

面向工业无线确定性传输的多路径路由与调度联合优化

陈荣均,王洪超*,王钦定,乔凯,田伟康,杨冬

(北京交通大学电子信息工程学院,北京 100044)

摘要: 随着工业无线网络和无线通信技术的快速发展,无线网络的确定性传输已成为一个重要的研究方向。然而,无线信道中的不确定因素,如多径衰落和同频干扰,给无线网络的确定性传输带来了诸多挑战。为了解决这些问题,Internet工程任务组(Internet Engineering Task Force, IETF)提出了可靠可用无线(Reliable and Available Wireless, RAW)架构,并在工业无线网络场景中使用时隙跳频(Time-Slotted Channel Hopping, TSCH)作为底层实现技术。为了确保可靠性和严格的时延要求,RAW设计了多种保障机制,包括通过数据包复制、消除与排序功能(Packet Replication, Elimination and Ordering Functions, PREOF)技术利用路径冗余提升传输的可靠性和确定性。然而,现有的调度方案未充分考虑PREOF以及路由和调度的联合优化,导致时频资源分配时存在冗余和资源浪费,从而影响了网络对关键流的调度能力。本文面向确定性流量传输的多路径路由与调度联合优化问题进行建模,并提出了一种基于分层强化学习的资源分配算法(Hierarchical Reinforcement Resource Allocation, HRRA)。其中,高层策略负责多路径路由的选择,低层策略则基于高层策略的路由决策进行时频资源的分配,同时考虑PREOF在聚合节点对冗余包的删除。针对拓扑规模的变化和流量的异构性,在高层策略引入图神经网络(Graph Neural Network, GNN)增强对输入特征表征能力。HRRA算法能够根据流的截止时间、可靠性等需求选择合适的动作,从而最大化调度流数量和资源利用效率。通过这种跨层优化架构和对PREOF的支持,HRRA不仅有效解决了资源冗余和调度能力不足的问题,还增强了对流的确定性通信需求的支持。实验表明,相比于DGRL+MWIS和EDF-MO等基准算法,HRRA分别提升了10.6%和36.6%的调度能力,同时实现了更高的资源利用效率。

关键词: 可靠可用无线网络;数据包复制-消除-有序转发;分层强化学习;图神经网络;网络资源调度

基金项目: 国家自然科学基金(No.62425104)

中图分类号: TN913.21

文献标识码: A

文章编号: 0372-2112(2026)01-0068-18

电子学报URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20250734

Joint Optimization of Multipath Routing and Scheduling for Industrial Wireless Deterministic Transmission

CHEN Rongjun, WANG Hongchao*, WANG Qinding, QIAO Kai, TIAN Weikang, YANG Dong

(School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: With the rapid development of industrial wireless networks and wireless communication technologies, deterministic transmission in wireless networks has emerged as an important research direction. However, the inherent uncertainties of wireless channels, such as multipath fading and co-channel interference, pose significant challenges to achieving deterministic transmission. To address these challenges, the internet engineering task force (IETF) proposed the reliable and available wireless (RAW) architecture, which adopts time-slotted channel hopping (TSCH) as the underlying technology in industrial wireless network scenarios. In order to ensure reliability and stringent delay requirements, RAW incorporates a variety of mechanisms, including the use of packet replication, elimination and ordering functions (PREOF) to exploit path redundancy and thereby enhance transmission reliability and determinism. Nevertheless, existing scheduling schemes have not sufficiently considered PREOF or the joint optimization of routing and scheduling. This results in redundancy and inefficient resource allocation in the time-frequency domain, limiting the network's ability to support critical flows. In this work, we formulate the joint optimization problem of multipath routing and scheduling for deterministic flow transmission and propose a hierarchical reinforcement learning-based resource allocation algorithm, termed hierarchical reinforcement resource allocation (HRRA). In HRRA, the high-level policy is responsible for selecting multipath routes, while the low-level policy allocates time-frequency resources based on the high-level routing decisions, explicitly accounting for the elimina-

tion of redundant packets by PREOF at aggregation nodes. To address variations in topology size and heterogeneous traffic demands, a graph neural network (GNN) is integrated into the high-level policy to enhance feature representation. The HRRRA algorithm selects appropriate actions according to flow requirements such as deadlines and reliability, thereby maximizing both the number of schedulable flows and overall resource utilization. Through this cross-layer optimization framework and explicit support for PREOF, HRRRA not only mitigates redundancy and improves scheduling efficiency but also better supports deterministic communication requirements. Experimental results demonstrate that, compared to baseline schemes such as DGRL+MWIS and EDF-MO, HRRRA improves scheduling capability by 10.6% and 36.6%, respectively, while achieving higher resource utilization.

Keywords: reliable and available wireless; packet replication, elimination and ordering functions; hierarchical reinforcement learning; graph neural network; network resource scheduling

Foundation Item(s): National Natural Science Foundation of China (No.62425104)

0 引言

随着工业物联网 (Industrial Internet of Things, IIoT) 技术的快速发展,越来越多的应用场景依赖网络通信实现实时数据传输和设备管理,如自动控制系统和监控系统^[1-2]。这些应用对时延和可靠性的要求极为严格,通常需要确保数据包能够在严格的截止时间之前到达,并且需要保证传输的高可靠性。如在预测性维护场景中,系统需通过传感器数据实时分析设备状态,通常要求端到端时延控制在 100~300 ms 之间,且传输可靠性达到 99% 以上^[3]。现有有线网络所采用的确定性保障技术有时间敏感网络 (Time-Sensitive Networking, TSN) 和确定性网络 (Deterministic Networking, DetNet) 等。然而,与有线确定性网络技术不同,无线通信信道给 IIoT 的确定性传输需求带来了诸多挑战。无线环境中存在的不确定因素,如同频干扰和多径衰落等,极易导致数据包丢失或传输错误,从而影响确定性通信的稳定性和可靠性^[4]。

为应对上述挑战,Internet 工程任务组 (Internet Engineering Task Force, IETF) 提出了可靠可用无线网络 (Reliable and Available Wireless, RAW) 标准化框架,将 DetNet 的能力扩展至无线场景^[5]。RAW 是基于 OSI 模型第三层 (Layer 3) 的方案,旨在通过各类可调度的无线链路实现确定性通信。为应对无线信道中的不确定因素,RAW 设计了多种保障机制,包括数据包复制-消除-有序转发 (Packet Replication, Elimination and Ordering Functions, PREOF)、旁听转发 (Overhearing) 和冗余预留 (Overprovisioning) 等。其中,PREOF 是 RAW 架构的核心机制之一^[6],它通过在多条路径上并行传输数据包,利用路径冗余换取可靠性的提升。具体做法是在分离节点对数据包进行复制,并分别沿多条路径发送。聚合节点接收各路径转发过来的副本后,保留正确到达的包并删除冗余副本,再将有效包继续转发。研究表明,PREOF 不仅能够有效提高传输可靠性,而且缩短了端到端时延,因此尤为

