). San Francisco, USA, 2016: 402-409

[31] Huang L, Ye M, Xue X, et al. Intelligent routing method based on dueling DQN reinforcement learning and network traffic state prediction in SDN. Wireless Networks, 2024, 30(5): 4507-4525

[32] Xu F, Lin Y, Huang J, et al. Big data driven mobile traffic understanding and forecasting: A time series approach. IEEE Transactions on Services Computing, 2016, 9(5): 796-805

[33] Shi H, Pan C, Yang L, et al. AGG: A novel intelligent net-

work traffic prediction method based on joint attention and GCN-GRU. Security and Communication Networks, 2021, 2021(1): 1-11

[34] Tang W, Xiao Y, Kong X, et al. CycleLLH: A novel network traffic prediction model based on periodic integration, 2024, 47(12): 2867-2888(in Chinese)
(唐文杰 ,肖一磊, 孔祥宇, 等. CycleLLH:一种基于周期性整合的新型网络流量预测模型. 计算机学报, 2024, 47(12): 2867-2888)

[35] Hameed A, Violos J, Leivadeas A, et al. Toward QoS prediction based on temporal transformers for IoT applications. IEEE Transactions on Network and Service Management, 2022, 19(4): 4010-4027

[36] Malik A, Aziz B, Adda M, et al. Smart routing: Towards proactive fault handling of software-defined networks. Computer Networks, 2020, 170: 1-14

[37] Liu Y, Guo R, Xu C, et al. A Q-learning-based fault-tolerant and congestion-aware adaptive routing algorithm for networks-on-chip. IEEE Embedded Systems Letters, 2022, 14(4): 203-206

[38] Valinataj M, Mohammadi S, Plosila J, et al. A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip. AEU-International Journal of Electronics and Communications, 2011, 65(7): 630-640

[39] Ke C K, Wu M Y, Hsu W H, et al. Discover the optimal IoT packets routing path of software-defined network via artificial bee colony algorithm//Proceedings of the 12th EAI International Conference, WiCON 2019, TaiChung, China, 2019: 147-162

[40] Yong B, Muqing W, Jing S, et al. Optimization strategy of SDN control deployment based on simulated annealing-genetic hybrid algorithm//Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC). Chengdu, China, 2018: 2238-2242

[41] Guo Y, Chen J, Huang K, et al. A deep reinforcement learning approach for deploying SDN switches in ISP networks from the perspective of traffic engineering//Proceedings of the 2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR). Taicang, China, 2022: 195-200

[42] Blial O, Ben Mamoun M, Benaini R. An overview on SDN architectures with multiple controllers. Journal of Computer Networks and Communications, 2016, 2016(1): 1-8

[43] Li Y, Cai Z P, Xu H. LLMP: exploiting LLDP for latency measurement in software-defined data center networks. Journal of Computer Science and Technology, 2018, 33(2): 277-285

[44] Sun P, Lan J, Li J, et al. A scalable deep reinforcement learning approach for traffic engineering based on link control. IEEE Communications Letters, 2020, 25(1): 171-175

[45] Bhavanasi S S, Pappone L, Esposito F. Routing with graph convolutional networks and multi-agent deep reinforcement learning// Proceedings of the 2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). Phoenix, USA, 2022: 72-77

[46] Li J, Ye M, Huang L, et al. An intelligent SDWN routing algorithm based on network situational awareness and deep reinforcement learning. IEEE Access, 2023, 11: 83322-83342

[47] Fontes R R, Afzal S, Brito S H B, et al. Mininet-WiFi: Emulating software-defined wireless networks//Proceedings of the 2015 11th International conference on network and service management (CNSM). Barcelona, Spain, 2015: 384-389

[48] Dumitrache C, Predusca G, Gavriloaia G, et al. Comparative analysis of routing protocols using GNS3, Wireshark and IPerf3// Proceedings of the 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). Ploiesti, Romania, 2022: 1-6

[49] O. Ryu. https://ryu-sdn.org/. (accessed Mar. 14, 2023)

