MA-CDMR:多域SDWN中一种基于多智能体深度强化学习的智能跨域组播路由方法

叶 苗[1),2)] 胡洪文[2)] 王 勇[1),2)] 何 倩[2)] 王晓丽[1),3)]
文 鹏[1),2)] 郑基浩[1),2)]

[1)](桂林电子科技大学认知无线电与信息处理教育部重点实验室 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 多域软件定义无线网络(SDWN)中的跨域组播路由问题不仅是NP难组合优化问题,随着网络规模的增加和组播组成员的动态变化,构建高效的跨域组播路由路径还需要及时灵活获取和维护全局网络状态信息并设计出最优跨域组播树问题的求解算法。针对现有求解方法对网络流量状态感知性能欠缺影响组播业务对QoS方面需求的满足,并且收敛速度慢难以适应网络状态高度动态变化的问题,本文设计和实现了一种基于多智能体深度强化学习的SDWN跨域组播路由方法(MA-CDMR)。首先,设计了组播组管理模块和多控制器之间的通信机制来实现不同域之间网络状态信息的传递和同步,有效管理跨域组播组成员的加入和离开;其次,在通过理论分析和证明最优跨域组播树包括最优的域间组播树和域内组播树两个部分的结论后,本文对每个控制器设计了一个智能体,并设计了这些多智能体之间的协作机制,以保证为跨域组播路由决策提供网络状态信息表示的一致性和有效性;然后,设计一种在线与离线相结合的多智能体强化学习训练方式,以减少对实时环境的依赖并加快多智能体收敛速度;最后,通过系列实验及其结果表明所提方法在不同网络链路信息状态下具有达到了很好的网络性能,平均瓶颈带宽相较于现有KMB、SCTF、DRL-M4MR和MADRL-MR方法分别提升了7.09%、46.01%、9.61%和10.11%;平均时延在与MADRL-MR方法表现相近的同时,相比KMB、SCTF和DRL-M4MR方法有明显提升,而丢包率和组播树平均长度等也均优于这些现有方法。本文工作源代码已提交至开源平台https://github.com/GuetYe/MA-CDMR。

关键词 组播树;软件定义无线网络;跨域组播路由;多智能体;深度强化学习
中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2025.01417

MA-CDMR: An Intelligent Cross Domain Multicast Routing Method Based on Multi-Agent Deep Reinforcement Learning in SDWN Multi Controller Domain

YE Miao[1),2)] HU Hong-Wen[2)] WANG Yong[1),2)] HE Qian[2)] WANG Xiao-li[1),3)]
WEN Peng[1),2)] ZHENG Ji-Hao[1),2)]

[1)](Key Laboratory of Cognitive Radio and Information Processing of Ministry of Education, 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 cross-domain multicast routing problem in a software-defined wireless network

收稿日期:2024-07-12,在线发布日期:2025-03-18。本课题得到国家自然科学基金项目(Nos. 地区62161006,面上62372353,重点U22A2098)、认知无线电与信息处理教育部重点实验室主任基金(No. CRKL220103)和广西研究生教育创新计划项目(No. YCBZ2023134)资助。**叶 苗**,博士,教授,博士生导师,主要研究领域为边缘存储和云存储、软件定义网络、无线传感器网络、人工智能方法和应用(包括深度强化学习和图神经网络)。E-mail: ym@mail.xidian.edu.cn。**胡洪文**,博士研究生,主要研究领域为软件定义网络、强化学习。**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、软件定义网络等。E-mail: ywang@guet.edu.cn。**何 倩**(通信作者),教授,博士,博士生导师,中国计算机学会(CCF)杰出会员,主要研究方向为分布式计算、软件定义网络。E-mail:heqian@guet.edu.cn。**王晓丽**,副教授,硕士生导师,主要研究方向为人工智能、分布式计算。**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。**郑基浩**,硕士研究生,主要研究方向软件定义网络、强化学习。

(SDWN) with multi-controller domains is a classic NP-hard combinatorial optimization problem. As the network size increases, the design and implementation of cross-domain multicast routing paths in this network necessitates not only the development of efficient algorithms for identifying the optimal cross-domain multicast tree but also the assurance of timely and flexible acquisition and maintenance of global network state information. To address the shortcomings of existing solutions have limit ability to sense the network traffic state, affecting the quality of service (QoS) of multicast services, and the difficulty adapting to the highly dynamically changing network state and slow convergence speed, this paper designs and implements a multiagent deep reinforcement learning-based cross-domain multicast routing (MA-CDMR) method for the SDWN with multi-controller domains. First, a multi-controller communication mechanism and a multicast group management module are designed to transfer and synchronize network information between different control domains of the SDWN, respectively, thus effectively managing the joining and leaving of members in the cross-domain multicast group. Second, a theoretical analysis and proof show that the optimal cross-domain multicast tree includes an interdomain multicast tree and an intradomain multicast tree. An agent is set up for each controller, a cooperation mechanism between multiple agents is designed to effectively optimize cross-domain multicast routing and improve the overall network performance and resource utilization, and a multiagent reinforcement learning-based method that combines online and offline training is designed to reduce the dependence on the real-time environment and increase the convergence speed of multiple agents. Finally, a series of experiments and their results show that the proposed method achieves good network performance under different network link information states, and the average bottleneck bandwidth is improved by 7.09%, 46.01%, 9.61%, and 10.11% compared with the existing KMB, SCTF, DRL-M4MR, and MADRL-MR methods, respectively; the average delay is similar to that of MADRL-MR, and significantly better than that of KMB, SCTF, and DRL-M4MR; the packet loss rate and the average length of multicast tree are also better than those of these existing methods. The average delay is similar to that of the MADRL-MR method while it is significantly improved over the KMB, SCTF and DRL-M4MR methods, while the packet loss rate and the average length of the multicast tree are also better than these existing methods. The source code of this work has been made available on the open-source platformhttps://github.com/GuetYe/MA-CDMR.

Keywords multicast tree; software defined wireless network; cross domain multicast routing; multi agent; deep reinforcement learning

1 引 言

随着无线通信技术的不断发展和应用场景的推广普及，其中的无线组播通信被广泛应用于多媒体会议、实时视频传输、团队协作和分布式计算等各种场景中。与点对点通信相比，组播通信[1]可以将相同的数据传输给多个接收者，实现在无线网络的中高效信息分发和共享，可以高效地节约网络带宽、降低网络负载。设计组播通信中的组播路由，本质在于构造一棵由源节点到所有目的节点的最优组播树[2]，以达到最大化瓶颈带宽，最小化传输时延和丢包率等性能指标，提高网络资源的利用率。然而，在高速动态变化的无线网络中构建最优组播树需要尽可能实时地获取全局网络链路状态信息。原有传统的无线网络管理方式效率低，难以实现全局网络状态信息的获取。

软件定义无线网络(Software Defined Wireless Network, SDWN)技术[3]将网络控制平面与数据平面分离，通过集中管理和灵活编程机制实现网络状态信息的获取和网络资源的全局优化配置，可以很好地解决传统无线网络管理方式的缺陷。然而，随

着网络规模和复杂性的增加，单域SDWN管理模式容易发生单点故障、扩展性差和难以适应异构性等性能瓶颈。为了克服这些限制并提高网络性能和可靠性，通常将SDWN网络由单控制器管理模式扩展到多控制器管理模式，从而还需考虑SDWN中多控制器域的组播路由问题。

在传统多域无线网络中，传统组播边界网关协议MBGP（MultiProtocol Border Gateway Protocol）[4]通过在边界路由器进行配置和建立对等连接，在跨域环境中传递组播路由信息，并根据这些信息实现组播数据的正确转发。但MBGP实现域间路由的配置较为繁琐，且与控制器之间缺乏直接集成，需要额外的开发和定制工作，缺乏全局视图导致的资源浪费等缺陷，可能会导致消息传递和同步的延迟和不一致[5]，并不适合在多域SDWN中组播路由路径的构建，而且由于域内和域间的网络链路信息均是高度动态变化的，组播源节点和组播目的节点还可能分布在不同域中，这就对组播组的发现、组播路由表的更新、跨域组播树的构建和跨域数据传输等方面提出了挑战。因此，如何在多域SDWN中设计出适应域内和域间网络链路信息高度动态变化的跨域组播树以满足高性能跨域组播业务需求是一项非常有意义而重要的研究。

求解最优组播树的一些经典优化方法有基于剪枝策略和贪心搜索策略的近似最优求解方法（例如KMB[6]和SCTF[7]）和启发式群智能优化方法（例如遗传算法GA[8]）。这些方法由于不够灵活性而无法适应高速动态变化网络流量的需求，而且通常只关注少数的优化性能指标，缺乏全局优化能力。深度强化学习算法[9]作为一种数据驱动的解决方案具备很好的学习能力，能够从大量网络数据中提取特征和模式，生成最优的组播路由策略，从而提升组播路由的性能与效果，相比传统方法具有更强的灵活性从而能适应复杂的网络拓扑和动态变化的流量需求，现阶段已有将深度强化学习用于求解组播问题，例如基于强化学习机制的路由方法Q-RL[10]，DRL-M4MR[11]，MADRL-MR[12]方法。但这些算法讨论的是单域SDN的组播问题，而在多域SDWN网络中的组播问题还需要解决控制器域间的消息传递和同步、组播组的管理、组播路由表的更新以及如何有效地构建跨越多个控制器域的组播路由路径的问题。

此外，现阶段用于解决路由优化问题的深度强化学习算法大多是Online强化学习[13]方式，其训练策略分为同策略on-policy[14]和异策略off-policy[15]两种方式，这两种策略均需要与环境交互。其实在大规模的网络环境下，智能体与环境的实时交互成本是比较高的，这在训练过程中需要进行大量的试错和探索，并且在线强化学习通常只能利用当前的实时数据，对历史数据的利用率较低。这可能会导致高昂的资源消耗和训练成本。而离线强化学习（Offline Reinforcement Learning）[16]通常不需要与环境进行交互，所以能够较好地适应大规模网络环境的场景，减少与环境实时交互的成本，能通过有效利用历史数据来提高学习效果和训练速度。对于实时动态变化的网络环境如果采用离线强化学习的方式来解决路由问题，还需要解决单纯使用离线数据可能导致的样本偏差和缺乏探索的问题。

通过以上现有方法的分析，本文对多域SDWN中跨域组播路由问题设计了一种基于多智能体深度强化学习的智能求解方法（MA-CDMR），为了解决不同域之间网络状态信息的同步和传递，以及组播组成员分布在不同控制器域中带来的组播组成员发现问题，设计了一个控制器通信模块和组播组管理模块以实现不同域之间信息的一致性，以及跨域组播组成员的发现和管理组播组成员的加入和离开；在通过理论分析和证明了最优跨域组播树（Optimal cross-domain multicast tree）包括域间组播树（Inter-domain multicast tree）和域内组播树（Intra-domain multicast tree）两个部分的结论后，为解决域内和域间组播树构建的问题分别设计了求解智能体，并通过为这些智能体设计了问题相关的协作机制，实现了进行跨域组播路由决策的网络状态信息表示的一致性和有效性；为了提升多智能体训练效果和加快收敛速度，减少与规模环境实时交互的成本，结合离线强化学习设计了一个在线与离线相结合的多智能训练方式；考虑对这些域内智能体而言，彼此之间在逻辑上是独立的，他们分别具备各自独立的状态空间，并在自己独立的动作空间上设计自己的动作策略，因此本文设计了一种完全去中心化（Fully decentralized，IL）的求解范式，每个智能体根据本地信息进行决策，避免了中心控制可能带来的干扰，能够更好地适应环境的变化，提高了多智能体的协作效率。

本文的主要研究贡献如下：

（1）针对软件定义无线网络SDWN多域场景的特点，提出了一种基于网络状态信息感知和多智能体深度强化学习的组播问题求解框架。根据已知

研究，本文首次通过对软件定义网络多域场景下的组播路由问题进行理论分析，将跨域组播树的求解问题分解为域间组播树构建和域内组播树构建的两个子问题，并对这些子问题分别设计了协作式多智能体强化学习求解算法对SDWN多域场景下的组播树问题进行求解。

（2）相比于传统的组播边界网关协议，本文基于SDWN控制逻辑集中和可编程的特点设计了控制器通信机制和组播组管理模块，以实现多个控制域间消息的传递和同步，以及对跨域组播组成员的发现和有效管理。这样的设计解决了传统组播边界网关协议配置复杂、难以适应网络状态高度变化的问题，通过灵活便捷地获取全局网络状态信息，实现控制器域之间更好的协调和协作，为后续设计数据驱动的组播流智能分发算法提供了全局数据视图的数据基础。

（3）针对SDWN多控制器域中跨域组播问题的具体背景特点，与其他求解方法考虑部分网络链路信息构建组播树不同，本文设计了综合考虑网络链路的带宽、时延、丢包率、错包率、AP间的距离以及组播树状态等矩阵组成的状态空间，使得智能体能够更好地感知网络链路状态信息的变化和组播树构建过程的变化；并且根据分解得到的域间组播树和域内组播树问题的特点，分别设计了相应的动作策略，提高了智能体的探索效率。设计的构建域间组播树的智能体动作空间为所有域之间的边的集合，每次动作选择为其中一条边；设计的构建域内组播树的智能体动作空间为域内节点集合，每次动作选择为下一跳节点。对智能体所采取的不同动作策略设计不同的奖励函数，如单步决策奖励、完成构造路径奖励、无效动作惩罚和环路惩罚。引导智能体构建出高效的域间和域内组播树。

（4）设计的多智能体通过协同学习和策略协调来实现跨域组播树的构建和优化，采用完全去中心化的求解范式以提高多智能体协作的稳定性，同时设计了离线和在线相结合的训练方式减少了与环境的交互频率和对实时环境的依赖，有效提升多智能体的收敛速度。

本文其余部分组织如下。第2节介绍了相关工作。第3节分析了组播、斯坦树和跨域组播树问题。第4节介绍了SDWN多控制器域智能跨域组播路由架构。第5节详细介绍了MA-CDMR算法。第6节介绍了实验环境和性能评估结果。第7节介绍了结论和未来的工作。

2 相关工作

本节中主要介绍软件定义网络中多控制器多域场景中跨域组播路由的现有求解方法并进行相应的一些分析。

传统组播树问题求解方法：L. Kou 等人[6]提出了一种基于最短路径和最小生成树算法（KMB）来构建斯坦纳树。Angelopoulos 等人[17]使用SCTF（Selective Closest Terminal First）算法来构建组播树，该算法通过采用Dijkstra算法计算从源到所有目的节点的路径，最后将从源到所有目的节点的拓扑路径作为近似斯坦纳树解。H. Takahashi 等人[18]提出了最小代价路径启发式算法（MPH）。V. J. Rayward-Smith 等人[19]设计了一种基于平均距离算法（ADH）。Naser等人[20]在SDN集群环境下提出了一种考虑各节点剩余功率的最小功耗组播树的数学模型，基于此模型提出了三种基于集群的组播路由策略。Weixiao等人[21]考虑组播传输的带宽、利用率和时延等网络性能，提出一种多约束组播路由突变机制。此外，针对组播树生成算法时间复杂度高的问题，提出了一种多域多控制器组播路由算法，该算法通过改进SPT算法，通过寻找前k条短路径，随机选择从组播源到目的的所有路径进行叠加，采用贪婪算法进行解环操作，得到组播树。Chiang等人[22]讨论了多域在线分布式多播流量工程，采用领域树、双候选森林构建和森林重路由的思想设计了一种竞争分布式算法，通过每个域中的控制器与相邻域中的控制器共享其网络状态信息，协同构建跨域组播路由，其中组播树的构建采用最短路径树SPT算法。Liu等人[23]基于软件定义广域网（SD-WAN）提出了一种多约束多目标路径优化算法（MCOPF）。该算法将请求流分为延迟敏感流和其他流。除了保证每个数据流的QoS（服务质量）外，算法设计的目标是最大限度地保证对延迟敏感的流请求，更好地利用网络资源。为了提高算法的有效性，采用了层次化的多控制器协同框架，将控制平面划分为多个层次，由根控制器管理全局视图，域控制器管理对区域网络。以上传统算法通常只关注单一的优化目标，如最短路径或带宽等，然而单一的目标并不能准确地反应网络的实际状况，可能导致无法实现全局最优和网络拥塞等问题，并且传统算法可扩展性有限，在大规模的网络环境中，路由器之间的路由信息交换和计算负载会急剧增加，导致网络的稳定性和性能下降。