适用于对确定性通信有严格要求的关键流量^[7]。

在工业场景中,RAW 采用 IEEE 802.15.4-TSCH (Time-Slotted Channel Hopping) 作为底层实现技术,以提供高效且具有确定性传输能力的无线通信解决方案^[8]。TSCH 通过将时间划分为多个固定的时隙,并在每个时隙内采用不同的信道,将数据分配到时隙与信道组成的 cell 资源块中进行传输。这种时隙调度再与信道跳频结合,可有效避免信道干扰和碰撞,保证数据传输的高可靠性与低时延。具体来说,TSCH 的时隙机制确保了每个节点在特定的时隙内进行数据传输,避免了不同节点之间的干扰。而信道跳频技术则通过在多个信道之间切换,进一步提高了抗干扰能力,确保了传输的可靠性。此外,TSCH 还通过为关键流量分配专用时隙和资源预留的方式,提供了优先级调度,从而实现了针对不同流量的服务质量保障。这些特性使得 TSCH 在实现 RAW 的过程中,能够满足 IIoT 对确定性、可靠性和实时性的严格要求。

TSCH 的调度功能对于实现其确定性通信能力至关重要,现有许多研究在这一方向开展研究工作。6TiSCH 工作组提出了分布式调度功能最小调度函数 (Minimal Scheduling Function, MSF)^[9],这是首个具备完整功能的调度算法,在实际部署中应用广泛。MSF 依托 6top 协议在邻居节点之间进行协商,根据业务负载的动态变化自适应增删 cell 资源。然而,由于其采用随机方式分配 cell,生成的调度可能会产生冲突,因此不适用于对时延要求严格的确定性通信场景^[10]。与之相比,集中式调度算法可以基于全网拓扑和流量需求,确定性地分配 cell 资源,更适合提供稳定且能够满足严格要求的流量传输^[11-12]。TASA 是一种经典的集中式调度算法,基于匹配和图着色技术,为每对节点分配 cell 资源^[13]。然而上述调度方案都没有考虑多路径路由下的情况。ODeSe 是一种多路径路由算法,基于 IIoT 系统中常用的低功耗有损网络路由协议 (Routing Protocol for Low-power and lossy networks, RPL) 进行扩展^[14]。结合自定义的集中式

TSCH 调度算法,该研究实现了完整的路由与调度优化。实验结果表明,与默认的 RPL 单路径算法相比,该方案在可靠性、时延和能耗等方面表现更优,有效实现了可靠性与能耗之间的平衡。然而,该方案虽然基于多路径路由,却并未深入研究 PREOF 及其对调度的影响。

传统的路由与调度分层优化方法(如 RPL+MSF)由于缺乏信息交互和资源协同,无法有效保障流的确定性传输,且调度能力有限。因此,许多研究通过跨层优化来增强 TSCH 网络对确定性需求流的调度能力。EDF-MO 是一种路由与调度联合优化方法^[15],其中,MO 是一种贪婪启发式算法,旨在减少最短路径路由中的路径重叠,EDF 则是一种截止优先调度算法。通过将调度层的冲突约束提前嵌入路由层,MO 有效减少了路径上的冲突,结合 EDF 的时隙分配,形成端到端的跨层联合优化,从而显著提升了流量调度的能力。ILP3 基于 TASA 算法,利用整数线性规划(Integer Linear Programming, ILP),通过同时优化路由选择和时隙信道分配实现了跨层联合优化^[16]。通过引入负载上界(即网络中根节点的最大负载)作为优化目标,平衡全局的网络负载,以实现最小化网络负载及平衡流量,并提高整体性能,从而提供跨层联合优化的全局基准解。然而上述跨层优化方案都未考虑多路径路由,也因此未在调度中支持 PREOF 功能。

随着深度强化学习(Deep Reinforcement Learning, DRL)技术的快速发展,其在解决高维、动态、非线性优化问题中已被证明具有良好的泛化能力与策略学习优势,逐渐成为无线网络资源调度领域的重要工具^[17]。Li 等人^[18]面向飞行自组网(Flying Ad-hoc Network, FANET)路由中高动态大规模场景下的低时延与高能耗问题,提出了基于强化学习(Reinforcement Learning, RL)的能量感知快速路由决策方法。具体而言,该方法利用无人机的本地观测信息构建多尺度跨层状态向量,包括一跳邻居的电池剩余量、信道增益与信噪比反馈、重广播计数以及一跳跳数和时延统计,通过加权求和的效用函数在每跳同时学习“是否转发”和离散的中继功率动作,从而联合优化数据帧传输效率和能量消耗。该工作还引入了邻居选择优先级和经验共享机制,增强了对路径质量的长期效用估计与风险评估,在策略更新时进一步压缩状态空间并缓解高动态下的量化误差。该方案能显著降低路由振荡,加速策略收敛,并在保持高可靠传输的同时获得更低的平均端到端时延和更少的能量消耗,从而有效延长 FANET 网络生命周期并提升高移动速率场景下的网络性能。HRL-TSCH 是一项代表性的分层强化学习(Hierarchical Reinforcement Learning, HRL)

方案^[19],通过将 TSCH 调度过程拆分为高层与低层两级策略,实现对动态信道质量的自适应调度优化。高层策略在较粗时间尺度上执行链路级的资源控制,低层策略在单个链路维度上选择具体的 cell 资源块,实现细粒度的时隙与频率分配,实现了对能耗、时延和带宽的优化。图神经网络(Graph Neural Networks, GNN)凭借其在处理图结构数据、建模拓扑关系方面的优势,已被广泛应用于多跳无线网络中的路径选择与冲突感知问题^[20-21]。Huang 等人^[22]面向多跳无线网络中路由路径选择的动态优化任务,提出了一种路由控制方法 DGRL,利用图卷积网络(Graph Convolutional Networks, GCN)提取网络全局拓扑与链路状态特征,并结合 Actor-Critic 架构学习路径选择策略,旨在在链路负载不均与拓扑变化的条件下,实现负载均衡与端到端时延最小化,优化路径规划策略的稳定性与效率。Zhao 等人^[23]则聚焦于共享信道环境下的调度优化问题,为提升在多链路干扰约束下的并行传输能力,提出了一种基于图感知的 MWIS 调度器。该方法通过 GCN 对冲突图进行结构嵌入,捕捉链路之间的干扰关系,并借助贪婪搜索或图启发式方法近似求解最大加权独立集问题,在保证冲突避免的前提下,选出一组可并发调度的高优先级链路,从而提升整体链路利用率与调度效率。然而,上述方案均聚焦于路由或调度的单一任务优化,在系统建模上未考虑二者之间的耦合关系,也缺乏用于支持路由、调度联合策略学习的统一决策框架与协同优化机制。