[50] Tayfour O E, Marsono M N. Collaborative detection and mitigation of distributed denial-of-service attacks on software-defined network. Mobile Networks and Applications, 2020, 25(4): 1338-1347

[51] Ye M, Zhao C, Wen P, et al. DHRL-FNMR: An intelligent Multicast routing approach based on deep hierarchical reinforcement learning in SDN. IEEE Transactions on Network and Service Management, 2024, 21(5): 5733-5755

YE Miao, Ph. D., professor Ph. D. supervisor. His primary research interests include edge storage and cloud storage, software-defined networking, wireless sensor networks, pattern recognition, machine learning, artificial intelligence methods and applications (including deep reinforcement learning and graph neural networks).

LI JIn-Qiang, Ph. D. candidate. His main research interests include software defined networking, reinforcement learning.

HE Qian, Ph. D., professor, Ph. D. supervisor. His main research interests include distributed computing, software defined network.

WANG Xiao-Li, Ph. D., associate Professor, master supervisor. His main research interests include artificial intelligence, distributed computing.

WANG Yu-Ping, Ph. D., professor, Ph. D. supervisor. His major research interests include evolutionary computation, optimization theory, and optimal design method for network and engineering, data mining.

WANG Yong, Ph. D., professor, Ph. D. supervisor. His main research interests include cloud computing, software defined networking, network traffic classification, Information security.

WEN Peng, Ph. D. candidate. His main research interests include software defined networking, reinforcement learning, and stochastic optimization and application.

Background

This paper addresses the issue of intelligent link failure-aware adaptive inter-domain routing in multi-domain Software-Defined Wireless Networks (SDWN), an important topic in the field of Information and Communication Technology (ICT). Existing methods struggle to adapt effectively to increasing network node scale and network failures, often exhibiting long adaptation times, slow convergence speeds, poor adaptability of routing strategies, and insufficiently timely and flexible forwarding. International efforts in this area have been limited, focusing mainly on traditional routing protocols and preliminary reinforcement learning approaches. However, these methods frequently lack sufficient awareness of network traffic conditions, fail to meet Quality of Service (QoS) requirements, and encounter challenges in highly dynamic network environments.

This paper proposes a novel SDWN inter-domain intelligent link failure-aware adaptive routing method (DHRL-ACTF), which is based on Transformer traffic prediction and hierarchical reinforcement learning, effectively addressing the above issues. The method designs an SDWN network measurement mechanism to flexibly acquire network status information, utilizing a Transformer-based prediction mechanism to sense future traffic trends. In the root controller module, a hierarchical reinforcement learning mechanism based on the Actor-Critic architecture is designed, decomposing the inter-domain routing problem into two sub-problems: inter-domain node selection and optimal path selection. Additionally, different action selection strategies are designed to optimize routing, minimize redundant branches, and dynamically adjust the agent's learning trajectory. Finally, a reward function is crafted using network link information and a reward-punishment mechanism to guide the agent in efficiently utilizing network resources, generating high-performance inter-domain intelligent paths.

This work was partly supported by the National Natural Science Foundation of China (Nos. 62161006, 62372353, U22A2098), Guangxi Wireless Broadband Communication and Signal Processing Key Laboratory (Nos. AD25069102), Key Laboratory of Cognitive Radio and Information Processing of Ministry of Education (No. CRKL220103), Innovation Project of Guangxi Graduate Education (No. 2025YCXS078). The funding projects have provided essential financial support and resources, ensuring the smooth conduct of this research and the production of high-quality outcomes. This support has been crucial in facilitating the research process and enhancing the overall quality of the results.

Our team has achieved significant progress in SDWN routing optimization, covering traditional routing protocols such as BGP and OSPF, as well as deep reinforcement learning strategies like PPO, DQN, and Q-learning, aiming to improve bandwidth utilization, reduce latency, and enhance resilience to link failures. The proposed DHRL-ACTF algorithm combines Transformer traffic prediction and hierarchical reinforcement learning, achieving superior performance in bandwidth, latency, and packet loss rate, significantly enhancing network service quality and adaptability. Our findings contribute to the advancement of SDWN and offer new perspectives for future network research.