智能优化求解方法：Wu等人[24]提出了一种基于混合群智能的多层多域光网络组播恢复算法，将基于PCE光网络架构的人工鱼模型与非合作博弈相结合进行域内路径搜索，并采用改进的果蝇优化算法寻找恢复路径，在提高收敛速度的同时获得更好的局部或全搜索路径选择。Ke等人[8]提出了一个基于遗传算法GA的组播路由方法，该方法用于在同时考虑流类型和网络状态时为大数据中心网络中的流调度问题设计最优的组播路由策略。Liu等人[25]提出了一种基于时移多层图（TS-MLG）框架的路由方案，该方案兼顾了空间和时间两方面，实现了域间的批量数据传输，可以通过常规路由计算确定跨域路由路径、存储点和传输时间。但该方案只能提高域间带宽的利用率，并且网络状态维护开销和计算复杂度较高。Zhao等人[26]在考虑差分服务质量（QoS）和节能的前提下，提出了一种基于子拓扑图的分层控制平面结构支持的智能域间路由方案。该方案能够实现多域传输质量（QoT）、能量感知路由和频谱分配（RSA）。但该方案只适合于域间路由，并没有解决域内路由问题。Li等人[27]在多域分组网络中基于深度学习设计了一个可扩展和协议无关的路径算法，使其支持数据平面的协议无关转发。但该算法只考虑了带宽作为QoS需求，没有考虑链路的时延、丢包率等信息，并且其训练复杂度较高，难以适应动态变化的网络环境。

基于强化学习的求解方法：Xu等人[28]提出了一种多域弹性光网络的分层强化学习框架以实现域间业务请求的路由、调制和频谱分配（RMSA）。该框架由一个高级DRL模块和多个低级DRL模块（每个域一个）组成，DRL模块之间相互协作。对于域间服务请求，高层模块从低层DRL模块中获得一些抽象信息，并为低层模块生成域间RMSA决策。但该方法只关注于域间业务请求路由。Dang等人[29]提出了一种基于多智能体强化学习（MARL）的优化链路权值的高效链路状态组播路由方法，该方法首先提供了一个整数线性规划（ILP）公式为基于最短路径树（SPT）的多播路由协议找到最优链路权重，以最小化总网络成本。然后设计了一个多智能体强化学习（MARL）解决方案来优化链路权重问题，以实现分布式方式的高效组播路由。但该方法的多智能体强化学习只用于优化链路权重，并没有考虑多个智能体之间交互的问题。Zhao等人[30]提出了一种多智能体强化学习跨域服务功能链路由方法，该方法综合考虑路径的嵌入成本、时延和每个域的负载情况作为路由的构建成本。但该方法仅考虑网络链路的时延，对于其他如带宽、丢包率等网络链路信息感知性较差。Bhavanasi等人[31]提出了一个基于图神经网络和多智能体深度强化学习的弹性路由方法，该方法使用图神经网络算法来设计强化学习智能体，以图格式编码的网络流量数据集作为训练输入，其优势在于允许强化学习策略（即给定的奖励函数）在任何拓扑中运行，并从路由和拥塞事件的变化中学习，而无需重新训练神经网络。但该方法在更大网络上重新学习最优策略的能力存在局限性。Ye等人[32]提出了一个基于多智能体强化学习和网络流量预测的SDN跨域智能路由算法，该方法为每个域设置一个Dueling DQN智能体，使用循环神经网络进行网络流量预测，其中智能体的动作设计为k-paths，k条路径均由Dijkstra算法生成，通过让智能体学习从中选择出最优路径，但生成的k条路径是固定的，并且无法及时随着网络链路流量的动态变化而改变路径，存在局部最优问题。Casas-Velasco等人[10]提出了一种基于Q-learning的组播方法，该方法考虑链路状态信息来做出路由决策，该文献考虑的主要是单播路由。C. Zhao等人[11]设计了SDN中基于DQN的深度强化学习智能组播路由方法，该方法综合考虑链路的带宽、时延和丢包率，且该方法存在智能体的收敛速度较慢的问题。

为解决以上问题，本文设计和实现了一种在SDWN多控制器域中基于多智能体深度强化学习的跨域组播路由方法（MA-CDMR）。通过设计一种多控制器通信机制和组播组（multicast groups）管理模块分别实现SDWN不同控制域之间网络信息的传递和同步，将最优跨域组播树（Optimal cross-domain multicast tree）分解成域间组播树（Inter-domain multicast tree）和域内组播树（Intra-domain multicast tree）两个部分，并为每个控制器设计了协助方式的多智能体强化学习求解算法，并且通过在线与离线相结合的多智能体强化学习训练方式减少对实时环境的依赖以加快多智能体收敛速度。

3　跨域组播问题描述与建模

3.1　组播和斯坦纳树问题

组播是一种在计算机网络中同时向多个目标节点传输数据的通信方式。组播问题的目标是构建一

棵组播树，以最小化传输成本。组播树是一个以组播源为根节点，并覆盖所有目的节点的树状结构。通过构建最小成本的组播树，可以有效地将数据传输给所有目的节点，同时减少网络带宽和传输延迟。最小成本组播树问题实际上对应了图论中的最小斯坦纳树问题[33]。

给定一个无线加权图$G(V,E,w)$表示，其中V是图G中的节点集合，E是图G中边的集合，w是边的权重。e_{ij}表示从节点i到节点j的边，$e_{ij}\in E$，边的权重为$w(e_{ij})$。再给定一个组播节点集合$M\subseteq V$，且$M=\{\text{src}\}\cup D$，其中src是源节点，D是组播目的节点集合，$D=\{d_1,d_2,\cdots,d_n\}$。图G'是图G中包含组播节点集M的子图。其中，图G'还包含一些不在M中的节点，这些节点称为斯坦纳节点。

最小斯坦纳树问题的目标就是在图G'中找出一棵包含M的生成树$T=(V_T,E_T)$，并使其边的权值之和最小。如公式(1)所示。

$$\min_{T\subseteq G,M\subseteq V_T}\sum_{e_{ij}\in E_T}w(e_{ij}) \tag{1}$$

其中，V_T表示树T中的所有节点，E_T表示树T所有的边。

3.2 跨域组播树问题

本文考虑如下软件定义无线网络在多控制器场景下的跨域组播问题及其求解方法，假设后续工作是在多域场景下进行，即整个无线网络已经被合理划分成了m个控制域$N_i,i=1,\cdots,m$(软件定义网络控制域的划分属于软件定义网络领域的另一个关键技术，本文后续讨论的跨域组播问题都是假定在整个网络被确定划分以后进行的，并且不考虑划分后的每个控制域有SDN交换机发生动态迁移情况的发生)，并且假设任何两个域之间的域间路径开销大于域内任意两个节点之间路径的开销，总结以上对问题背景的描述，我们有如下两个假设成立：

假设1. 软件定义无线网络经过合理划分后，每个域内SDN交换机节点及其连接关系固定，不需要考虑有交换机节点迁入迁出情况的发生。

假设2. 任何两个域之间链路的域间路径开销大于域内任意两个节点之间路径的开销。

组播树包括组播源和组播的目的节点，它们全都在同一域中时就对应了第三章的工作，本章主要讨论它们在不同域中的情形。组播源用src表示，组播的目的节点为$d_i,i=1,\cdots,n$，这n个组播目的节点有可能有部分与src在同一个控制域中，也有可能都不在同一个控制域中。组播路由路径包括域内路由路径部分和域间路由路径部分。如图1中所示，其中蓝色节点为src，橙色节点为d_i，红色外框节点为域与域之间边界节点(BN，Boundary node)，不是一般性，用$BN_{i,a}$表示域N_i中的第a个边界节点，用映射函数$dm:N_i\rightarrow\{d_j\}$表示任给一个域N_i内包含的目的节点d_j的集合，用映射函数$Domain(d_i):d_i\rightarrow N_j$表示任给一个目的节点$d_i$所对应的域$N_j$，用映射函数$BND:N_i\rightarrow\{BN_{ia}\}$表示任给一个域$N_i$所对应的边界节点$BN_{ia}$的集合。这三个映射函数$dm(N_i)$、$Domain(d_i)$和$BND(N_i)$在多控制域的划分后就确定了的，可以认为是已知的。

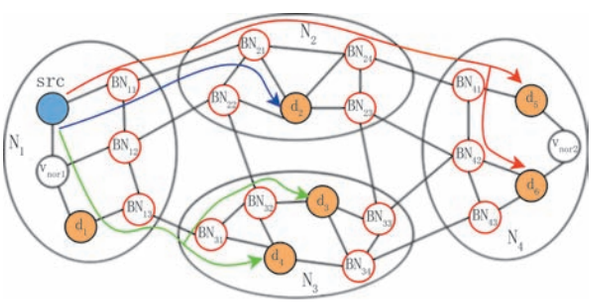


图1 多域场景下的组播树

组播路由路径包括域内路由路径部分(为了表述方便本文后续部分也称之为域内组播树)和域间路由路径部分(为了表述方便本文后续部分也称之为域间组播树)两部分，最优跨域组播树的问题可以通过如下一些定义来表示：

定义1. 组播树中从源节点src到目的节点d_i域间路径PN_i：表示从源节点src达到目的节点d_i时依次经过的每个控制域的域序列$PN_i=<N_{i,1},N_{i,2},\cdots,N_{i,j},\cdots,N_{i,|PN_i|}>=(\mathcal{N}_i,\mathcal{E}_i)$，其中$N_{i,j}\in\mathcal{N}_i$表示域间路径$PN_i$经过的第$j$个域，$\mathcal{N}_i$表示域间路径$PN_i$经过的所有域的集合，$\mathcal{E}_i$表示$\mathcal{N}_i$中各个域之间的连通关系，即域间路径$PN_i$中相邻的域$N_{i,j-1}$和$N_{i,j}$之间的边$<N_{i,j-1},N_{i,j}>$。

例如：图1中的红色、蓝色和绿色表示的三条域间路径分别为：$PN_2=<N_1,N_2>$，$PN_3=PN_4=<N_1,N_3>$，$PN_5=PN_6=<N_1,N_2,N_4>$。

定义2. 域间组播树：源节点src到所有目的节点d_i的域间路径PN_i所对应的一棵树，$i=1,\cdots,n$。这棵树只包含各个域之间的链路，可以看作是以域为“节点”、以表示域间相邻连通关系表示的“边”形成的一个图中的一棵组播树。

可以用n条源节点src到目的节点域间路径节

点d_i域间路径$PN_i, i=1,\cdots,n$，表示任意一颗域间组播树T_{int}，即$T_{int}=\{PN_1,\cdots,PN_i,\cdots PN_n\}$；也可以用域与域之间边界节点（BN，Boundary node）连接边表示这棵域间组播树T_{int}，其边的集合为$P_{int}=\{(BN_{ia},BN_{jb})|BN_{ia}\in BND(N_i),BN_{ja}\in BND(N_j)\}$，其中$(BN_{ia},BN_{jb})$表示连接域$N_i$和域$N_j$的边，$BN_{ia}$表示域$N_i$中的第$a$个边界节点，$BN_{ja}$表示域$N_j$中的第$b$个边界节点，则$T_{int}=P_{int}$，如图2所示的一颗域间组播树，其中$BN_{11}$、$BN_{13}$、$BN_{21}$、$BN_{24}$、$BN_{34}$、$BN_{43}$为选取的边界节点，$N_1$为源域，$N_2,N_3,N_4$为目的域，此时的这颗域间组播树可以表示为：$P_{int}=\{(BN_{11},BN_{21}),(BN_{13},BN_{21}),(BN_{24},BN_{41}),(BN_{34},BN_{43})\}$。一般情况下有

$$T_{int}=\{PN_1,\cdots,PN_i,\cdots,PN_n\}\ or$$
$$T_{int}=P_{int}=\{(BN_{ia},BN_{jb})|$$
$$BN_{ia}\in BND(N_i),BN_{jb}\in BND(N_j)\}\quad(2)$$

定义3.　定义的域间组播树其实就是跨域组播树的域间路由路径部分，与此相对应的就是跨域组播树的域内路由路径部分，关于域内路由路径部分我们有如下域内组播树和域内组播森林的定义。

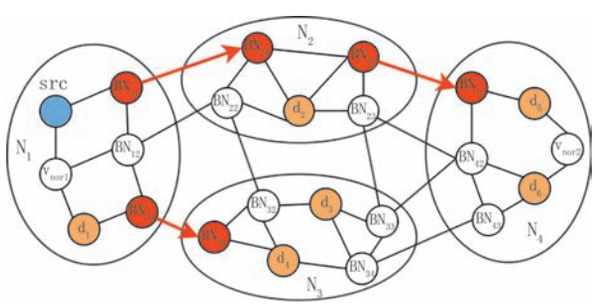


图2　域间组播树

定义4.　域N_i内的域内组播树和域内组播森林T_i：从源节点src达到各个目的节点$d_i, i=1,\cdots,n$时，分别经过各个域的域内路由路径部分，用T_i表示域N_i内的路径部分，很显然，只要有路径经过其中一个域N_i，则肯定有$T_i\neq\varnothing$，T_i有可能是一棵树，也有可能是多棵树组成的森林，可以分为如下几种情形分别进行定义：

（1）域N_i内没有组播树的任何一个要到达的目的节点，这时组播路径只是经过该域，从边界节点BN_{ia}进入该域N_i，从其他边界节点BN_{ib}离开该域N_i，若只有一个进入域N_i的边界节点BN_{ia}，则域N_i内的路径部分就只有一个连通分量，此时T_i就可以看成是一棵以BN_{ia}为根，以离开的边界节点BN_{ib}为叶子节点的树，我们称之为域内组播树T_i，如图3(a)所示；若有多个进入域N_i的边界节点BN_{ia}，则域N_i内的路径部分可能有一个连通分量，如图3(b)所示，也有可能有多个连通分量，如图3(c)所示，此时T_i就可以看成是一个以多个分别以不同BN_{ia}为根的树组成的森林，我们称之为域内组播森林T_i。

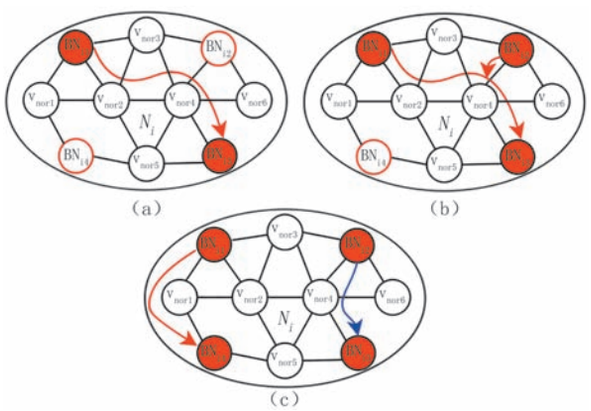


图3　域内无目的节点情况

（2）域N_i内有组播树要到达的目的节点$d_i\in dm(N_i)$，如果这样的目的节点只有一个，即$|dm(N_i)|=1$，则组播路径只会通过唯一的一个边界节点BN_{ia}进入这个域N_i，此时T_i就是一棵以BN_{ia}为根的树（如果没有离开该域N_i的边界节点，则该树T_i是一条链，如图4(a)所示，如果还有离开该域N_i达到其他域的边界节点BN_{ib}，则该树T_i就是一棵以d_i和BN_{ib}为叶子节点的普通意义上的树，我们称之为域内组播树T_i，如图4(b)所示）；如果这样的目的节点不止一个，即$|dm(N_i)|>1$，则组播路径有可能会通过不止一个边界节点BN_{ia}进入这个域N_i分别到达这几个目的节点，则域N_i内的路径部分可能有一个连通分量，如图4(c)所示，也有可能有过个连通分量，如图4(d)所示，此时T_i就可以看成是一个以多个分别以不同BN_{ia}为根的树组成的森林，我们称之为域内组播森林T_i。