综上所述,现有文献缺乏对路由与调度的联合优化策略,且鲜有工作同时支持多路径路由传输与 PREOF 机制。当前多数研究采用不相交路径或多路径独立传输模式,虽然简化了调度冲突处理,但往往导致调度资源重复分配。在资源高度受限的 IIoT 环境中,这类冗余调度不仅效率低下,更难以满足关键业务对高可靠性与高资源利用率的严苛要求。此外,现有方法多为分层优化架构,缺乏跨层协同机制,导致路径选择与调度决策彼此独立,难以实现全局优化,导致调度能力受限。为有效应对多路径转发中产生的资源冗余,以及分层调度导致的性能瓶颈,亟需设计一种面向确定性流量的联合调度方案,能够实现路由与调度的跨层协同优化,支持多路径传输与 PREOF 机制,从而更好地满足 RAW 场景下对低时延、高可靠性与高资源效率的调度需求。

路由与调度的联合优化属于典型的 NP-hard(NP 困难)组合优化问题,传统方法在求解过程中常面临解空间庞大、计算复杂度高以及难以扩展至大规模网络等挑战^[24-25]。相比之下,DRL 具备处理高维动作空间、适应动态环境与实现快速在线决策的能力,已在

资源调度等任务中展现出显著优势^[26]。DRL 不仅能将主要计算前移至离线训练阶段,在在线阶段实现高效推理,还能在有限计算开销下输出高质量甚至近似最优的解,有效降低实时优化开销^[27]。因此,本文引入 DRL 框架求解联合路由与调度问题,采用 HRL 方法将决策过程分为高层和低层两个阶段,高层策略负责多路径路由选择,而低层策略基于高层策略的动作进行时频和信道的分配。此外,考虑到不同拓扑规模和异构流量,在高层策略中引入了 GNN,用于对拓扑结构等非欧几里得数据进行特征提取。所提分层结构显著增强了算法的适应性、探索与利用效率,从而更好地应对不同拓扑的网络环境和复杂的调度需求。对比基准算法 DGRL+MWIS 和 EDF-MO,所提分层强化学习资源分配算法(Hierarchical Reinforcement Resource Allocation, HRRA)提高了 10.6% 和 36.6% 的调度成功率,以及 9.9% 和 35.1% 的资源利用效率。

本文的主要贡献如下。

(1)设计了一种支持 PREOF 机制的 TSCH 调度方法,特别是在聚合节点避免为冗余数据包分配时频资源造成的无效占用,为资源受限场景下的有效调度提供了支持。

(2)形成了路由与调度的联合优化问题,在 PREOF 支持下使用多路径路由,通过跨层协同优化路径选择和时频资源分配,以最大化调度成功率和资源利用效率,为解决确定性传输的优化调度问题奠定了基础。

(3)提出了面向确定性传输需求的 HRRA,可对工业场景中的关键流进行高效调度并满足其确定性需求。HRRA 引入 GNN 增强输入特征表征能力,因而能够适应不同规模拓扑和异构流量。与基准算法相比,HRRA 有效提高了调度成功率,并在 PREOF 支持下提高了资源利用效率。

本文所有符号描述如表 1 所示。

表 1 符号描述

Table 1 Notations

符号	含义
G	网络拓扑图
$v_m \in \mathcal{V}$	G 中第 m 个节点和节点集合
$e_n \in \mathcal{E}$	G 中第 n 条链路和链路集合
$\mathcal{P}_{(v_s, v_d)}$	从节点 v_s 到节点 v_d 的可达路径集合
$P_{(v_s, v_d), k}$	从节点 v_s 到节点 v_d 的第 k 条路径
$r_{(v_s, v_d)}$	从节点 v_s 到节点 v_d 的多路径路由
$f_h \in \mathcal{F}$	第 h 个流和流的集合
$v_{(h, s)}, v_{(h, d)}$	流 f_h 的源节点和目的节点
p_h	流 f_h 的周期
d_h	流 f_h 的截止时间需求
rel_h	流 f_h 的可靠性需求
plr_e	链路 e 的丢包率
sf	时隙帧长度
$O(v_m)$	发送端是 v_m ,接收端是其他节点的链路的集合
$I(v_m)$	接收端是 v_m ,发送端是其他节点的链路的集合
$S_t^i(v_m, v_n)$	时隙 t 是否存在 cell 分配给链路 (v_m, v_n) ,如果存在则为 1,反之为 0
$x_{(v_m, v_n)}(c, t)$	时隙为 t 信道为 c 的资源块 (c, t) 上是否存在从 v_m 到 v_n 的传输,存在则为 1,否则为 0
$S_{f_h}^p$	流 f_h 在路径 p 上的链路 (v_m, v_n) 调度的时隙编号
δ_{f_h}	分配的资源是否满足流 f_h 的确定性需求,满足为 1,否则为 0

1 系统模型和调度约束

1.1 系统架构

本文聚焦于工业场景下的确定性流量传输,其中,关键业务对通信性能提出了较高要求。以过程自动化场景为例,传感器周期性地采集关键工艺参数(如温度、压力、振动),并通过无线网络传输至控制器或监控平台。此类应用通常要求数十毫秒级的

端到端时延以及接近 100% 的可靠性,以确保系统在出现异常时能够及时响应。传统的分布式路由 RPL、分布式调度 MSF 等,在提供严格的通信保障时面临困难,而集中式资源调度因为具有全局信息、资源统一调度避免冲突的优势,更适合提供确定性调度和稳定的性能。本文采用如图 1 所示的集中式网络架构,包含应用层、控制层和数据层 3 层。应用层承载用户

需求和各类应用,通过北向接口将用户需求传递给控制层。控制层运行在本地或者云端,负责对网络信息的收集分析,并通过DRL智能体进行包括路径选择和TSCH调度的全局资源分配。数据层由普通传感器和Sink节点两类传感器组成,所有传感器之间基于IEEE802.15.4-TSCH进行通信,此外,Sink节点通过有线接口与控制层直连。数据层定期向控制层反馈网络状态信息,同时接收并部署控制层下发的资源调度方案。在预定的无冲突资源调度方案下,数据层能够实现确定性的端到端数据传输。

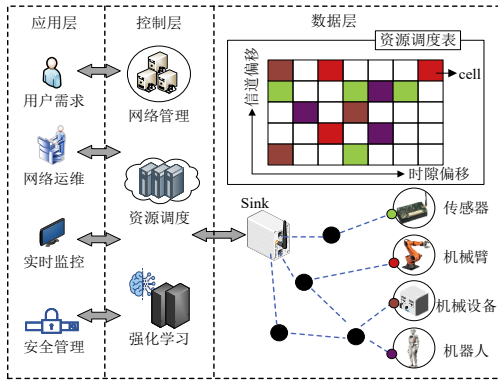


图1 系统架构

Figure 1 System architecture

1.2 网络模型

TSCH网络可以建模为一个有向图 $G(\mathcal{V}, \mathcal{E})$,其中 $\mathcal{V}=\{v_1, v_2, \dots, v_M\}$ 表示网络节点的集合, $\mathcal{E}=\{e_1, e_2, \dots, e_N\}$ 表示节点之间链路的集合。用 $e_n=(v_1, v_2)$ 表示发送节点 v_1 和接收节点 v_2 之间的链路。 $p_{(v_s, v_d),*} \in \mathcal{P}$ 表示一条源节点为 v_s ,目的节点为 v_d 的路径,用排序的节点表示,例如 $p_{(v_s, v_d),1}=\{v_s, v_2, \dots, v_d\}$ 。在工业无线场景中,单路径传输往往无法保障关键流的确定性传输需求,多路径路由已被证明是一种有效的确定性保障技术。多路径路由 $r_{(v_s, v_d)}$ 定义为 $r_{(v_s, v_d)}=\{p_{(v_s, v_d),1}, p_{(v_s, v_d),2}, \dots, p_{(v_s, v_d),k}\}$,包含多条从源节点 v_s 到目的节点 v_d 的路径,但这些路径历经不同的中间节点或链路。传统的多路径路由往往采用不相交的多条路径,这种方法要求多条路径之间没有相交节点或者相交链路,在某些拓扑和流需求情况下可行解较少,难以匹配合适的资源调度结果。而且,这种方法往往占用更多节点和链路,对于资源受限的TSCH网络是一种需要权衡的负担^[11]。因此,本文采用相交的多路径路由,多条路径之间共享某些较高剩余能量的节点或者较高信道质量的链路,在满足关键流的确定性传输需求的同时相比严格不相交的多路径路由方案有更好的能耗表现。而且,PREOF支持相交的多路径路由在分离点的复制和聚

合点的删除,可进一步优化能耗性能。

TSCH针对确定性传输场景设计,采用时分多址与频分多址相结合的资源分配方式,实现时频二维资源的cell调度。如图1所示,时隙与信道组成可调度的最小单位cell(由时隙偏移和信道偏移共同确定)。调度表以时隙帧为周期运行,每个时隙帧包含固定数量的时隙,时隙数和信道数可根据实际应用需求自定义配置(典型配置为每个时隙帧包含101个时隙,2.4 GHz频段下可用16个信道)。时频资源块cell分为共享cell与专用cell两类。共享cell通过竞争方式占用资源,而专用cell则为指定的发送方与接收方预留,避免了竞争冲突,能够为点到点通信提供确定性保障。本文采用集中式调度方法,由控制层统一管理并调度专用cell资源,为关键业务流计算最优的时隙与信道分配方案,最大限度减少冲突,提升确定性传输性能。

在实际信道使用过程中,为应对无线信道中的多径衰落及外部干扰,TSCH引入了跳频机制。具体而言,每个时隙的开始会进行信道的映射,计算方法如下:

$$\text{frequency} = F[(\text{ASN} + \text{Ch}_{\text{offset}}) \bmod N_{\text{ch}}] \quad (1)$$

其中, $F[]$ 为将信道编号映射为实际通信信道的映射函数;绝对序列号(Absolute Sequence Number, ASN)为时隙编号; $\text{Ch}_{\text{offset}}$ 为信道偏移; N_{ch} 为可用信道数; \bmod 表示取模操作。TSCH网络通过全局统一的调度表进行调度和数据传输,为不同链路分配专用的时隙与信道资源。本文假设全网实现时钟同步,所有节点的时隙编号保持同步一致。值得注意的是,跳频操作本质上是对信道偏移与实际信道的统一映射,因此不会改变调度表的确定性传输特性。资源调度算法只需专注于时频二维资源的分配与管理,具体的信道分配则由跳频机制在执行时统一映射到实际信道。本文针对关键流的传输进行路由与调度的跨层联合优化,在提升流量调度成功率的同时优化能量利用效率。

1.3 流量模型

本文考虑 $|\mathcal{F}|$ 个关键数据流,每个流 $f_h \in \mathcal{F}$ 包含一个数据包,从源节点周期性生成,形成从普通节点到Sink节点的上行汇聚流量。但本文设计的调度算法同样适用于从Sink到传感器等普通节点的下行数据传输资源调度。流 f_h 的特征用6元组表示: $(v_{h,s}, v_{h,d}, r_h, p_h, d_h, \text{rel}_h)$,其中, $v_{h,s}$ 为源节点, $v_{h,d}$ 为目的节点; r_h 为流 f_h 的产生时间, p_h 为流 f_h 的周期,用时隙数量表示,流 f_h 由节点 $v_{h,s}$ 周期性生成,因此生成时间为 $r_h + p_h * k, k \in \mathbb{N}_0$;截止时间通常设为周期的一定比例,例如 $d_h = 0.8p_h$; rel_h 为流 f_h 的端到端可靠性需求,用包的到达率(Packet Delivery Ratio, PDR)表示。

考虑 TSCH 网络以时隙帧的长度周期运行,为兼容所有周期流的调度,设置时隙帧长度等于所有周期 p_h 的最小公倍数,即 $\text{sf} = \text{LCM}(p_h), \forall f_h \in \mathcal{F}$ 。

1.4 可靠性模型

为确保关键流的可靠传输,所选多路径路由提供的可靠性 $P_{p_{c^*}}$ 需满足或超过流 f_h 的可靠性需求 rel_h 。在多路径路由中,相交类型的多路径路由由可靠性计算比不相交类型的复杂,主要取决于多条路径之间相交的方式。本文采用文献[28]中的可靠性模型描述相交类型的多路径路由可靠性。

对于节点不相交类型的多路径路由,数据包端到端的成功传输只需在其中一条路径上传输成功即可:

$$P_{p_{c^*}}^{\text{series}} = \prod_{e \in p_{(v_m, v_n, k)}} (1 - \text{plr}_e) \quad (2)$$

$$P_{p_{c^*}}^{\text{SP}} = 1 - \prod_{i=1}^N \left(1 - \prod_{j=1}^M P_{p_{c^*}}^{\text{series}} \right) \quad (3)$$

对于链路不相交类型的多路径路由,数据包端到端的成功传输需要保证其在各段链路上都传输成功:

$$P_{p_{c^*}}^{\text{parallel}} = 1 - \prod_{k=1}^{|p_{(v_m, v_n, k)}} (1 - P_{p_{c^*}}^{\text{series}}) \quad (4)$$

$$= 1 - \prod_{k=1}^{|p_{(v_m, v_n, k)}} \left(1 - \prod_{e \in p_{(v_m, v_n, k)}} (1 - \text{plr}_e) \right)$$

$$P_{p_{c^*}}^{\text{PS}} = \prod_{j=1}^M \left(1 - \prod_{i=1}^N P_{p_{c^*}}^{\text{PP}} \right) \quad (5)$$