对定义4中定义的域N_i内域内组播树和域内组播森林T_i，我们这里简单地用边集合表示：

$$T_i=\{(v_1,v_2),(v_2,v_3),\cdots,(v_{i,l},v_{i,l-1})\}\quad(3)$$

其中，$(v_{i,l},v_{i,l-1})$表示域N_i中连接域内节点$v_{i,l-1}$和域内节点$v_{i,l}$的边。

对没有域间组播树经过的域N_i，此时要求解的组播树没有任何一条链路经过域，这可以认为其域内组播树或者域内组播森林$T_i\neq\varnothing$，则所有域$N_i(i=1,\cdots,m)$的域内组播树的集合T_{intra}表示为：

$$T_{intra}=\bigcup_{i=1}^{m}T_i\quad(4)$$

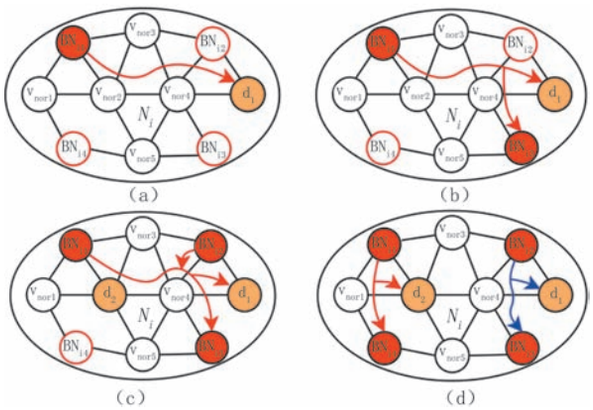


图4 域内有多个目的节点情况

其中，m表示域的个数。

图5给出了一个域内组播树的例子，4棵域内组播树分别用T_1, T_2, T_3和T_4表示。如图5所示，其中src为源节点，d_1, d_2, d_3, d_4, d_5和d_6为目的节点，v_{nor1}和v_{nor2}为普通节点，$T_1=\{(src, BN_{11}), (src, v_{nor1}), (v_{nor1}, d_1), (d_1, BN_{13})\}$，$T_2=\{(BN_{21}, BN_{24}), (BN_{21}, d_2)\}$，$T_3=\{(BN_{31}, BN_{32}), (BN_{32}, d_3), (BN_{31}, d_4)\}$，$T_4=\{(BN_{41}, d_5), (BN_{41}, BN_{42}), (BN_{42}, d_6)\}$。

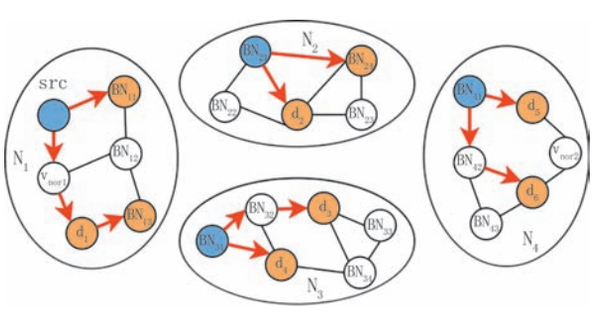


图5 域内组播树

定义5. 跨域组播树T_{cd}：将域间组播树T_{int}和所有域内组播树T_{intra}的所有路径部分并起来就组成了跨域组播树T_{cd}：

$$T_{cd}=T_{int}\cup T_{intra} \tag{5}$$

定义6. 最优跨域组播树：每一棵组播树可以定义为一组从源节点src到n个目的节点$d_i(i=1,\cdots,n)$的路径$p_i(i=1,\cdots,n)$的集合（类似我们之前的工作MADRL-MR[12]中的组播树的定义），可以表示$T=\{p_1,\cdots,p_k,\cdots,p_n\}$，其中$p_k$为组播树$T$中源节点$src$到目的节点$d_k$的路径，$n$是目的节点的数量。如果每条路径$p_k\in T$均有一个最小代价，那么组播树$T$就是一个端到端的最小代价树。定义每条路径$p_k$的成本为$c_k$，其优化目标为最大化$p_k$的瓶颈带宽$bw_k$，最小化$p_k$的时延$delay_k$、丢包率$loss_k$、错包率$err_k$和无线接入点AP间的距离$dist_k$来计算，并且所有指标参数均通过Max-Min归一化[34]到[0,1]之间。每条路径p_k的构造成本可以表示为c_k定义如下：

$$\begin{aligned}\mathrm{cost}(p_k)=c_k=&\beta_1(1-bw_k)+\beta_2 delay_k+\\&\beta_3 loss_k+\beta_4 err_k+\beta_5 dist_k\end{aligned} \tag{6}$$

其中，

$$\begin{aligned}bw_k&=\min_{e_{ij}\in p_k}(bw_{ij})\\delay_k&=\sum_{e_{ij}\in p_k} delay_{ij}\\loss_k&=1-\prod_{e_{ij}\in p_k}(1-loss_{ij})\\err_k&=1-\prod_{e_{ij}\in p_k}(1-err_{ij})\\dist_k&=average\left(\sum_{e_{ij}\in p_k} dist_{ij}\right)\end{aligned} \tag{7}$$

其中，p_k为组播树T中源节点src到d_k的路径，bw_{ij}表示e_{ij}的剩余带宽。$delay_{ij}$表示e_{ij}的时延。$loss_{ij}$表示e_{ij}的丢包率。err_{ij}表示e_{ij}的错包率。$dist_{ij}$表示e_{ij}的距离。

最优跨域组播树就是要找到这样一棵斯坦纳树T_{cd}^*，从这棵斯坦纳树的源节点到所有的目的节点的路径成本c_k之和最小，如公式(8)所示。

$$\begin{aligned}T_{cd}^*&=\mathop{\mathrm{argmin}}_{T_{cd}}\mathrm{cost}(T_{cd})=\\&\mathop{\mathrm{argmin}}_{p_k\in T_{cd}}\sum_{k=1}^{n}\mathrm{cost}(p_k)\end{aligned} \tag{8}$$

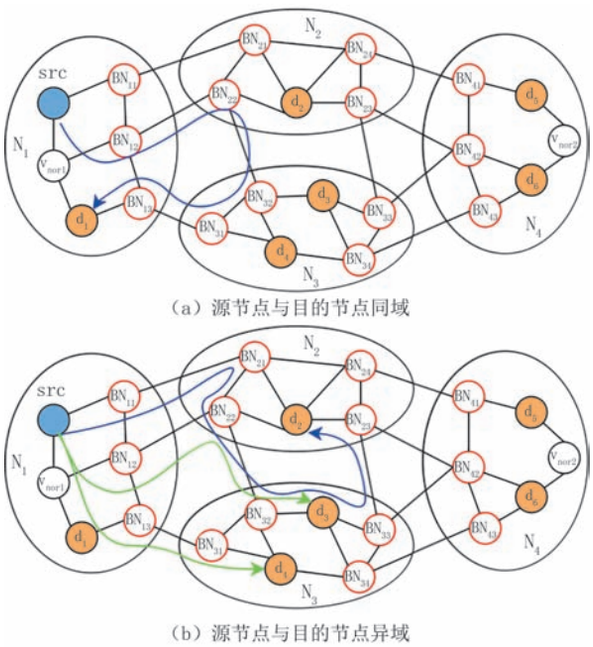
代价最小的组播树T就是最优跨域组播树，依据公式(5)，可以有如下式子成立：

$$\mathrm{cost}(T_{cd})=\mathrm{cost}(T_{int}\cup T_{intra}) \tag{9}$$

通过定义1到定义5，本文给出了最优跨域组播树的定义，依据这些定义，我们知道跨域组播树可以分解为域间组播树和域内组播树来进行求解，依据本节第一段中给出的假设2：“假设任意两个域间的路径开销大于域内任意两个节点之间路径的开销”，我们可以得到如下最优跨域组播树的性质：最优组播树路由路径不存在域间路径环路，接下来我们结合图6，给出这个性质的分析，然后给出其相关定理和数学证明。

对图6中的各种从源节点到任一个目的节点的情形，逐一分析如下：

(1)当组播源和其中任意一个组播目的节点在同一域中时，如图6(a)所示，src与目的节点d_1处于同一



（a）源节点与目的节点同域

（b）源节点与目的节点异域

图6　多域场景下源节点与目的节点同域和异域

个域N_1中，如果src到目的节点d_1的路径是图6(a)中蓝色标注的路径情况，经过域N_1, N_2, N_3再回到N_1，从而形成域间路径环路$N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_1$这样的情形，通常这样的回路路径较长，涉及不同域间控制器之间的协同操作，带来额外多域的开销，依据假设2可以判断域间路径环路$N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_1$的开销大于从源节点$src$在域$N_1$中直接经过域$N_1$内交换机达到目的节点的开销，因此同一个域中的源节点到同域的目的节点的跨域路径是不存在，即认为src到d_1的路径上的节点都位于域N_1内。

(2)另一种情况是，源节点与目的节点不在同一个域中，如图6(b)所示，src在域N_1中，d_2在域N_2中，d_3, d_4在域N_3中，如图6(b)中的蓝色标注路径，从N_1经过N_2, N_3再回到N_2，再到达目的节点d_2，从而也形成了域间路径环路$N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_2$，类似图6(a)中的分析，依据假设2，从$src$到达域$N_2$后也是不会再离开域$N_2$再回域$N_2$的，因此从$src$到达目的节点$d_2$也不会发生域间路径环路。

(3)从源节点到达同一域中的多个不同目的节点，如图6(b)中绿色标注路径所示，目的节点d_3和d_4处于同一个域N_3，但src到d_3通过边界节点BN_{12}经过N_2的边界节点BN_{22}再到N_3，而src到d_4直接通过边界节点BN_{13}到N_3，这样在域N_1与N_3之间就存在两条不同的跨域路径$N_1 \rightarrow N_3$和$N_1 \rightarrow N_2 \rightarrow N_3$，如果这两条路径的开销分别用$\mathrm{cost}(N_1 \rightarrow N_3 \rightarrow d_3)$和$\mathrm{cost}(N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow d_4)$表示，则有

$$\begin{aligned}&\mathrm{cost}(N_1 \rightarrow N_3 \rightarrow d_3) + \mathrm{cost}(N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow d_4) = \\ &\quad \mathrm{cost}(N_1 \rightarrow N_3) + \mathrm{cost}(BN_{32} \rightarrow d_3) + \\ &\quad \mathrm{cost}(N_1 \rightarrow N_2 \rightarrow N_3) + \mathrm{cost}(BN_{31} \rightarrow d_4) > \\ &2 \times \min\left(\mathrm{cost}(N_1 \rightarrow N_3), \mathrm{cost}(N_1 \rightarrow N_2 \rightarrow N_3)\right) + \\ &\quad \mathrm{cost}(BN_{32} \rightarrow d_3) + \mathrm{cost}(BN_{31} \rightarrow d_4)\end{aligned}$$

最后一个大于符号的判断的依据也是假设2。由此可以判断从源节点到达同一域中的多个不同目的节点不会走不同域间路径。

总结以上几种情形的分析，可以有如下定理成立：

定理1.　对最优跨域组播树$T_{cd} = T_{int} \cup T_{intra}$上的路径有：

(1) 对任意一条从源节点src到目的节点d_i域间路径序列：$PN_i = <N_{i,1}, N_{i,2}, \cdots, N_{i,j}, \cdots, N_{i,k}, \cdots, N_{i,|PN_i|}>$，对$\forall j,k$，当$j \neq k$时，有$N_{i,j} \neq N_{i,k}$成立。

(2) 对任意两个目的节点d_i和d_j，当$\mathrm{Domain}(d_i) = \mathrm{Domain}(d_j)$时有$PN_i = PN_j$成立。

证明. (1)该结论表示的就是域间路径序列PN_i进入到某一个域N_j后不会再经过其他域后再回到域N_j，我们可以采用反证法证明结论的成立，假设定理1不成立，如果存在j,k，当$j \neq k$时，有$N_{i,j} = N_{i,k}$成立，则$\mathrm{cost}(PN_i) = \mathrm{cost}(N_{i,1} \rightarrow N_{i,j}) + \mathrm{cost}(N_{i,j} \rightarrow N_{i,k}) + \mathrm{cost}(N_{i,k} \rightarrow N_{i,j}) > \mathrm{cost}(N_{i,1} \rightarrow N_{i,j}) + \mathrm{cost}(N_{i,k} \rightarrow N_{i,j})$，这说明存在一条到目的节点$d_i$开销更少的域间路径序列，与$T_{cd}$为最优跨域组播树的前提矛盾，因此原命题(1)成立。

(2)该结论表示从源节点到达同一域中的不同目的节点不会走不同域间路径。假设$PN_i = <N_{i,1}, N_{i,2}, \cdots, N_{i,k}, \cdots, N_{i,|PN_i|}>$和$PN_j = <N_{j,1}, N_{j,2}, \cdots, N_{j,k}, \cdots, N_{i,|PN_i|}>$，他们分别达到目的节点$d_i$和$d_j$，由于$\mathrm{Domain}(d_i) = \mathrm{Domain}(d_j)$，这两个目的节点处于同一个域$N_h$，$PN_i$通过域$N_h$边界节点$BN_{h1}$进入到域$N_h$再到达$d_i$，$PN_j$通过域$N_h$边界节点$BN_{h2}$进入到域$N_h$再到达$d_j$，则这两条路径总共的开销为

$$\begin{aligned}&\mathrm{cost}(PN_i) + \mathrm{cost}(PN_j) = \\ &\quad \mathrm{cost}(PN_i) + \mathrm{cost}(BN_{h1} \rightarrow d_i) + \mathrm{cost}(PN_j) + \\ &\quad \mathrm{cost}(BN_{h2} \rightarrow d_j)。\end{aligned}$$

不是一般性，假设PN_i是两条路径PN_i和PN_j开销最小的那条，又由假设2，对于域N_h内的两个节点BN_{h1}和d_j，$\mathrm{cost}(BN_{h1} \rightarrow d_j)$小于任意一条域间链路的开销，因此上式我们有

$$\begin{aligned}&\text{cost}(PN_i)+\text{cost}(PN_j)=\\&\text{cost}(PN_i)+\text{cost}(BN_{h1}\rightarrow d_i)+\text{cost}(PN_j)+\\&\text{cost}(BN_{h2}\rightarrow d_j)\geqslant\text{cost}(PN_i)+\\&\text{cost}(BN_{h1}\rightarrow d_i)+\text{cost}(BN_{h1}\rightarrow d_j)。\end{aligned}$$

由于只有是同一条路径，即$PN_i=PN_j$时，代价开销$\text{cost}(PN_i)$才会仅计算一次，因此，等号当且仅当$PN_i=PN_j$才成立。

这说明从源节点出发，可以经过PN_i和PN_j开销最小的那条（比如前面假设的PN_i）到达域N_h边界节点BN_{h1}进入到域N_h再分别到达目的节点d_i和d_j的开销是最小的。因此，对跨域组播树T_{cd}，在当$\text{Domain}(d_i)=\text{Domain}(d_j)$时，只有$PN_i=PN_j$时，$T_{cd}$才是最优跨域组播树。因此原命题(2)也成立。证毕。

定理2. 对最优跨域组播树$T_{cd}=T_{int}\cup T_{intra}$的路径经过的任何一个域，域内只存在域内的域内组播树，不存在域内组播森林。

从定理1结论及其证明中可以看出，进入任何一个域的边界节点只有一个，说明只会有一棵树存在，不会有多棵树存在，因此不存在域内组播森林，限于篇幅，本文这里就没有继续给出证明过程，但证明的过程其实与定理1(2)的证明是类似的。