对于某些复杂情况下相交类型的多路径路由,其可靠性介于使用上述两类方法计算的结果之间:

$$P_{p_{c^*}}^{\text{PS}} \leq P_{p_{c^*}}^{\text{NSP}} \leq P_{p_{c^*}}^{\text{SP}} \quad (6)$$

$$\prod_{j=1}^M \left(1 - \prod_{i=1}^N P_{p_{c^*}}^{\text{PP}} \right) \leq P_{p_{c^*}}^{\text{NSP}} \leq 1 - \prod_{i=1}^N \left(1 - \prod_{j=1}^M P_{p_{c^*}}^{\text{series}} \right) \quad (7)$$

本文使用权重参数 $\{\alpha, \beta\} = \{0.5, 0.5\}$ 近似获取这种情况下多路径路由的可靠性:

$$P_{p_{c^*}}^{\text{NSP}} = \alpha \times \text{Upper} + \beta \times \text{Lower} \quad (8)$$

1.5 约束条件

TSCH 网络的资源调度需要综合考虑多项约束条件,如需满足关键流的确定性传输需求,满足节点半双工传输约束等。

(1) 半双工约束

所有节点只有一个处于半双工模式的无线收发机,因此在同一个时隙内,节点只能选择发送或者接收,即不能同时发送与接收:

$$\sum_{(v_m, v_n) \in O(v_m)} S_{f_h}^t(v_m, v_n) + \sum_{(v_m, v_n) \in I(v_m)} S_{f_h}^t(v_n, v_m) \leq 1, \quad (9)$$

$$\forall (v_m, v_n) \in \mathcal{V}, \forall f_h \in \mathcal{F}, \forall t \in T$$

其中, $O(v_m)$ 表示发送端是 v_m , 接收端是其他节点的

链路的集合; $I(v_m)$ 表示接收端是 v_m , 发送端是其他节点的链路的集合; $S_{f_h}^t(v_m, v_n)$ 表示在时隙 t 是否存在 cell 分配给链路 (v_m, v_n) , 如果存在则为 1, 反之为 0; $S_{f_h}^t(v_n, v_m)$ 表示链路 (v_n, v_m) 对应的资源分配情况。因此,约束(9)表示同一个时隙 t 节点 v_m 不能同时为发送节点和接收节点。

(2) 资源块约束

对于专用 cell 的调度,本文考虑一个 cell 只能调度一个传输过程,不考虑旁听转发等 cell 复用情况:

$$\forall c \in \{1, 2, \dots, C\}, t \in \{1, 2, \dots, T\}: \sum_{(v_m, v_n) \in c} x_{(v_m, v_n)}(c, t) \leq 1 \quad (10)$$

其中, $x_{(v_m, v_n)}(c, t)$ 表示链路 (v_m, v_n) 在时隙为 t , 信道为 c 的资源块 (c, t) 上传输。

(3) 干扰约束

处于干扰范围的两条链路容易产生传输错误,影响关键流的确定性调度。本文根据邻接节点干扰原则,若两条链路 (v_m, v_n) 与 (v_n, v_o) 存在公共节点 v_n , 则视为处于冲突范围,不能同时被分配到同一个 cell:

$$\forall v_m, v_n, v_o, (v_m, v_n) \in \mathcal{E}, (v_n, v_o) \in \mathcal{E}, \quad (11)$$

$$\forall c, t: x_{(v_m, v_n)}(c, t) + x_{(v_n, v_o)}(c, t) \leq 1$$

(4) 聚合点转发时间约束

在多路径路由的聚合节点(如果不是目的节点),由于不同路径存在传输时延差异,为确保确定性传输,需要等待所有路径中最晚到达的包到达后再进行转发调度,即转发 cell 的调度时间必须晚于所有路径上包的最迟到达时间:

$$\forall f_h \in \mathcal{F}, \forall (v_m, v_n) \in \mathcal{E}, \quad (12)$$

$$\forall p \in \mathcal{P}_f: s_{f_h, (v_m, v_n)}^p \geq \max_{p' \in \mathcal{P}_f} \left\{ s_{f_h, (v_m, v_n)}^{p'} \right\} + 1$$

其中, $s_{f_h, (v_m, v_n)}^p$ 表示路径 p 上流 f_h 在链路 (v_m, v_n) 调度的时隙编号。

图 2 详细展示了这种情况下的转发时间约束情况。从节点 v_1 到节点 v_5 有 3 条路径,分别是 (v_1, v_2, v_5) , (v_1, v_3, v_5) 和 (v_1, v_4, v_5) 。在 PREOF 功能的支持下,节点 v_1 对流 f_1 进行复制转发,复制的流分别从节点 v_2 、 v_3 、 v_4 转发给节点 v_5 。聚合节点 v_5 在数据包消除功能(Packet Elimination Function, PEF)的支持下删除冗余包,只保留一份包进行转发。假设副本在经过链路 (v_1, v_2) 和 (v_1, v_3) 时分别经历节点故障和链路故障导致包传输错误,在聚合节点 v_5 为其调度转发 cell 将导致资源浪费。因此,聚合节点 v_5 在调度转发 cell 时,需要将其预留在最晚到达包的时隙之后,约束(12)确保了聚合点的转发时隙在最晚到达包时隙之后。对于聚合节点 v_5 的资源调度,现有研究未考虑 PEF 机制对冗余包的删除,因此为多个冗余复制包同时预留转发资源,造成资源浪费。本文考虑聚合节点保留一