从以上假设1、假设2，和定义1～定义6，以及定理1、定理2，我们可以知道：由于域间路径的通信会涉及不同控制器的协同操作，增加域间通信数量会增加协同交互成本，因此域间路径开销远大于域内路径开销，并且每个域内的可选路径有足够冗余度能保证域内网络节点间的连通性，组播路径跨越的任意两个域之间只会存在一条域间链路，这才降低网络流量等组播开销，本文后续研究都是在基于此结论下进行。

最后给出本文对定义的最优跨域组播树的代价开销的具体形式。

假设域间组播树和每棵域内组播树分别为最小代价树，最小代价树的定义方式和定义5最优跨域组播树相同，均表示为一组从源节点src到n个目的节点$d_i(i=1,\cdots,n)$的路径$p_i(i=1,\cdots,n)$的集合，$T=\{p_1,\cdots,p_k,\cdots,p_n\}$。求$T$中每条路径$p_k$的最小代价$c_k$，从而求得$T$的最小代价。假设每个域内的域内组播树$T_i$的构建成本表示为$C_i$，域$N_i$内的目的节点（如果该域有离开该域的边界节点，则该边界节点也成了该域内组播树T_i的目的节点）个数用n_i表示，则有：

$$C_i=\sum_{k=1}^{n_i}c_k \tag{10}$$

域间组播树的成本C_{int}计算方式与域内组播树也相同。则跨域组播树的总成本$\text{cost}(T_{cd})$由域内和域间组播树的成本组成，最小化其总成本就可以如下表示：

$$\min\text{cost}(T_{cd})=\sum_{i=1}^{m}C_i+C_{int} \tag{11}$$

其中，$\sum_{i=1}^{m}C_i$表示对所有域内组播树的构建成本求和，m表示域的个数，C_{int}表示域间组播树的构建成本。

4 SDWN多控制器域智能跨域组播路由体系架构

SDWN多智能体跨域组播路由策略是将跨域组播树的构建分解成域间组播树和多棵域内组播树的构建。通过多智能体协作实现跨域组播路由。其架构如图7所示，下面进行详细解释。

控制平面中每个域设有一个本地控制器，通过本地控制器周期性地获取对应域的网络状态信息，并通过控制器通信机制将本地控制器收集的信息同步到根控制器，由根控制器管理全局网络资源。②将控制器平面收集的全局网络链路信息（NLI，Network Link Information）处理并储存在知识平面中。③知识平面中多智能体通过感知学习不同时刻的NLI进行训练。④由域间智能体通过全局网络信息决策出域间组播树，即多域之间的路径和边界节点的选择。再通知到每个控制器域的智能体完成域内组播树的构建。最终由多个智能体协作完成最优跨域组播树的构建。⑤将决策出的跨域组播树同步到根控制器和本地控制器。⑥在下一网络流量到来之前，由控制平面向数据平面的无线接入节点下发并安装流表。最后由数据平面完成流量的转发。

4.1 数据平面

数据平面负责处理和转发网络中的数据包，其包含多个网络子域。每个域由无线接入节点（Access Point，AP）和站点（Station，STA）组成。每个域中的AP通过无线Mesh方式组成一个多跳的无线网络[35]，AP分为域内节点（Intra-domain node，IN）和边界节点（Boundary node，BN），每个域内AP下连接着一个STA。由AP接收来自终端设备的无线数据包，根据控制平面下发的指令和策略，选择最佳的路由路径，将数据包从源设备转发到目标设备。每个域周期性地与本地控制器交互，将当前域

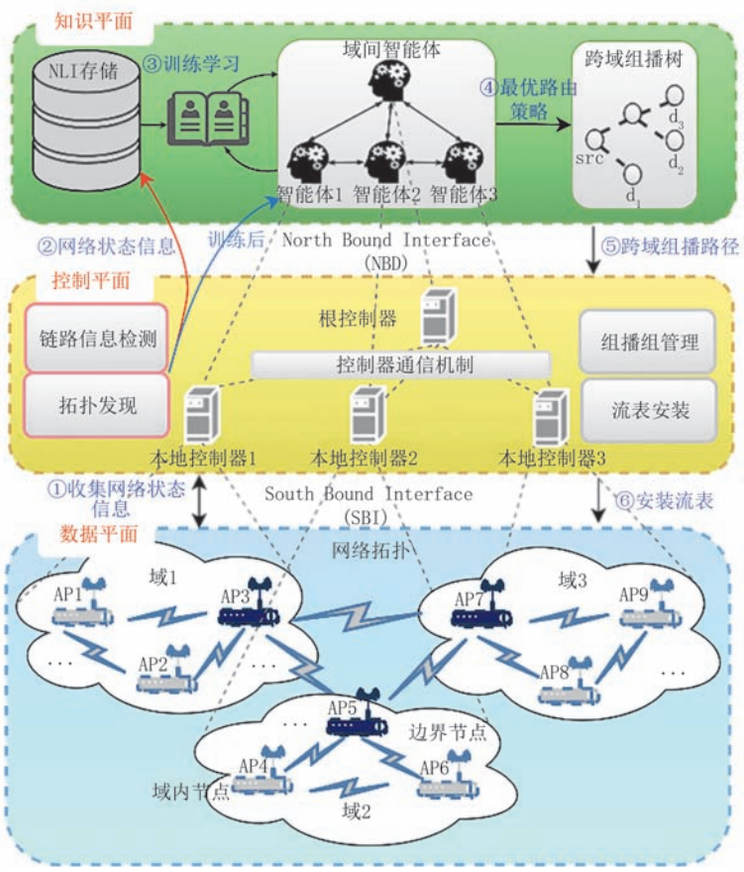


图7　SDWN多智能体跨域组播路由结构

的无线网络链路信息传递给控制平面。

4.2　控制平面

控制平面负责收集和分析网络状态信息，制定决策，并将命令和策略下发到数据平面中的AP，其中包含一个根控制器和多个本地控制器。本地控制器通过南向接口与数据平面中的AP进行通信，将对应域的网络状态信息通过控制器通信机制（Controller Communication Mechanism，CCM）同步到根控制器。由根控制器构建网络的全局视图，并管理和调度全局网络资源。控制平面通过北向接口与知识平面交互，方便知识平面策略的下发和部署。其中包括网络拓扑发现、链路信息检测、控制器通信机制、组播组管理和流表安装等功能。

网络拓扑发现和链路信息检测是通过控制器向数据平面发送数据包来获取相关信息。其中，网络拓扑发现是周期性的发送LLDP（LLDP，Link Layer Discovery Protocol）[36]请求数据包，AP进行回复时，会将设备的端口连接、id等信息封装到Reply数据包中，最终由控制器解析构建网络拓扑。链路信息检测是通过发送PortStatsRequest请求数据包来获取端口信息。控制器解析回复消息得到发送数据包数tx_p、接收数据包数rx_p、发送字节数tx_b、接收字节数rx_b、发送错包数tx_{err}和接收错包数rx_{err}，以及端口发送字节的持续时间t_{dur}。通过以上参数计算出剩余带宽bw_{ij}、已用带宽ubw_{ij}、丢包率$loss_{ij}$和错包率err_{ij}等网络链路信息，计算公式如下：

$$ubw_{ij}=\frac{\left|\left(tx_{bi}+rx_{bi}\right)-\left(tx_{bj}+rx_{bj}\right)\right|}{t_{durj}-t_{duri}} \tag{12}$$

$$bw_{ij}=bw_{max}-ubw_{ij}$$

$$loss_{ij}=\frac{tx_{pi}-rx_{pj}}{tx_{pi}} \tag{13}$$

$$err_{ij}=\frac{tx_{erri}+rx_{errj}}{tx_{pi}+rx_{pj}}\cdot 100\% \tag{14}$$

其中，tx_{b*}、rx_{b*}和t_{dur*}分别代表节点*发送字节数、接收字节数和持续时间。tx_{p*}、tx_{err*}和rx_{p*}、rx_{err*}分别代表节点*发送的数据包数、错包数和接收的数据包数、错包数。

在SDN[37]中，由于两台交换机通信需要经过控制器转发，所以两台交换机的网络链路时延需要近似计算。控制器通过解析经过链路的数据包的时间戳信息，得到控制器到两个交换机的链路往返时延RTT_1和RTT_2，以及根据LLDP报文计算出两台交

换机的正向传输时延T_{fwd}和回复传输时延T_{re}，由此近似地计算链路的时延$delay_{ij}$，如公式(15)所示。

$$delay_{ij}=\frac{\left(T_{fwd}+T_{re}-RTT_1-RTT_2\right)}{2} \tag{15}$$

此外，再通过无线AP的部署坐标所计算出的两个AP间的距离$dist_{ij}$。由以上计算出的网络链路信息(NLI)经过Max-Min归一化[34]后存入知识平面的NLI Storage。

控制器通信机制保证了本地控制器与根控制器之间稳定且快速的通信。传统的SDN多域通信是通过MBGP协议，然而该协议在SDN环境中配置繁琐，且会在消息更新时可能造成导致多个控制器之间的消息的不一致。本文基于RESTful API设计了一个本地控制器与根控制器间的通信机制。这样设计实现的CCM实现了松耦合，每个控制器可以独立开发和部署，提高了可维护性和可扩展性。同时，它具备可移植性，可以在不同的环境和平台上运行。具有较好的灵活性，使得根控制器和本地控制器能够以统一的方式进行交互，并适应不同的网络环境和需求。实现这样的CCM离不开RESTful API的支持。RESTful API是标准化的，具有明确定义的规范和约束，减少了兼容性问题，它具有很好的可扩展性，能够轻松地扩展和增强通信能力。不仅如此，由于RESTful架构还是一种无状态、基于HTTP的分布式设计，天然支持高并发，每个请求独立处理，服务器无需维护会话状态，能简化系统并提升了扩展性。它还支持水平扩展，通过负载均衡和分布式部署提高并发能力，同时利用HTTP缓存机制减少服务器压力。在SDN控制器实现中，RESTful API广泛被用于了流表管理、设备配置、网络监控和多个控制器间的协同工作(如ONOS、OpenDaylight和Ryu)，使得网络管理更加高效、灵活。当然SDN控制器设计实现也需要看应用场合需求，如果是核心网或者数据中心网络，还需要硬件编程平台来实现。由于我们这里侧重算法层次的验证，因此选用RESTful架构实现SDN控制器是足够了的。

组播组管理(Multicast group management，MGM)功能主要是分析出组播源和组播目的节点所在的域，以及管理组播节点的加入与离开。组播组管理功能通过监测网络流量和分析数据包，确定组播源和组播目的节点所在的域，并协调各个域之间的通信，确保组播流能够正确地跨域传输。同时可以与各个域中的控制器进行通信，向其提供组播流的相关信息，并协调组播流在各个域之间的传输路径。此外，可以根据组播用户的请求判定加入与离开，并对组播树进行更新。

(1)节点加入：当第k域N_k中有节点v_i发出动态加入请求$req_k=(v_i, add)$，对于N_k获取当前组播树T_k和网络拓扑信息，若v_i不在T_k中，则向当前域的智能体发出请求，构建v_i到T_k的最小成本路径p_i，并更新$T_k \leftarrow T_k \cup p_i$；同时标记$v_i$为在线组播成员节点接收数据。

(2)节点离开：当N_k中组播组成员v_i发出节点离开请求$req_k=(v_i, leave)$，获取当前组播树T_k，找到v_i到T_k路径p_i，并对其进行剪枝操作。执行$T_k \leftarrow T_k - p_i$，删除对应流表，并将v_i标记为离线节点停止向其发送数据。

流表安装功能通过SDN北向接口接收知识平面的指令，下发组播流表。在接收到节点加入和离开请求时，修改或者删除现有的组表项，实现组成员的加入和删除，确保数据的准确转发。

4.3 知识平面

知识平面是在SDN架构上新增的一个重要组成部分，本文的多智能体深度强化学习智能跨域组播路由算法就是运行在该平面。它包括存贮由控制平面收集数据平面的网络链路信息的NLI Storage；并且需要将NLI处理成流量矩阵TM提供给智能体训练学习。由多个智能体经过训练后协作完成跨域组播树的构建，再下发指令到控制平面，由控制平面向数据平面下发流表项。

依据本文第三节对跨域组播问题的描述与建模，我们通过构建具有最小成本的斯坦纳树来求解组播问题，将跨域组播树分解为域间组播树和多个域内组播树等多个组播树构建问题。为了与构建域内组播树的智能体进行区分，我们将构建域间组播树的智能体称为域间智能体。域间智能体的状态空间和奖励函数与域内智能体相同，但动作空间不相同。域间智能体的动作空间为不同域之间连接边的集合，而域内智能体的动作空间是域内节点集合。这是因为在构建域间组播树时，我们将域抽象为节点，但相邻域之间可能多条连接的边，因此不能以节点集合作为动作空间。具体设计本文将在第五节详细描述。下面先通过两个简单的例子介绍多个智能体协作构建跨域组播树的过程。

Example 1：图8是由4个域构成的多域网络拓扑，相邻域之间通过边界节点BN相连接，整体表示的

是域间组播树的构建过程。其中，图8(a)中标注源节点和目的节点，它们分别处于不同的域中。图8(b)中我们将域抽象为节点，N_1为源域，N_2，N_3，N_4为目标域，并且只保留相邻域连接的边以及边界节点，设定每条边的权重。我们将相邻域间的边作为域间智能体的动作，以最小成本为目标。域间智能体将会选择(BN_{11}，BN_{21})、(BN_{13}，BN_{21})、(BN_{24}，BN_{41})和(BN_{34}，BN_{43})4条边来连接源域和目标域，这样构造的域间组播树的成本最小$C_{int}=6$，并且确定了每个域到达其他域的边界节点。最终构建的最小成本域间组播树T_{int}，如图8(c)所示。

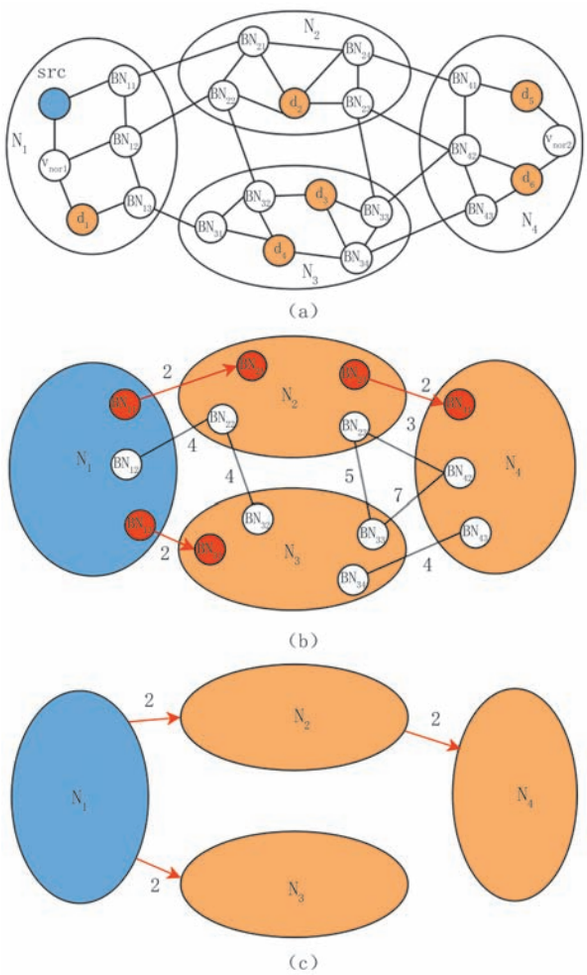


图8　域间组播树构建

Example 2：图9展示的是域内组播树的构建过程。先通过构建域间组播树，选定跨域的边界节点。如图9(a)所示，根据图8中构建的域间组播树，我们将选择的边界节点标红。这些边界节点在每个域中可以看成源节点和目的节点。如图9(b)所示，我们将BN_{11}，BN_{13}等同于域N_1的目的节点；将BN_{21}等同于N_2中的源节点，BN_{24}等同于目的节点；将BN_{31}等同于域N_3的源节点；将BN_{41}等同于域N_4的源节点。再由相对应的域内智能体构建源节点到目的节点的域内组播树T_1，T_2，T_3和T_4。最终每个域构建的域内组播树如图9(c)所示。