一个正确的包进行转发并为其分配 cell,从而使调度算法有效支持 PREOF。

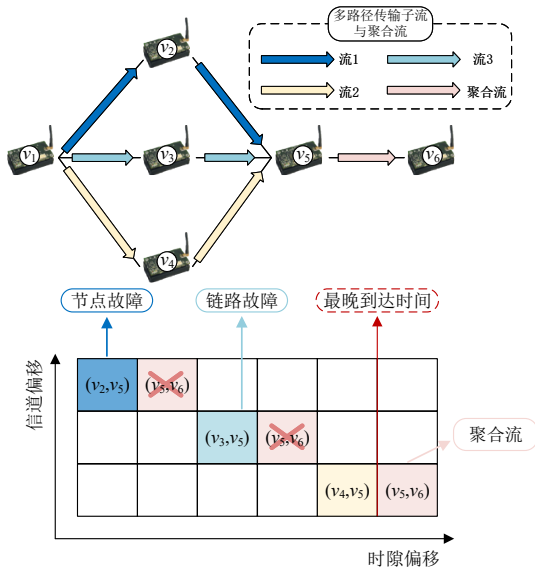


图2 聚合节点转发资源调度

Figure 2 Forwarding resource scheduling of aggregator nodes

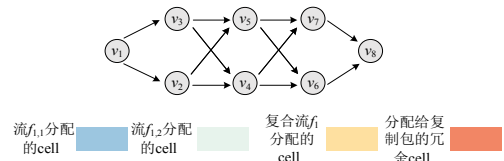
2 示例与问题描述

本节首先以具体示例揭示现有多路径路由与资源调度方案的局限性,进而简要概述本文提出的优化思路,然后对所研究的优化问题进行形式化定义与建模。

2.1 问题背景示例

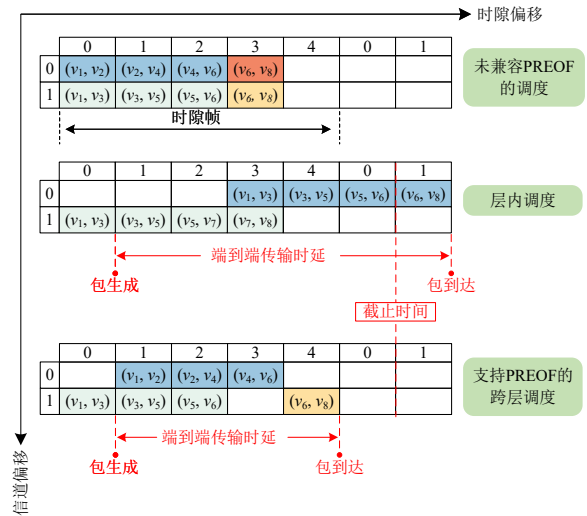
TSCH网络通过为关键业务预留资源提供确定性传输保障。由于单一路径难以满足高可靠性需求,因此,多路径路由技术被广泛应用。PREOF机制基于相交类型的多路径路由,在分离点实现数据复制,在聚合点进行冗余包的删除。然而,如果调度算法不支持PREOF,即在聚合点对冗余包删除以及相应转发资源的调度,就会出现为已删除的冗余包分配多余cell的情况,导致网络可调度性降低、带宽与能耗资源浪费。图3(b)给出了这一问题的示例。在图3(a)所示的网络拓扑中,流 f_1 在分离点 v_1 被复制后分别经两条路径 $(v_1, v_2, v_4, v_6, v_8)$, $(v_1, v_3, v_5, v_6, v_8)$ 传输,在调度算法不兼容PREOF的情况下,聚合节点 v_6 会为已经删除的包分配冗余的cell(橙色资源块),造成资源的浪费。此外,现有调度方案通常采用分层设计,即路由和调度分别独立完成,缺乏有效的信息交互和联合优化。典型的TSCH网络采用RPL协议进行路由选择,MSF算法实现媒体访问控制(Media Access Control, MAC)层调度,这种“层内调度”方式使路由选择无法感知底层资源状态,导致所选路由难

以在cell层面满足关键流的截止时间约束。图3(b)中展示了层内调度导致的这一问题:当选择的两条路径为 $(v_1, v_3, v_5, v_6, v_8)$ 和 $(v_1, v_3, v_5, v_7, v_8)$ 时,层内调度未考虑到链路 (v_1, v_3) 在时隙0上的资源紧张情况(假设 $cell(0, 0)$ 存在其他传输占用),因此路径 $(v_1, v_3, v_5, v_6, v_8)$ 的传输端到端时延超过了包的截止时间需求。在上述多路径传输过程中,PREOF机制本身引入的操作开销较小^[28-29],且通过如MRA等基于序列号的维护算法^[6,30],可进一步降低其计算与存储负担,从而有效减轻对端到端传输时延的影响。因此,在实际调度中可忽略该部分带来的开销,问题重点聚焦于如何高效组织多路径路由与cell资源的调度分配。针对这一问题,本文提出了一种支持PREOF机制的跨层调度方法,如图3(b)所示。该方法在聚合点联合优化路由与二层cell分配,兼容PREOF尤其是在聚合点支持PEF,对冗余包进行删除,可有效提升网络的可调度性并节约无线资源。



(a) 示例拓扑及流分配图例

(a) Illustrative example of topology and flow distribution



(b) 具体示例

(b) Illustrative example

图3 问题背景示例

Figure 3 A motivating example

为解决上述示例中现有方案的不足,并提升TSCH调度的可调度性与能量效率,本节对跨层优化的资源调度问题进行建模,并在后文提出优化算法加

以求解。

2.2 问题描述

在 TSCH 网络中,路由与调度的资源分配通过将链路映射到 cell 来实现。满足调度约束和流需求的情况下,各跳的 cell 分配级联构成流的端到端资源分配。本文将该过程建模为一个跨层资源分配优化问题,联合考虑基于 PREOF 的多路径路由选择与时频资源调度 cell 分配的协同优化,以最大化成功调度流的数量,在此基础上最大化资源利用效率。资源利用效率定义为成功调度流数量与占用 cell 数量的比值。

$$\max \sum_{h=1}^H \delta_{f_h} \quad (13)$$

$$\max \eta = \frac{N_{\text{flow}}}{N_{\text{cell}}} = \frac{\sum_{h=1}^H \delta_{f_h}}{\sum_{(u,v) \in E} \sum_{c=1}^C \sum_{t=1}^T x_{(u,v)}(c,t)} \quad (14)$$

s.t. (9)(10)(11)(12)

$$P_{p_{e^*}} \geq \text{rel}_h, \forall f_h \in \mathcal{F} \quad (15)$$

$$d_{(v_h, v_{h,d})} \leq d_h, \forall f_h \in \mathcal{F} \quad (16)$$

其中,目标函数(13)最大化成功调度流数量,(14)最大化资源利用效率;约束(9)~(12)用于规范 cell 的分配,确保无竞争的资源调度,保证关键流传输的确定性;约束(15)保证满足流的可靠性需求;约束(16)保证满足流的时延需求。

上述问题涉及跨层的路由选择与资源调度,变量众多且约束复杂,属于典型的组合优化问题,已被证明为 NP-Hard。即便采用启发式算法,也往往面临计算复杂度高、耗时长久的困境。近年来,DRL 因无需先验环境知识与完整系统模型,已广泛应用于通信网络中的资源优化、流量调度与路由等问题。DRL 智能体能够通过与环境的持续交互,自动提取特征并学习策略,从而近似获得最优决策。

3 算法设计

本节围绕 HRRRA 算法展开详细设计,包括整体架构设计、GNN 模型构建,以及基于 GNN 的 HRL 模型设计,从而阐释算法在多路径路由与调度联合优化中的实现思路。

3.1 架构设计

为实现高效的 TSCH 资源分配并保证确定性流量的传输,DRL 智能体通过“试错-反馈”的方式,在状态 $s \in \mathcal{S}$ 与动作 $a \in \mathcal{A}$ 之间学习最优映射,以最大化累计奖励 \mathcal{R} 。然而,多路径路由选择与 cell 分配构成的联合优化属于典型的组合难题,直接以单层 DRL 求解往往难以收敛。对此,HRL 提供了合适的框架^[19]:使高层策略负责多路径路由选择,低层策略根据高层决