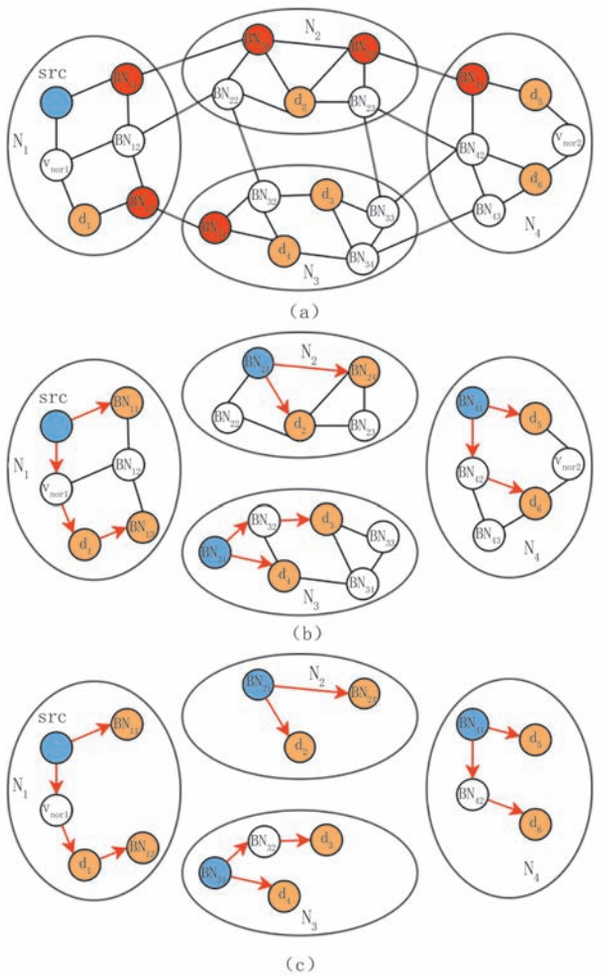


图9　域内组播树构建

通过以上两个简单的例子展示了域间组播树和多个域内组播树的构建过程。首先通过域间智能体构建域间组播树，在确定了每个域的边界节点后，再由相对应的域去构建域内组播树。最后将域间组播树和多个域内组播树相结合，即得到跨域组播树，如图10所示。

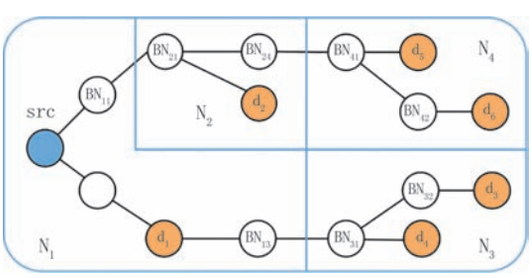


图10　跨域组播树

5 MA-CDMR多智能体跨域组播路由算法

依据前面4.3节知识平面中关于组播树求解思路的设计，本文给出的域间组播树求解算法MA-CDMR的流程图如图11所示。首先通过多控制器SDWN架构获取全局网络拓扑和链路信息，然后将这些网络信息提供存储到知识平面的NLI Storage中，提供给智能体训练学习。其次，先通过域间智能体感知全局网络信息，决策出最佳域间组播树，从而确定每个域通往邻域的边界节点。然后再将信息同步到每个域的域内智能体，由相应域的域内智能体完成最佳域内组播树的构建。最后将最佳域间组播树和多个域内组播树组合，生成最佳跨域组播树，得到全局网络的最佳跨域路由的转发路径。

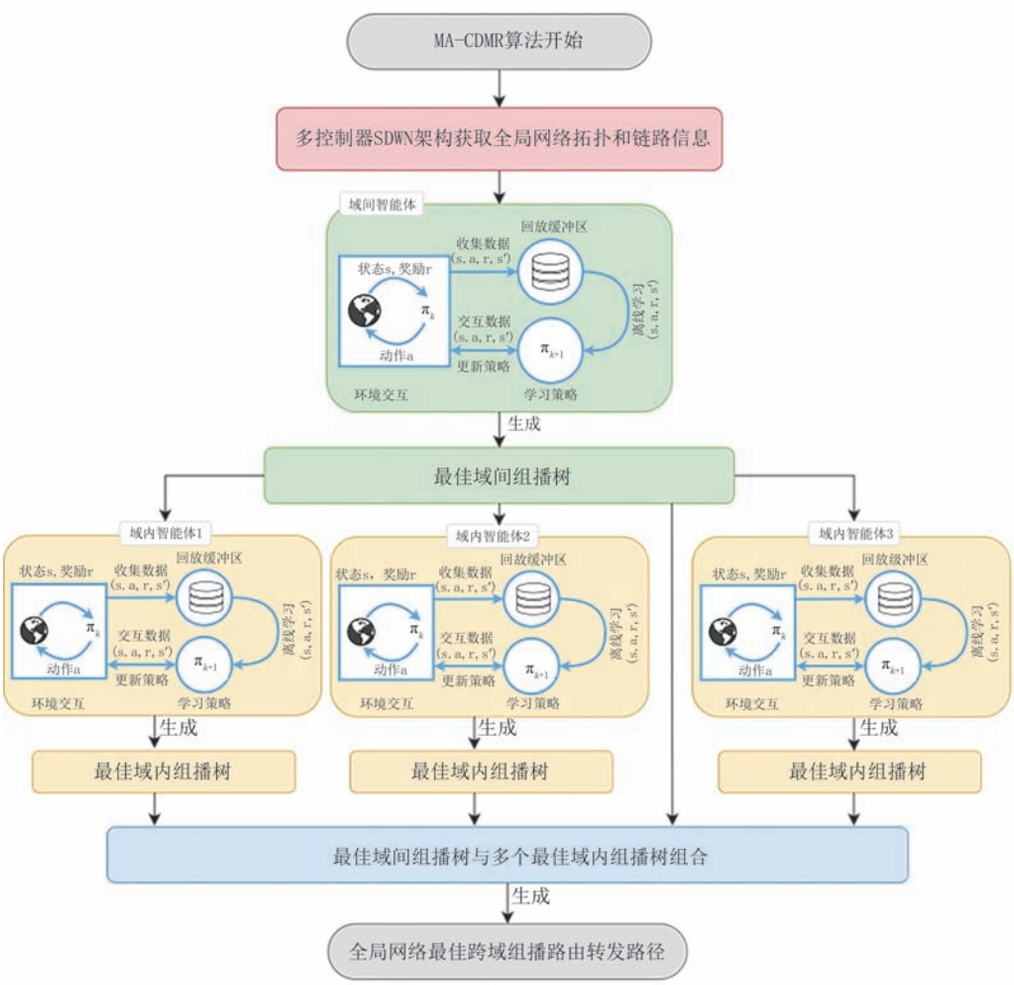


图11 MA-CDMR算法流程图

下面详细介绍MA-CDMR算法，分别从智能体的状态空间、动作空间和奖励函数，以及多智能体的训练策略等设计进行详细介绍。最后再介绍算法伪代码。

5.1 智能体设计

在MA-CDMR算法中我们设计了两种智能体，一种是构建域间组播树的域间智能体，另一种是构建域内组播树的域内智能体。域间智能体只有一个，而域内智能体是每个域均有一个。两种智能体的状态空间和奖励函数是相同的，但动作空间不相同。下面进行详细介绍。

(1)状态空间

智能体的状态空间决定着它感知和理解环境的能力，应该包含足够的环境信息使智能体做出有意义的决策。因此，我们以网络链路的剩余带宽***bw***、时延***delay***、丢包率***loss***和错包率***err***等流量矩阵，以及组播树的构建状态矩阵$\boldsymbol{M}_T$组成一个多通道矩阵$\boldsymbol{X}=[\boldsymbol{bw},\boldsymbol{delay},\boldsymbol{loss},\boldsymbol{err},\boldsymbol{dist}]$作为智能体的状态$s$。其多通道的状态矩阵如图12所示。

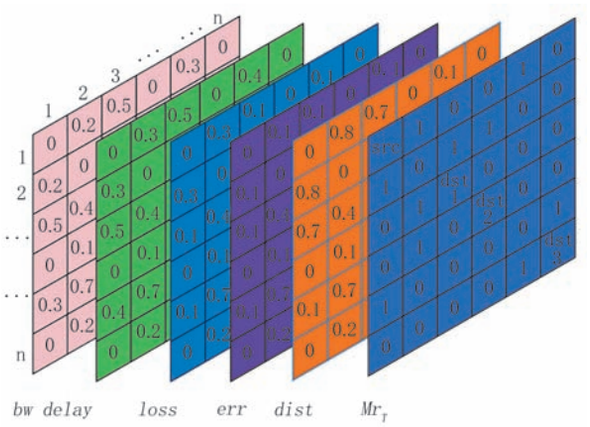


图12　智能体的状态矩阵

多通道矩阵X变化的所有可能性的集合为智能体的状态空间$\boldsymbol{S}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n\}=\{s_1,s_2,\cdots,s_n\}$。例如选择了某条边加入组播树中，则智能体的状态从$s_t$变为$s_{t+1}$。通过这样设计的状态空间可以提供关于网络链路和节点之间的重要信息，以及组播树构建的具体情况，使智能体能够根据当前网络状况做出相应的决策，选择出最佳路径来构建组播树。

(2)动作空间

动作空间决定了智能体可以执行的操作范围和粒度。动作空间的设计应该尽可能包括智能体需要采取的所有可能的行动，并且适应环境的特点和任务的要求。因此我们分别对域间智能体和域内智能体的动作空间进行设计。

域间智能体的动作空间$\mathcal{A}_{int}$：我们在构建域间组播树时，是将每个域抽象为一个节点，而这个节点与相邻节点可能有多条连接的边，所以我们以所有域之间连接的边的集合作为域间智能体的动作空间，$\mathcal{A}_{int}=\{(BN_{11},BN_{21}),(BN_{12},BN_{32}),\cdots,(BN_{ia},BN_{jb})\}=\{e_1,e_2,\cdots,e_k\}$，其中$(BN_{ia},BN_{jb})$表示连接域$N_i$和域$N_j$的边，$BN_{ia}$表示域$N_i$中的第$a$个边界节点，$k$是不同域之间边的总数。其中以当前域与领域连接的边为有效动作，其余为无效动作。通过这样设计能够使智能体选择出最适合跨域的边界节点和路径，从而构建出最佳域间组播树。

域内智能体的动作空间$\mathcal{A}_{intra}$：由域内节点之间连接的情况复杂且连接的边较多，为了降低动作空间的复杂度，我们以域内节点的集合作为域内智能体的动作空间，$\mathcal{A}_{intra}=\{v_1,v_2,\cdots,v_n\}=\{a_1,a_2,\cdots,a_n\}$，节点$v$和动作$a$一一对应。其中以当前节点的下一跳节点作为有效动作，其余为无效动作。通过这样设计可以减少智能体的无效探索，能够更加快速地决策出最优路径。

(3)奖励函数

奖励函数决定了智能体在学习过程中如何评估行动的好坏，它能够给智能体的行为提供明确的反馈，引导智能体朝向期望的目标前进。而我们的优化目标是最大化组播树剩余带宽，最小化时延、丢包率、错包率和无线接入点AP间的距离。为此，我们以网络链路信息参数来设计奖励函数。由于我们是以链路或节点的集合来设计动作空间，对于不同时刻的状态s，会产生有效动作和无效动作。而智能体执行一个有效动作a，将状态s_t转变为s_{t+1}，可能会产生两种不同的正反馈和一种负反馈。执行一个无效动作则会产生一种负反馈。下面分别对不同的情况设计奖励函数。

如果执行当前动作a_t后，添加的下一跳节点或者链路只是给组播树添加了一个普通节点，为这种正反馈我们设计了一个单步奖励R_{part}，以加入组播树的链路e_{ij}的剩余带宽bw_{ij}、时延$delay_{ij}$、丢包率$loss_{ij}$、错包率err_{ij}和AP之间的距离$dist_{ij}$来计算奖励。如公式(16)所示。

$$R_{part}=\beta_1 bw_{ij}+\beta_2(1-delay_{ij})+\beta_3(1-loss_{ij})+\beta_4(1-err_{ij})+\beta_5(1-dist_{ij}) \tag{16}$$

如果添加的下一跳节点或者链路是给组播树添加了一个目的节点d_k，为这种正反馈我们设计了一个子任务奖励R_{end}。以源节点src到该目的节点d_k的整条链路的剩余带宽bw_k、时延$delay_k$、丢包率$loss_k$、错包率err_k和链路的平均距离$dist_k$来计算奖励。如公式(17)所示。

$$R_{end}=\beta_1 bw_{\mathrm{k}}+\beta_2(1-delay_{\mathrm{k}})+\beta_3(1-loss_k)+\beta_4(1-err_k)+\beta_5(1-dist_k) \tag{17}$$

如果添加的下一跳节点或者链路使得组播树形成环路，为这种负反馈一个惩罚值$R_{loop}=C_1$。

如果a_t是一个无效动作，既不是当前节点的下一跳节点，也不是当前域与邻域连接的边，对于这种负反馈我们也给予一个惩罚值$R_{hell}=C_2$。

(4)智能体的策略更新

本算法中所有的智能体均使用的是Actor-Critic框架[38]。该框架分为两个部分：Actor(策略网络$\pi_\theta(a|s)$，参数为θ)和Critic(价值网络V_ω，参数为ω)。其中策略网络Actor是与环境交互，并在Critic价值函数的指导下用策略梯度学习一个更好的策略。其采用的是策略梯度更新。如公式(18)所示。

$$\nabla_\theta J(\theta)=\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T}\psi_t\nabla_\theta\log\pi_\theta(a_t|s_t)\right] \quad (18)$$

其中，T是和环境交互的最大步数，$\pi_\theta(a|s)$是策略函数，a_t是当前的动作，s_t是当前的状态。ψ_t是时序差分残差[39]，用于指导策略梯度学习，其计算方式如公式(19)所示。

$$\psi_t=r_t+\gamma V_\omega(s_{t+1})-V_\omega(s_t) \quad (19)$$

其中，r_t为本回合获得的奖励值，γ是衰减因子，$\gamma\in[0,1]$。$V_\omega(s_t)$是在状态s_t下，执行动作a_t产生的期望值。

Critic价值网络V_ω采用时序差分残差的学习方式，对于单个数据定义如下价值函数的损失函数，如公式(20)所示。

$$\mathcal{L}(\omega)=\frac{1}{2}\left(r+\gamma V_\omega(s_{t+1})-V_\omega(s_t)\right)^2 \quad (20)$$

将上式中$r+\gamma V_\omega(s_{t+1})$作为时序差分目标，不会产生梯度来更新价值函数。因此，价值函数的梯度如公式(21)所示。

$$\nabla\mathcal{L}(\omega)=-\left(r+\gamma V_\omega(s_{t+1})-V_\omega(s_t)\right)\nabla_\omega V_\omega(s_t) \quad (21)$$

5.2 多智能体的训练策略设计

多智能体的训练情况相比于单智能体更加复杂，因为每个智能体在与环境交互的同时也在和其他智能体进行直接或间接的交互，并且每个智能体会不断学习并更新自身策略。因此对于每个智能体而言，环境是非稳态的。另一方面，对于多域环境而言，每个域的智能体的训练目标不同，不同的智能体需要最大化自身的奖励，这可能需要大规模的分布式训练来提升效率。针对以上问题，本文采用完全去中心化(fully decentralized)的训练方法，其也被称为独立学习(IL，Independent learning)[40]。

在MA-CDMR算法中，对于每个域内智能体而言，全局的网络环境是非稳态的，但其负责的子域网络环境是稳态的。对于域间智能体，它的策略执行优先于域内智能体，且面向的是全局网络，所以其环境也是稳态的。因此，采用IL的训练方式，每个智能体都可以在自身的环境中独立地学习，不考虑其他智能体的改变。并且随着网络的扩大和智能体的增加具有较好的可扩展性。

为了提升多智能体的收敛速度，降低在大规模网络环境中的训练成本，本文在Online强化学习的基础上，结合离线强化学习的优势，设计了一个离线和在线相结合的训练方法。为显示其他三种训练策略与本文设计的训练策略相区别，我们将其以图片的形式展示，如图13所示。

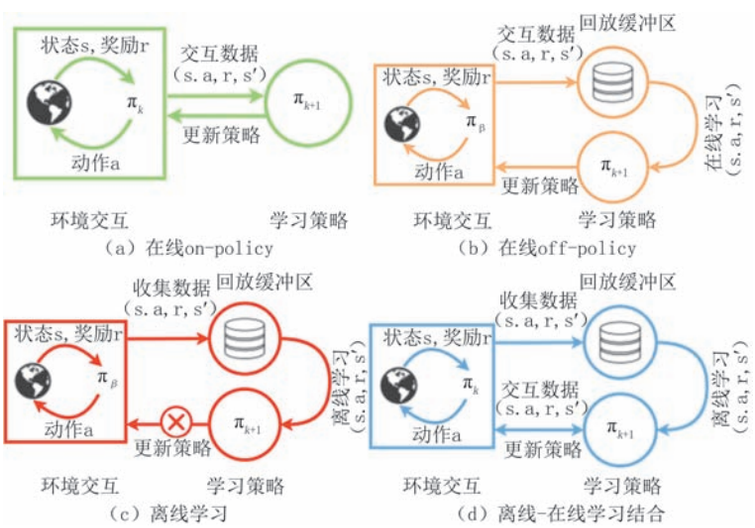


图13 四种训练策略

图13(a)，13(b)分别是在线强化学习的on-policy和off-policy[41]，均需要与环境实时交互，其中on-policy与环境交互和使用数据更新的策略为同一个策略π_k。而off-policy与环境交互产生数据策略π_β和使用数据的策略π_k为不同的策略。图13(c)为离线强化学习的训练策略[16]，通过在训练模型之前收集数据集，再使用这些数据集离线训练智能体，与off-policy同样为异策略，但不与环境交互更新策略。图13(d)为本文结合在线和离线的优势设计的训练策略，与环境交互和使用数据的是同一策略。首先收集一定的数据，然后使用离线数据训练一个初始策略，再使用在线强化学习中的on-policy与环

境进行交互，在实际环境中对该策略进行改进。通过离线和在线混合训练策略可以减少在线学习的需求，减少在线强化学习中与实时环境的交互频率，提高数据利用效率，并且可以更好地应对实际环境中的动态变化。

5.3　MA-CDMR算法设计

MA-CDMR算法将跨域组播路由分为域间组播树和域内组播树构建等两个阶段，首先通过域间智能体感知全局网络环境决策出域间组播树，选择出最佳的域间转发路径以及每个域转发的边界节点。再将信息同步至每个域内智能体，由域内智能体开始构建域内组播树。最终将域间组播树和多个域内组播树组合，生成最佳跨域组播树。对于组播用户的加入与离开，分别进行组播路径的构建和组播树的剪枝。MA-CDMR算法的详细实现见算法1。

算法输入多域的全局网络拓扑$G(V,E)$，网络链路信息NLI，组播节点集合$M=\{\text{src}\}\cup D$，以及深度强化学习的各种超参数。算法输出从源节点到所有域的目的节点的最优跨域组播树。第1-3行是初始化相关参数；第4行是分析控制器收集的信息，通过组播组管理模块确定源节点和各目的节点所在的域。第5-14行是离线训练智能体的函数，目的是得到将离线训练的Actor网络和Critic网络，并将其返回；第17-18行重置环境得到域间智能体状态s_t，进入域间智能体循环；第20-22行是域间智能体从s_t输出的动作集合中采样出a_t；执行动作a_t获得奖励值r_t和下一状态s_{t+1}，收集离线数据(s_t,a_t,r_t,s_{t+1})存入域间数据缓存区B_{int}，此时不与环境进行交互，对域间智能体进行离线训练；第24行是将构建的域间组播树同步到每个域内智能体，确定每个域的转发边界节点，进入域内智能体循环；第26-28行是每个域内智能体根据得到状态s_t，同时从自身s_t输出的动作集合中采样出a_t，执行动作a_t获得奖励值r_t和下一状态s_{t+1}；第29-31行是收集域内智能体训练离线数据得到对应的域内数据缓存区B_{intra}，对每个域内智能体进行离线训练；第33-34行是每个域内智能体判断自身的域内组播树T_{intra}是否构建完成，若构建完成退出循环，否则对应域内智能体更新状态$s_t \leftarrow s_{t+1}$；第35-38行域内智能体与环境进行交互，在线更新网络参数，改进由离线训练的策略；第41-46行是域间组播树未构建完成，域间智能体更新状态$s_t \leftarrow s_{t+1}$，并与环境进行交互，在线训练调整域间策略。第47-50行判断域间组播树和所有的域内组播树是否均构建完成，如果未完成，继续循环；如果构建完成则组合T_{int}和T_{intra}构建跨域组播树，并退出循环进入下一个回合ep。

算法1的训练时间复杂度主要受训练回合数、神经网络规模和训练步数的影响，通常表示为$O(N\cdot M\cdot T\cdot|\theta|)$，其中$N$表示训练回合数，$M$是域的数量，同时$M+1$表示智能体数量，$T$是训练步数，$|\theta|$是参数数量，但域的数量通常较小，并且我们采用的是完全去中心化的训练方式，每个域的智能体是单独进行训练，所以域的数量增加并不会影响单个域智能体的训练性能。测试时间复杂度则为$O(T'\cdot|\theta|)$，这里T'是测试步数。这表明算法1只是训练复杂度较高，而测试复杂度较低，只受网络大小和环境步数的影响。

算法1.　MA-CDMR

输入：全局网络拓扑$G(V,E)$，网络链路信息NLI，组播节点集合$M=\{\text{src}\}\cup D$，权重因子β_l，$l=1,2,\cdots,5$，Actor网络学习率α_1，Critic网络学习率α_2，衰减因子γ，离线训练批次的批量大小k，更新频率n_{update}，训练迭代回合数$episodes$，域的数量m。

输出：从源节点到所有域的目的节点的最优跨域组播树。

1. 初始化域间和域内Actor网络参数θ_{int}，θ_{intra}；
2. 初始化域间和域内Critic网络参数ω_{int}，ω_{intra}；
3. 初始化域间和域内数据缓存区B_{int}，B_{intra}；
4. 根据组播组管理模块分析源节点和目的节点所在域；
5. **FUNCTION** offline training(B)
6. **FOR** *batch* in B **DO**　　//分批次训练
7. 采样批量大小为k的数据(s_t,a_t,r_t,s_{t+1})，并计算每个状态的策略$\pi_\theta(a|s)$和值函数$V_\omega(s)$
8. 根据策略$\pi_\theta(a|s)$选择动作a，得到离线数据中的r和s_{t+1}；
9. 将s_t和s_{t+1}输入Critic网络计算$V_\omega(s_t)$和$V_\omega(s_{t+1})$，再根据r和$V_\omega(s_t)$和$V_\omega(s_{t+1})$计算出时序差分残差；
10. 更新Critic网络参数ω以最小化均方差损失$\omega \leftarrow \omega-\alpha_2\cdot\nabla\mathcal{L}(\omega)$；
11. 更新Actor网络参数θ以最大化目标函数$\theta \leftarrow \theta+\alpha_1\cdot\nabla_\theta J(\theta)$；
12. **END**
13. 返回训练好的Actor网络和Critic网络；
14. **END FUNCTION**
15. **FOR** $ep \leftarrow 1$ to $episodes$ **DO**
16. **FOR** 网络链路信息矩阵M_{NLI}^{int}、M_{NLI}^{intra} in NLI仓库**DO**
17. 根据src和D所在的域重置环境得到

域间M_{NLI}^{int}；
18. 堆叠域间M_T^{int}和M_{NLI}^{int}得到域间智能体状态s_t；
19. **WHILE** True **DO** //域间智能体循环
20. 域间智能体从s_t输出的动作集合中采样出a_t；执行动作a_t获得奖励值r_t和下一状态s_{t+1}；
21. 域间智能体将(s_t, a_t, r_t, s_{t+1})存入域间数据缓存区B_{int}；
22. 收集一段时间数据到B_{int}，进行离线训练offline training(B_{int})；
23. **IF** 域间组播树T_{int}构建完成**THEN**
24. 将信息同步至各域内智能体，确定每个域的转发边界节点BN；
25. **WHILE** True **DO** //域内智能体循环
26. 每个域内智能体根据BN、src、D构建各自域的域内M_T^{intra}；
27. 每个域内智能体堆叠自身的M_T^{intra}和M_{NLI}^{intra}得到状态s_t；
28. 域内智能体从自身s_t输出的动作集合中采样出a_t；执行动作a_t获得奖励值r_t和下一状态s_{t+1}；
29. 将(s_t, a_t, r_t, s_{t+1})存入对应的域内数据缓存区B_{intra}；
30. 每个域内智能体收集一段时间数据到B_{intra}；
31. 每个域内智能体进行离线训练offline training(B_{intra})；
32. 每个域内智能体构建对应域的域内组播树T_{intra}；
33. **IF** 域内组播树T_{intra}构建完成**THENBREAK**；
34. 域内智能体状态更新$s_t \leftarrow s_{t+1}$；
35. **FOR** $i \leftarrow 1$ to n_{update} **DO**；
36. 在线更新$\omega_{intra} \leftarrow \omega_{intra} - \alpha_2 \cdot \nabla \mathcal{L}(\omega_{intra})$；
37. 在线更新$\theta_{intra} \leftarrow \theta_{intra} + \alpha_1 \cdot \nabla J(\theta_{intra})$；
38. **END**
39. **END**
40. **END**
41. **ELSE** // 域间组播树未构建完成
42. 域间智能体状态更新$s_t \leftarrow s_{t+1}$；
43. **FOR** $i \leftarrow 1$ to n_{update} **DO**
44. 域间智能体在线更新$\omega_{int} \leftarrow \omega_{int} - \alpha_2 \cdot \nabla \mathcal{L}(\omega_{int})$；
45. 域间智能体在线更新$\theta_{int} \leftarrow \theta_{int} - \alpha_1 \cdot \nabla J(\theta_{int})$；
46. **END**
47. **IF**T_{int}和所有T_{intra}均构建完成**THEN**
48. 组合T_{int}和T_{intra}构建跨域组播树；
49. **BREAK**；
50. **END**
51. **END**
52. **END**
53. **END**

6 实验设置与性能评估

为了验证本文所设计方法的性能，本文进行了系列实验进行验证。本节先介绍了实验环境，包括使用到的软硬件平台，然后给出了实验测试性能指标，再通过对不同超参数设置的对比实验确定超参数具体设置值的合理性，最后将本文设计的方法与经典的组播树优化方法KMB和SCTF，以及用强化学习构建组播树的DRL-M4MR和MADRL-MR等算法进行了对比实验并进行分析，从而验证本文设计方法的性能。

6.1 实验环境

实验使用的服务器软件系统为ubuntu 18.04.6，硬件配置为64核处理器和GeForce RTX 3090显卡。在服务器上安装Mininet-Wi-Fi 2.3.1b[42]作为SDWN的仿真平台①。控制器使用Ryu 4.3.4[43]。通过Iperf[44]工具发流模拟现实网络流量。

实验使用的多域网络拓扑如图14所示。全局网络的链路参数符合均匀分布，在一定范围随机生成。其中链路带宽和时延的随机范围分别是5 Mbps～40 Mbps和1 ms～10 ms，无线AP之间的距离设置在30 m～120 m范围内。模拟流量的发流分布图15所示。

① 所做工作已经将源代码提交到开源平台https://github.com/GuetYe/MA-CDMR

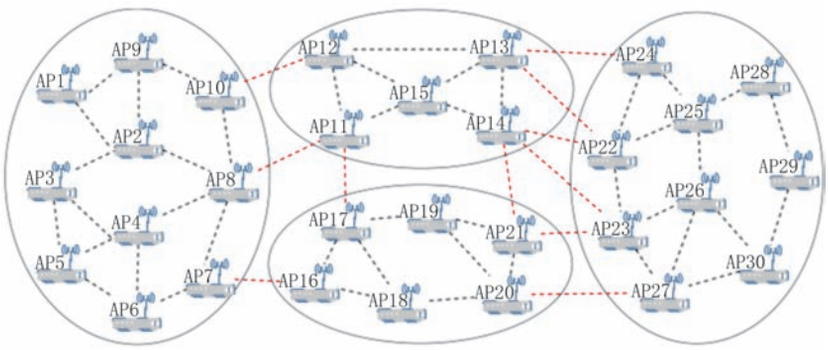


图14　多域网络拓扑

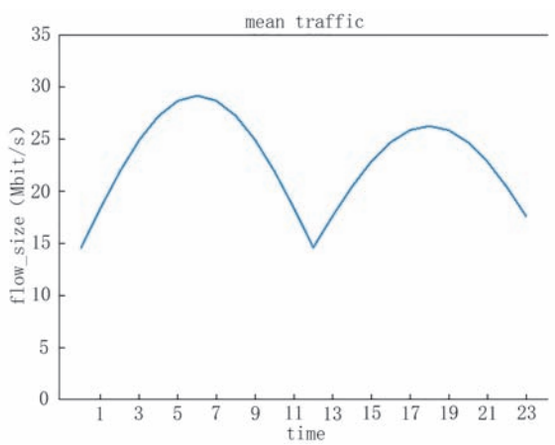


图15　模拟流量分布图

6.2　性能指标

我们使用智能体奖励值的收敛情况，以及组播树的瓶颈带宽、时延、丢包率、长度和组播树中无线AP间的平均距离作为本算法的评估指标。

(1)奖励值的描述在5.1节的奖励函数设计中，具体可以参照其中的公式。

(2)对于组播树的剩余带宽、时延、丢包率、长度以及组播树中无线AP间的平均距离，我们采用多次测量取平均值作为评价指标。如公式(22)所示。

$$\begin{aligned}
\overline{bw}_{tree} &= average\frac{\sum_n\sum_{p_k\in tree} bw_k}{n\cdot K}\\
\overline{delay}_{tree} &= average\frac{\sum_n\sum_{p_k\in tree} delay_k}{n\cdot K}\\
\overline{loss}_{tree} &= average\frac{\sum_n\sum_{p_k\in tree} loss_k}{n\cdot K}\\
\overline{len}_{tree} &= average\frac{\sum_n len_{tree}}{n}\\
\overline{dist}_{tree} &= average\frac{\sum_n\sum_{p_k\in tree} dist_k}{n\cdot K}
\end{aligned}\tag{22}$$

其中，$\overline{bw}_{tree}$、$\overline{delay}_{tree}$和$loss_{tree}$分别代表多次测量后组播树的平均瓶颈带宽、时延和丢包率。$\overline{len}_{tree}$和$\overline{dist}_{tree}$是多次测量后组播树的平均长度和无线AP之间的平均距离。bw_k、$delay_k$、$loss_k$和$dist_k$是组播树中源节点到不同目的节点路径p_k的瓶颈带宽、时延、丢包率和距离。n是在某一时刻测量的次数。K是p_k的数量。

6.3　实验参数设置

本文我们使用的是Actor-Critic网络作为智能体的核心框架，每个智能体原本都是采用在线学习的on-policy进行训练。为了验证我们设计的离线-在线学习相结合的训练方式的效果，我们将采用在线学习的on-policy和离线-在线学习结合训练方式的智能体进行对比实验，其结果如图16所示。