策的路由为各跳分配 cell 资源。HRL 通过将原本耦合的联合动作空间划分为两个更小的子空间,分别由高层与低层策略独立决策,相比单层结构中需对路由与调度进行一体化决策的方式,至少减少了一个线性量级的动作维度^[31]。随着网络规模、候选路径数量或时隙帧长度的增长,这种分层结构带来的维度压缩优势将进一步放大。因此,这种两层策略以交替迭代的方式进行训练可有效降低搜索维度并提升收敛效率。同时考虑到网络拓扑属于非欧几何结构,且流量特性异构,为提高模型对复杂状态的表征与泛化能力,本文在高层策略中引入 GNN 提取输入状态特征^[32],形成基于 GNN 的 HRL 方法 HRRRA。HRRRA 通过 DRL 提升了对大规模状态空间的搜索能力,同时利用 GNN 增强模型泛化能力,可实现对多路径路由与调度的跨层联合优化。

3.2 GNN 模型设计

网络拓扑本质上属于图结构数据,传统 DRL 在处理此类数据时存在两方面不足:一是无法充分利用节点间的拓扑关系,导致性能提升受限;二是传统前馈神经网络的输入维度固定,当模型在某一固定规模拓扑上训练后,一旦节点数量或结构发生变化,模型难以识别并适应新的网络,从而缺乏泛化能力。为了解决上述问题,GNN 被设计用于处理图结构数据。因此,本文在 HRL 的高层策略中采用基于 GNN 的结构以提取网络拓扑特征。GNN 通过多层局部消息传递与逐层信息聚合,能够捕获节点间的复杂关联并生成高层次拓扑表示。具体而言,本文选用广泛应用的消息传递神经网络(Message Passing Neural Network, MPNN)对图结构数据进行编码,为高层策略提供输入特征。通过 GNN 对拓扑信息的有效提取,便于后续高层策略的多路径路由动作构建。

MPNN 通过迭代的消息传递机制在节点之间传播和聚合消息,主要包括消息传递、聚合更新和读出 3 个部分设计。在消息传递阶段,首先需对节点和边特征进行嵌入以便后续特征传递。对每个节点 v_m ,其初始特征向量定义为 $\mathbf{x}_{v_m} = [\text{src}_h, \text{dst}_h, \tilde{d}_s(v_m), \tilde{d}_t(v_m), \widehat{\text{slack}}_h(v_m)]^T$ 。其中, $\text{src}_h, \text{dst}_h \in [0, 1]$ 分别表示 v_m 是否为当前流 f_h 的源节点或目的节点; $\tilde{d}_s(v_m)$ 和 $\tilde{d}_t(v_m)$ 为到源节点和目的节点的最短跳数并按图直径归一化; $\widehat{\text{slack}}_h(v_m) = \max\{0, d_h - \hat{T}_h(v_m)\}$ 表示相对于 f_h 截止时间 d_h 的时间裕度。然后通过线性映射 $\mathbf{h}_m^{(0)} = \mathbf{W}\mathbf{x}_{v_m} + \mathbf{b}$ 投影为隐藏向量 $\mathbf{h}_m^{(0)} \in \mathbb{R}^d$,得到初始的隐藏向量。边的初始特征向量 $\mathbf{e}_{(m,n)}$ 定义为 $\mathbf{e}_{(m,n)} = [\widetilde{\text{plr}}_{(v_m, v_n)}, \widetilde{\text{cap}}_{(v_m, v_n)}, \tilde{t}_{\text{earliest}}(v_m, v_n), \widetilde{\text{betw}}_{(v_m, v_n)}]^T$ 。其中, $\widetilde{\text{plr}}_{(v_m, v_n)}$ 为链路的丢包率; $\widetilde{\text{cap}}_{(v_m, v_n)}$ 为可用 cell 占总 cell 的比例,表示链路 (v_m, v_n) 上 cell 的可用性;

$\tilde{t}_{\text{earliest},(v_m,v_n)}$ 表示最早可用时隙; $\widetilde{\text{betw}}_{(v_m,v_n)}$ 为链路介数中心性, 具体定义为对于所有 (src, dst) 对, 计算 k 个备选路径 (比如 k 个最短路径) 中, 每跳链路被经过的比例。该特征反映链路在多路径选择中的重要性, 该值越高, 表明该链路更可能成为流量聚集的关键链路。此特征的引入有助于 GNN 捕捉到网络中潜在的资源瓶颈, 从而提升路由决策的全局最优性。和节点特征一样, 将边特征经过线性映射 $\mathbf{e}_{(m,n)}^{(0)} = \mathbf{W}_e \mathbf{e}_{(m,n)} + \mathbf{b}_e$ 嵌入为统一维度 d , 便于后续消息融合。MPNN 利用迭代的消息传递算法在节点间传播消息。在消息传递阶段, 消息传递过程重复 T 次。具体而言, 在第 t 次迭代时, 节点 v_m 从邻居节点 v_n 收到隐藏状态 $\mathbf{h}_n^{(t-1)}$, 然后利用消息函数将这些消息和自己的隐藏状态 $\mathbf{h}_m^{(t-1)}$ 进行结合。对于每一条从邻居节点 $v_n \in \mathcal{N}(v_m)$ 指向 v_m 的边 (v_n, v_m) 计算消息向量:

$$\mathbf{m}_{n \rightarrow m}^{(t)} = m(\mathbf{h}_n^{(t-1)}, \mathbf{h}_m^{(t-1)}, \mathbf{e}_{(n,m)}^{(t-1)}) \quad (17)$$

其中, $\mathbf{h}_n^{(t-1)}$ 和 $\mathbf{h}_m^{(t-1)}$ 分别是上一层邻居节点 v_n 和节点 v_m 的隐藏向量, $\mathbf{e}_{(n,m)}^{(t-1)}$ 是上一层边的隐藏向量。消息函数 $m(\cdot)$ 由前馈神经网络实现。

在消息聚合阶段, 为了进一步增强模型对关键链路的关注, 本文采用图注意力网络 (Graph Attention Network, GAT) 对邻居节点的特征信息进行加权聚合。为进一步增强模型对关键链路的关注能力, GAT 在消息聚合阶段引入了可学习的注意力机制, 使模型能够根据邻居节点的重要性动态调整其信息贡献。例如, 当某条链路的丢包率显著低于其他邻居链路时, GAT 会赋予其更高的注意力权重, 从而增强其在节点状态更新过程中的影响力。该机制有助于模型识别出对全局可靠性更关键的局部结构信息。具体而言, 对于节点 v_m , GAT 会针对其每个邻居节点 v_n 计算注意力权重系数 $\alpha_{n \rightarrow m}^{(t)}$ 来表示节点 v_n 通过边 (v_n, v_m) 对节点 v_m 的重要性:

$$\alpha_{n \rightarrow m}^{(t)} = \frac{\exp\left\{\text{LeakyReLU}\left(\mathbf{w}^T \left[\mathbf{h}_m^{(t-1)} \parallel \mathbf{h}_n^{(t-1)} \parallel \mathbf{e}_{(n,m)} \right]\right)\right\}}{\sum_{k \in \mathcal{N}(v_m)} \exp\left\{\text{LeakyReLU}\left(\mathbf{w}^T \left[\mathbf{h}_m^{(t-1)} \parallel \mathbf{h}_k^{(t-1)} \parallel \mathbf{e}_{(k,m)} \right]\right)\right\}} \quad (18)$$