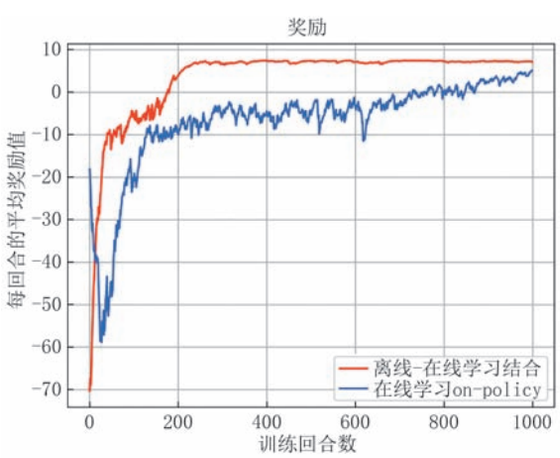


图16　智能体训练策略设置

从图16结果可以观察出，使用在线强化学习on-policy训练方式趋向于收敛，但收敛速度较慢，这是因为在这种训练方式下智能体需要与环境进行实时交互，需要不断地更新智能体的策略，导致需要较长的训练时间才可以收敛。而采用我们设计的离线-在线学习相结合的学习方式收敛速度更快，训练较短的时间即可达到收敛值，这是因为减少了智能

体与环境的交互，先使用提前收集的数据集进行离线训练，再与环境交互调整当下智能体的策略，减少智能体与环境交互的训练成本，从而加速智能体的收敛速度。

在深度强化学习模型训练过程中，超参数的设置对实验结果有着显著的影响。在实验中，我们选取一组在多域环境中域间和域内可选路径都较为复杂的跨域组播节点为代表，$M=\{\text{src}\}\cup D=\{3\}\cup\{6,15,18,19,26,28\}$。首先，设置正反馈奖励值$R_{part}$、$R_{end}$和负反馈惩罚值$R_{hell}$、$R_{loop}$。这里采用了之前的相关工作[12,45]中开展的关于大量对R_{part}、R_{end}参数权重因子的实验结果，将剩余带宽、时延、丢包率、错包率和AP之间的距离等参数的权重设置为$[0.7,0.3,0.1,0.1,0.1]$，以及此次通过大量的对负反馈惩罚值R_{hell}、R_{loop}的测试实验，发现$R_{hell}=-0.7$和$R_{loop}=-0.5$时效果最好。本文在此基础上，修改R_{part}和R_{end}的比例，使得智能体能够更加快速地达到目的节点，以此进行大量的实验，其结果如图17所示。

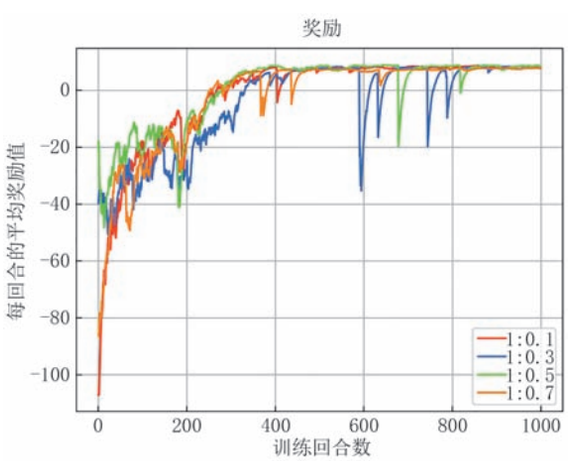


图17 四种R_{part}和R_{end}比例的奖励对比

我们将比例设置为1:0.1、1:0.3、1:0.5和1:0.7四种进行实验，也尝试过将比例设置为1:1，但最终智能体无法收敛。对于单步的奖励不能设置得过大，过大会诱导智能体走更多的步数，从而无法到达目的节点。图17结果表示1:0.1效果最佳。

其次，Actor的学习率α_1和Critic的学习率α_2都对训练过程和性能产生重要影响。α_1决定了策略更新的速度和幅度。α_2则决定了值函数的更新速度和幅度。较高的学习率可加快策略的更新速度和收敛速度，但可能导致不稳定性和过拟合。较低的学习率提高稳定性，但学习速度慢，可能需要更多的训练迭代才能达到收敛。为此本文对α_1和α_2的设置进行实验，其结果如图18所示。

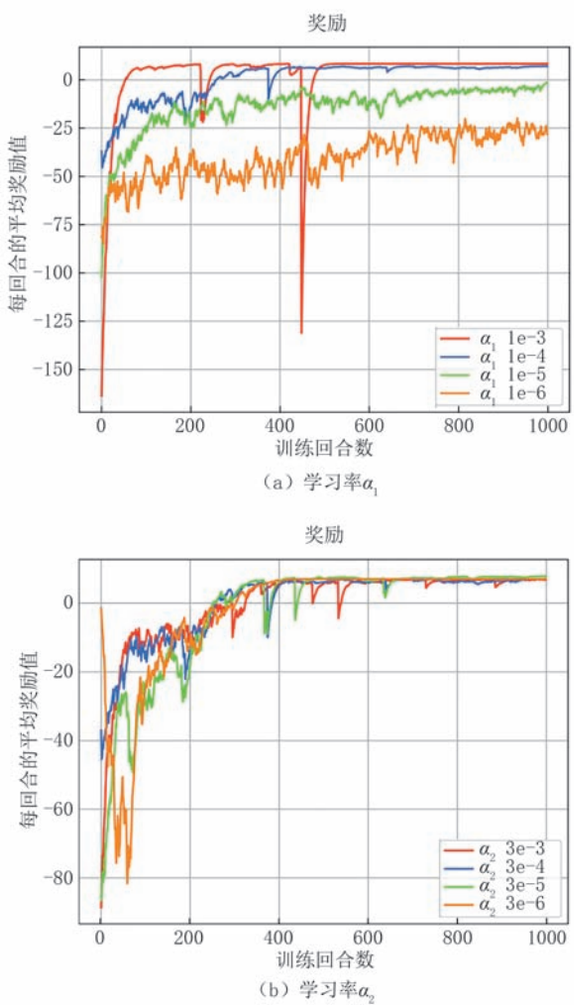


图18 学习率设置

在对α_1进行实验时，我们先将α_2设置为3e-4，由图18(a)结果显示，将α_1设置为1e-4收敛效果最好。然后再在α_1为1e-4的基础上，调整α_2的值，由图18(b)所示，当将α_2设置为3e-4时收敛效果最好。

接着，设置奖励值的衰减因子γ。衰减因子决定了当前时刻奖励与未来奖励的相对重要性。较高的衰减因子会更加重视未来奖励，使得智能体更加长远地考虑决策，但可能导致收敛速度较慢。较低的衰减因子会更加重视即时奖励，训练过程可能更快，但可能导致长期规划能力不足。为此本文对γ的设置进行实验，其结果如图19所示。图中结果表

示当γ设置为0.9时，其收敛速度较慢，但是奖励值收敛最为稳定。

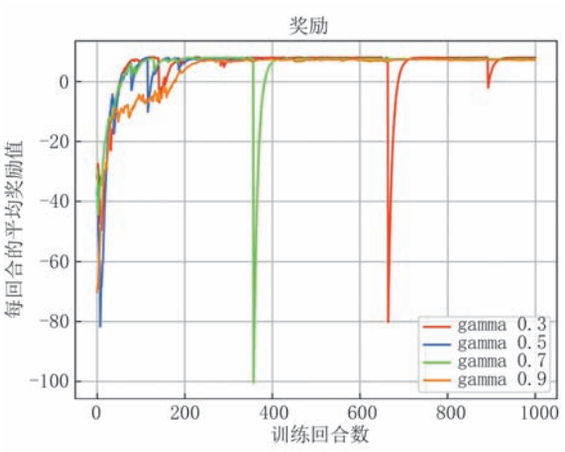


图19　衰减因子设置

然后，设置在线学习的更新频率n_{update}。较高的n_{update}可以加快收敛速度和提高训练的稳定性，减少训练所需的迭代次数，并减少抖动。然而，过高的n_{update}可能增加计算开销和导致过拟合风险。为此本文对n_{update}的设置进行实验，其结果如图20所示。

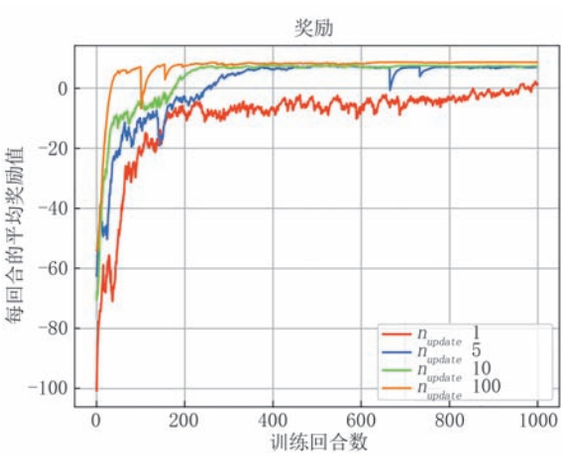


图20　更新频率设置

当设置$n_{update}=1000$时智能体无法收敛，所以我们将其设置1、5、10和100。可以观察到当$n_{update}=100$时效果最佳，但$n_{update}=10$时效果只比设置为100时收敛速度较慢，为了防止过拟合，最终我们将n_{update}设置为10。

最后，设置离线学习中批量大小k。离线强化学习中的数据集通常被分成批次来进行训练，其每批次的批量大小k是一个重要超参数。k较大可以加快训练速度，减少参数更新的方差，但会增加内存消耗。然而，k较大也可能导致模型过拟合训练数据，模型的泛化能力变差。为此本文对k的设置进行实验，其结果如图21所示。

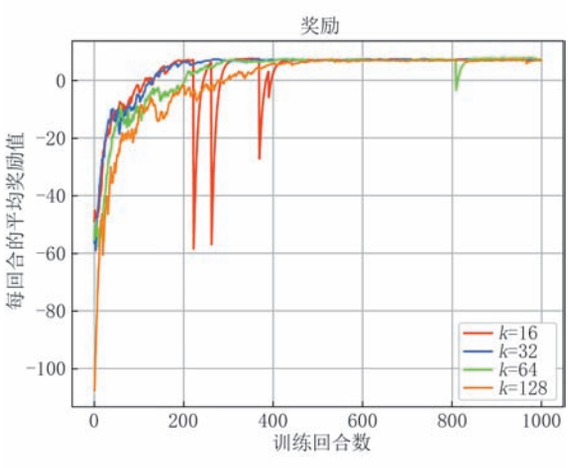


图21　批量大小设置

图21中结果显示k设置为16、32、64和128收敛的奖励值接近，但$k=32$时收敛速度更快，收敛效果最好。因此我们将k设置为32。

6.4　对比实验及结果

为了评估本文算法的性能，将MA-CDMR与4种SDN组播方法进行比较，其中包括经典的组播树优化方法KMB和SCTF，以及最近提出的基于强化学习的组播路由方法DRL-M4MR和MADRL-MR。需要解释的是，由于传统组播边界网关协议MBGP是由于MBGP是BGP的扩展，任何使用内部（IBGP）或外部BGP（EBGP）的网络都可以使用MBGP，用与BGP中指定路由策略类似的方法来提供多重策略的控制手段。一般每个自治域系统AS都必须配置MBGP，因此这样的配置和管理方式非常繁琐；而且和MBGP不同，本文采用SDN架构来获取全局视图，两种方法没有可比性，因此本文没有和MBGP进行对比。下面是对每个对比算法的扩充描述：

KMB：该方法由Kou、Markousky和Berman提出[6]，它被用于求解边的距离最小化的Steiner树问题。

SCTF：Selective Closest Terminal First，它可用于有向图中直接计算，在文献[17]中，SCTF的思想被用于求解组播树。

DRL-M4MR，该方法来自文献[11]，它通过deep Q network（DQN）强化学习算法来计算组播路由路径。

MADRL-MR，该方法来自文献[12]，是我们为解决单域环境中组播路由而设计的算法，将组播任务拆分成多个子任务由多个智能体在同一环境中协作完成组播树的构建。

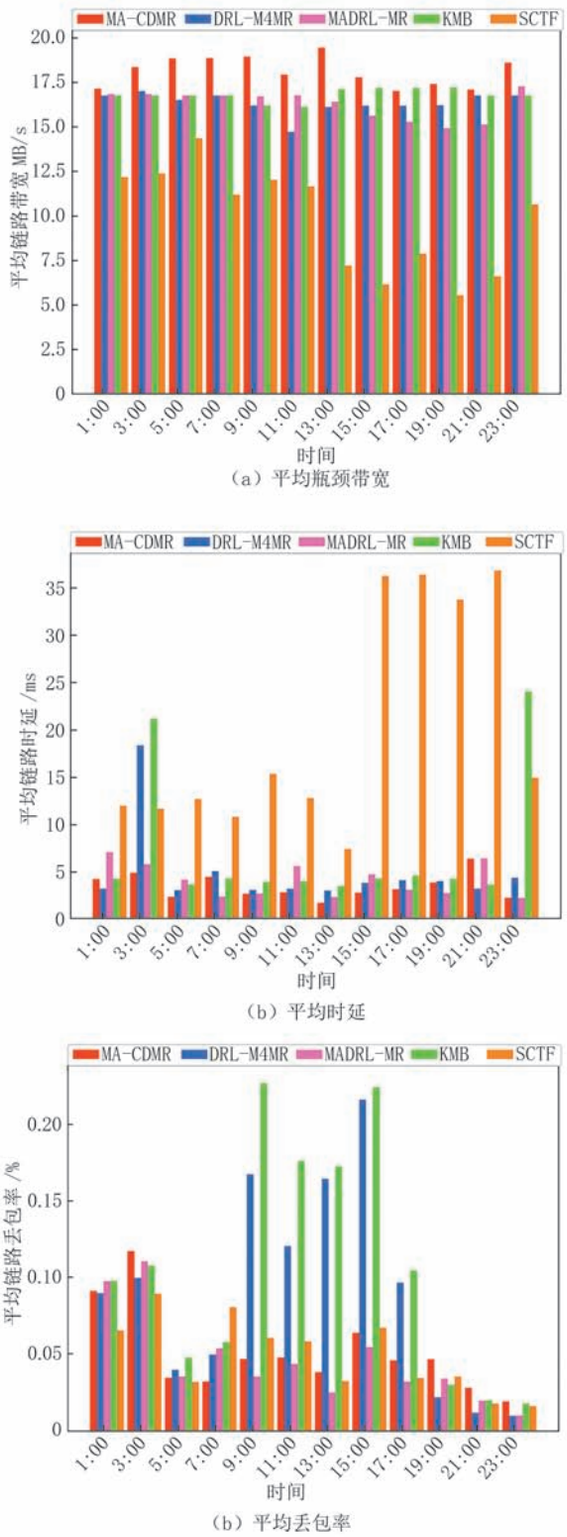
我们在多域环境下分别实现了以上提到的各种SDN组播算法，具体为：给定源节点和目的节点，依据各种算法先生成每个时刻跨域组播树。分别计算相应的各条路径瓶颈带宽的平均值，每条链路的平均时延和平均丢包率，以及链路的条数和链路的平均距离。这些性能指标的计算方式参见6.2节。然后再按每个时刻对它们进行平均，计算得到平均瓶颈带宽，平均时延，平均丢包率以及平均跨域组播树长度和平均链路距离等性能指标。

图22(a)是比较MA-CDMR与DRL-M4MR、MADRL-MR、KMB和SCTF等算法所构建跨域组播树的平均瓶颈带宽。结果表明，MA-CDMR在跨域组播树的平均瓶颈带宽上明显优于SCTF，平均性能提高了46.01%。略优于DRL-M4MR、MADRL-MR和KMB，分别提高了9.61%、10.11%和7.09%。

图22(b)是比较MA-CDMR与DRL-M4MR、MADRL-MR、KMB和SCTF等算法所构建跨域组播树的平均时延。结果表明，MA-CDMR的平均时延都小于KMB和SCTF，平均性能分别提高26.39%和78.74%。相比于DRL-M4MR性能提高了12.47%，与MADRL-MR的值非常接近，仅提高了7.17%。因此表明MA-CDMR在平均时延上有较好的效果。

图22(c)是比较MA-CDMR与DRL-M4MR、MADRL-MR、KMB和SCTF等算法所构建跨域组播树的平均丢包率。结果表明，所有算法的平均丢包率都很小。MA-CDMR的平均丢包率在整体性能上优于DRL-M4MR和KMB，平均性能分别提高了1.76%和26.94%，与MADRL-MR和SCTF的值十分接近。

图23（a）是比较MA-CDMR与DRL-M4MR、MADRL-MR、KMB和SCTF等算法所构建跨域组播树的平均长度。图中结果表明，MA-CDMR所构建的跨域组播树的长度比DRL-M4MR和MADRL-MR更短，这体现了



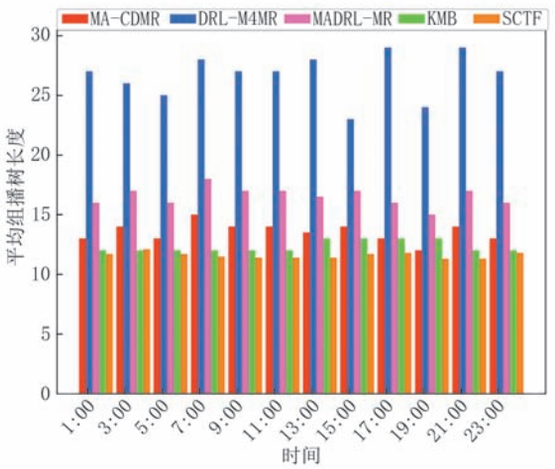
（a）平均瓶颈带宽

（b）平均时延

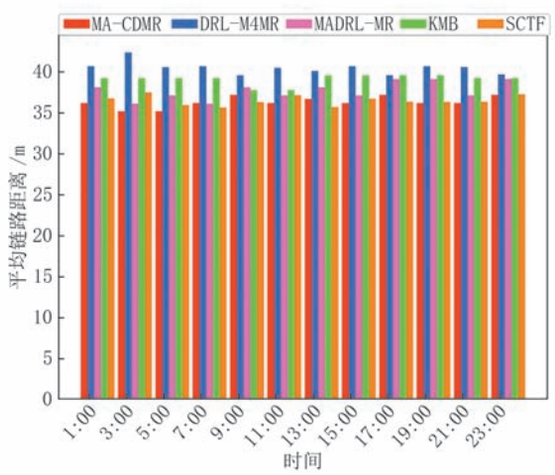
（b）平均丢包率

图22 跨域组播树的带宽、时延和丢包率对比

MA-CDMR在多域环境中策略的优势，但比KMB和SCTF具有更长的长度，这表明算法在组播树的构建中考虑的参数更多，在选择节点



（a）平均组播树长度



（b）平均链路距离

图23　平均长度和距离对比

加入组播路径时考虑的节点更多。在平衡长度和性能的情况下，MA-CDMR算法会做出折中的选择。

图23(b)是比较MA-CDMR与DRL-M4MR、MADRL-MR、KMB和SCTF等算法所构建跨域组播树中无线AP节点间的平均距离。结果表明，MA-CDMR构建的组播树AP节点之间的平均距离方面取得了较好的效果，整体优于其他四种算法。尽管MA-CDMR的组播树平均长度相对于KMB和SCTF较长，如图23(a)所示，但AP节点之间的距离并没有表现出相同的趋势。说明该算法考虑了无线AP节点间的距离，取得了较好的效果，并且跨域的多智能体策略要优于MADRL-MR算法的单域多智能体子任务策略。

7　结　论

在本文中，我们提出了MA-CDMR算法，一种在SDWN多控制器域中基于多智能体深度强化学习的智能跨域组播路由方法。首先，我们对多控制器域中的跨域组播路由进行了问题分析和建模，将跨域组播树分解为域间组播树和多棵域内组播树，分两个阶段对其进行构建，以此求其近似最优解。其次，设计了两种智能体，分别为域间智能体和域内智能体，域间智能体负责构建域间组播树，每个域内智能体分别负责构建对应域的域内组播树。通过SDWN多控制器域网络架构采集的流量数据，设计智能体的状态空间；并根据这两种智能体任务的特点，分别设计了相对应动作空间，设计域间智能体的动作空间为所有域之间边的集合，每次动作为选择一条相邻域的域间路径；设计域内智能体的动作空间为相对应域的域内节点结合，每次动作为选择一个节点加入域内组播树。最后根据不同情况设计相应的奖励函数。

根据大量的对比实验，验证了MA-CDMR算法在多控制器域中比经典的组播树优化方法KMB和SCTF，以及用强化学习构建组播树的DRL-M4MR和MADRL-MR等算法具有更优的性能。

此外，本文为主要研究工作均在控制平面路由构建问题，在未来研究中，应该进一步探索控制器部署的最佳实践。这包括确定控制器的位置、数量和布局，以最大程度地提高组播路由的性能。也可以探索数据平面的P4编程方面，以进一步提高组播路由的性能和灵活性。通过优化P4编程模型，设计高效的组播路由功能，并将智能算法与P4编程相结合，可以实现更智能、自适应和高质量的组播路由。在更大规模的网络环境中，可能同时存在多个源节点的组播树传输数据，在各个域内不仅需要考虑同源的组播树，还可能存在不同源的组播树，需要优化不同源组播树之间的协作与竞争实现组播路由的优化，可以在本方法的基础上添加域内组播森林的优化方法实现组播路由的优化。

致谢　在此，我们向对本文的工作给予支持和建议的同行，尤其是认知无线电与信息处理教育部重点实验室网络和广西教育大数据应用与

网络安全协同创新中心智能优化讨论班上的老师和同学,以及编辑和审稿人表示感谢。

参 考 文 献

[1] Farhan K A, Abdel-Fattah F, Altarawneh F, et al. Survey paper on multicast routing in mobile ad-hoc networks//Proceedings of theIEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). Amman, Jordan, 2019: 449-452

[2] Luo S, Xing H, Li K.Near-optimal multicast tree construction in leaf-spine data center networks. IEEE Systems Journal, 2020, 14(2):2581-2584

[3] Costanzo S, Galluccio L, Morabito G, et al. Software defined wireless network (SDWN): An evolvable architecture for W-PANs//Proceedings of the IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI). Turin, Italy, 2015: 23-28

[4] Rekhter, Y, LiT, Hares S. RFC 4271: A border gateway protocol 4 (BGP-4), ed: RFC Editor, 2006

[5] 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

[6] Kou L, Markowsky G, Berman L.A fast algorithm for Steiner trees.Acta informatica, 1981, 15: 141-145

[7] Ramanathan S. Multicast tree generation in networks with asymmetric links. IEEE/ACM transactions on Networking, 1996, 4(4): 558-568

[8] Ke W, Wang Y, Ye M, et al. A priority-based multicast flow scheduling method for a collaborative edge storage datacenter network.IEEE Access, 2021, 9: 79793-79805

[9] Ashour O, St-Hilaire M, Kunz T, et al. A survey of applying reinforcement learning techniques to multicast routing//Proceedings of theIEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). New York, USA, 2019: 1145-1151

[10] 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

[11] Zhao C, Ye M, Xue X, et al. DRL-M4MR: an intelligent multicast routing approach based on DQN deep reinforcement learning in SDN.Physical Communication, 2022, 55: 1-16

[12] Hu H, Ye M, Zhao C, et al. Intelligent multicast routing method based on multi-agent deep reinforcement learning in SDWN. Mathematical Biosciences and Engineering, 2023, 20(9): 17158-17196

[13] Padakandla S.A survey of reinforcement learning algorithms for dynamically varying environments. ACM Computing Surveys (CSUR), 2021, 54(6): 1-25

[14] Zhang Z, Ong Y S, Wang D, et al. A collaborative multiagent reinforcement learning method based on policy gradient potential. IEEE transactions on cybernetics, 2019, 51(2): 1015-1027

[15] Chu K F, Lam A Y S, Li V O K. Traffic signal control using end-to-end off-policy deep reinforcement learning. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(7): 7184-7195

[16] Prudencio R F, Maximo M R O A, Colombini E L.A survey on offline reinforcement learning: Taxonomy, review, and open problems. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(8): 10237-10257

[17] Angelopoulos S. Parameterized analysis of the online priority and node-weighted steiner tree problems. Theory of Computing Systems, 2019, 63: 1413-1447

[18] Takahashi H. An approximate solution for steiner problem in graphs.Math. Japonica, 1980, 24(6): 573-577

[19] Rayward-Smith V J.The computation of nearly minimal Steiner trees in graphs. International Journal of Mathematical Education in Science and Technology, 1983, 14(1): 15-23

[20] Naser J I, Kadhim A J. Multicast routing strategy for SDN-cluster based MANET. International Journal of Electrical & Computer Engineering (IJECE), 2020, 10(5): 4447-4457

[21] Ji W, Yang S, Zhang B, et al.Multi-domain multicast routing mutation scheme for resisting DDoS attacks//Proceedings of theInternational Wireless Communications and Mobile Computing (IWCMC). Dubrovnik, Croatia, 2022: 142-147

[22] Chiang S H, Wang C H, Yang D N, et al. Distributed multicast traffic engineering for multi-domain software-defined networks. IEEE Transactions on Parallel and Distributed Systems, 2022, 34(2): 446-462

[23] Liu R, Li S, Wang H, et al.A QoS routing optimization algorithm based on hierarchical multi-controller coordination//Proceedings of the IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). Chengdu, China, 2019,1: 1820-1825

[24] Wu Q, Liu J, Zheng M, et al.Multicast recovery algorithm for multilayer multi-domain optical networks based on hybrid swarm intelligence//Proceedings of the7th International Conference on Information Science, Computer Technology and Transportation (ISCTT). Xishuangbanna, China, 2022: 1-4

[25] Liu Q, Lin X, Yue S, et al. A routing scheme for bulk data transfers in multi-domain OCS networks with assistive storage//Proceedings of theIEEE International Conference on Communications (ICC).Shanghai, China, 2019: 1-6

[26] Zhao J, Li F, Ren D, et al. An intelligent inter-domain routing scheme under the consideration of diffserv QoS and energy saving in multi-domain software-defined flexible optical networks.Optics Communications,2016, 366: 229-240

[27] Li D, Fang H, Zhang X, et al. DeepMDR: a deep-learning-assisted control plane system for scalable, protocol-independent, and multi-domain network automation. IEEE Communications Magazine,2021, 59(3): 62-68

[28] Xu L, Huang Y C, Xue Y, et al. Hierarchical reinforcement learning in multi-domain elastic optical networks to realize joint RMSA. Journal of Lightwave Technology,2023, 41(8): 2276-2288

[29] Dang D, Nguyen K K, Mba V A, et al. Efficient link-state

multicast routing by optimizing link-weight with MARL// Proceedings of the IEEE Military Communications Conference (MILCOM). Boston, USA, 2023: 787-792

[30] Zhao D, Lu Y, Li X, et al.Cross-domain service function chain routing: multiagent reinforcement learning approaches. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, 69(12): 4754-4758

[31] Bhavanasi S S, Pappone L, Esposito F. Dealing with changes: resilient routing via graph neural networks and multi-agent deep reinforcement learning. IEEE Transactions on Network and Service Management, 2023, 20(3): 2283-2294

[32] 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

[33] Gilbert E N, Pollak H O. Steiner minimal trees SIAM journal on applied mathematics, 1968, 16(1): 1-29

[34] Henderi H, Wahyuningsih T, Rahwanto E.Comparison of Min-Max normalization and Z-score normalization in the K-nearest neighbor (kNN) algorithm to test the accuracy of types of breast cancer. International Journal of Informatics and Information Systems, 2021, 4(1): 13-20

[35] Abdollahi M, Ni W, Abolhasan M, et al. Software-defined networking-based adaptive routing for multi-hop multi-frequency wireless mesh. IEEE Transactions on Vehicular Technology, 2021, 70(12): 13073-13086

[36] Carthern C, Wilson W, Rivera N. Data link layer//Cisco Networks: Engineers' Handbook of Routing, Switching, and Security with IOS, NX-OS, and ASA. Berkeley, USA: Apress, 2021: 39-57

[37] McKeown N. Software-defined networking. INFOCOM keynote talk, 2009, 17(2): 30-32

[38] Qiu S, Yang Z, Ye J, et al.On finite-time convergence of actor-critic algorithm.IEEE Journal on Selected Areas in Information Theory, 2021, 2(2): 652-664

[39] Park B, Kim T, Moon W, et al. Off-policy reinforcement learning with loss function weighted by temporal difference error//Proceedings of theIntelligent Computing Technology and Applications. Zhengzhou, China,2023: 600-613

[40] Feriani A, Hossain E. Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: a tutorial. IEEE Communications Surveys & Tutorials, 2021, 23(2): 1226-1252

[41] Rahimi Gorji S, Granmo O C. Off-policy and on-policy reinforcement learning with the Tsetlin machine. Applied Intelligence, 2023, 53(8): 8596-8613

[42] "Mininet-WIFI." https://mininet-wifi.github.io/

[43] "Ryu." https://ryu-sdn.org/(accessed Feb.16, 2024)

[44] "Iperf." https://iperf.fr(accessed Feb.16, 2024)

[45] 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

附录：相关专业术语表

中文	英文
最优组播树	Optimized multicast trees
软件定义无线网络	Software defined wireless network
多域SDWN	multi-domain software defined wireless networks
跨域组播树	Cross-domain multicast tree
在线强化学习	Online reinforcement learning
离线强化学习	Offline reinforcement learning
多智能体深度强化学习	Multi-agent reinforcement learning
最优跨域组播树	Optimal cross-domain multicast tree
域间组播树	Inter-domain multicast tree
域内组播树	Intra-domain multicast tree
最小斯坦纳树	Steiner minimal trees
斯坦纳节点	Steiner points
域间路径	Inter-domain path

YE Miao, Ph. D., professor, Ph. D. supervisor. His 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).

HU Hong-Wen, Ph. D. candidate. His main research interests include software defined networking, reinforcement learning.

WANG Yong, Ph. D., professor, Ph. D. supervisor. His

main research interests include cloud computing and software defined networking.

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

WANG Xiao-Li, Ph. D., associate professor. Her main research interests include artificial intelligent, distributed computing.

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

ZHENG Ji-Hao, M. S. candidate, His main research interests include software defined networking, reinforcement learning.

Background

This paper investigates the cross-domain multicast routing issue in multi-domain Software-Defined Wireless Networks (SDWN), a subject matter within the Information and Communication Technology (ICT) domain. The existing solutions to this issue include approximate optimal methods based on pruning and greedy search strategies, as well as heuristic swarm intelligence optimization algorithms. These solutions are often inflexible and ill-suited to the demands of high-speed, dynamically changing network traffic, focusing on a limited range of optimization metrics and lacking comprehensive global optimization capabilities. Additionally, routing methods that employ reinforcement learning mechanisms, such as Q-RL, DRL-M4MR, and MADRL-MR, are hampered by their insufficient awareness of network traffic states, which prevents them from adequately meeting Quality of Service (QoS) requirements. These methods also generally exhibit slow convergence, posing challenges for their implementation in highly dynamic network environments.

This research has developed and implemented a multiagent deep reinforcement learning-based cross-domain multicast routing (MA-CDMR) in SDWN. The methodology includes a multicast group management module and a multi-controller communication mechanism, which together facilitate the efficient transmission and synchronization of network state information across various domains, thereby effectively managing the membership dynamics of cross-domain multicast groups. Each controller is endowed with an agent, and a collaborative mechanism among these agents is established to ensure the consistency and effectiveness of network state information representation for cross-domain multicast routing decisions. Moreover, the methodology adopts a combined online and offline multi-agent reinforcement learning training paradigm, which reduces the dependency on real-time environments to enhance the convergence rate of multi-agents.

This work is partly supported by the National Natural Science Foundation of China (Nos. 62161006, 62372353, U22A2098), Key Laboratory of Cognitive Radio and Information Processing of Ministry of Education (No. CRKL220103), Innovation Project of Guangxi Graduate Education (No. YCBZ2023134). 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.

Prior to the commencement of this project, our research group had already established a noteworthy record of achievements in the field of ICT. Our research has spanned a wide array of tasks, transitioning from unicast to multicast, and has engaged with diverse methodologies, ranging from swarm intelligence optimization to reinforcement learning algorithms. Additionally, our contributions have extended across both wired and wireless networks, demonstrating a comprehensive and interdisciplinary approach to advancing ICT.