其中, $[\cdot]$ 表示向量级联, \mathbf{w} 为可训练参数向量。计算得到所有 $\alpha_{n \rightarrow m}$ 后, 节点 v_m 对所有邻居的消息通过聚合函数 $a(\cdot)$ 进行聚合生成信息 $\mathbf{M}_m^{(t)}$:

$$\mathbf{M}_m^{(t)} = a\left[m\left(\mathbf{h}_m^{(t-1)}, \mathbf{h}_n^{(t-1)}\right) \mid v_n \in \mathcal{N}(v_m)\right] \quad (19)$$

常见聚合函数包括求和、求平均和取最大值等。这里使用求和进行聚合:

$$\mathbf{M}_m^{(t)} = \sum_{v_n \in \mathcal{N}(v_m)} \alpha_{n \rightarrow m}^{(t)} \mathbf{m}_{n \rightarrow m}^{(t)} \quad (20)$$

在状态更新阶段, 节点 v_m 将自身上一层表示与聚合后的消息融合, 得到在第 t 层的新表示:

$$\mathbf{h}_m^{(t)} = f_m(\mathbf{h}_m^{(t-1)}, \mathbf{M}_m^{(t)}) \quad (21)$$

其中, f_m 是节点更新函数, 可用多层感知机 (Multi-layer Perceptron, MLP) 实现。具体来说, 将节点 v_m 原有表示 $\mathbf{h}_m^{(t-1)}$ 与聚合消息 $\mathbf{M}_m^{(t)}$ 拼接, 经过非线性变换得到更新后的 $\mathbf{h}_m^{(t)}$ 。为了稳定训练, 每层更新后加入激活函数 (如 ReLU) 和归一化操作。经过 T 层消息传递后, 每个节点的表示 $\mathbf{h}_m^{(T)}$ 包含了其 T 跳邻域内的拓扑和状态信息。一般设置 T 等于 2 或 3, 使节点能获取源和宿周围的局部拓扑信息以及中间跳的链路质量信息, 从而在隐藏向量中隐含的编码潜在路径的可靠性。

完成 T 层消息传递后, 获得每个节点的最终隐藏表示 $\mathbf{h}_m^{(T)}$, 以及在消息传递过程中融合了边特征的邻居关系。考虑到决策目标是在边上输出评分, 即判断哪些链路应被选入路由路径集合, 设计输出层以边为单位进行评分。具体地, 对于任意边 $(v_m, v_n) \in \mathcal{E}$, 利用其两端节点的最终表示和边自身特征来计算评分:

$$s_{(m,n)} = f_o(\mathbf{h}_m^{(T)}, \mathbf{h}_n^{(T)}, \mathbf{e}_{(m,n)}^{(T)}) \quad (22)$$

其中, f_o 是输出层的评分函数。将 $\mathbf{h}_m^{(T)}$ 与 $\mathbf{h}_n^{(T)}$ 以及边特征 $\mathbf{e}_{(m,n)}$ 拼接成一个向量, 然后通过全连接网络映射为标量分数 $s_{(m,n)}$ 。此分数经过 Sigmoid 激活函数压缩到 0~1 区间, 表示链路 (v_m, v_n) 作为路由的一条候选边的备选概率。

图 4 展示了链路之间的消息传递过程, 对应的伪代码在算法 1 中展示。

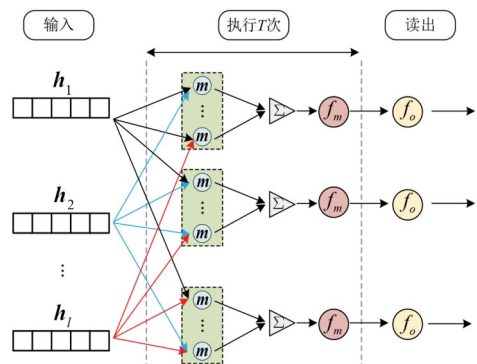


图 4 MPNN 结构

Figure 4 MPNN architecture

3.3 基于 GNN 的分层强化学习模型

本节将介绍基于 GNN 的 HRL 模型。首先对所提出的方法进行整体概述, 然后分别从高层策略和底层策略阐述状态、动作与奖励的设计, 最后介绍模型的训练过程。

算法 1 MPNN 消息传递算法输入: 网络拓扑 $G(\mathcal{V}, \mathcal{E})$ 节点嵌入向量 $\mathbf{h}_m^{(0)}$ 边嵌入向量 $\mathbf{e}_{(m,n)}^{(0)}$ 迭代次数 T 输出: 边评分 $s_{(m,n)}$ 1. FOR 训练轮次小于 T DO2. FOR $\mathbf{e}_{(m,n)} \in \mathcal{E}$ DO

3. 消息向量

$$\mathbf{m}_{n \rightarrow m}^{(t)} = m(\mathbf{h}_n^{(t-1)}, \mathbf{h}_m^{(t-1)}, \mathbf{e}_{(m,n)}^{(t-1)})$$

4. 消息聚合

$$\mathbf{M}_m^{(t+1)} = \sum_{v_n \in \mathcal{N}(v_m)} \alpha_{n \rightarrow m}^{(t)} \mathbf{m}_{n \rightarrow m}^{(t)}$$

5. 节点更新

$$\mathbf{h}_m^{(t+1)} = f_m(\mathbf{h}_m^{(t)}, \mathbf{M}_m^{(t+1)})$$

6. END

7. END

8. 读出函数进行边评分

$$s_{(m,n)} = f_o(\mathbf{h}_m^{(T)}, \mathbf{h}_n^{(T)}, \mathbf{e}_{(m,n)}^{(T)})$$

3.3.1 算法概述

为应对路由与调度联合优化中复杂性与可扩展性的挑战,本文采用分层强化学习框架,如图 5 所示(红色实线展示高层和低层的交互过程),将多路径路由选择建模为高层策略,将时频资源分配建模为低层策略,并通过两层策略的协同迭代实现整体优化。在保证 IIoT 关键流严格时延与可靠性需求的前提下,该框架旨在最大化可调度流数量与资源利用效率。高层策略的多路径路由选择和低层策略的调度决策结果作为动作反馈给环境,环境模拟基于 PREOF 机制的多路径传输过程,并动态维护与传输性能相关的统计信息,进而生成奖励信号反馈至高层与低层策略,用于共同优化路径选择与资源分配策略。通过这种反馈机制,实现对 PREOF 机制的自适应支持。具体实现中,利用深度强化学习的试错式累积奖励机制以探索最优资源分配策略,同时在高层策略中结合 GNN 以提取网络拓扑的高阶特征,从而提升对复杂状态空间的搜索能力,并增强模型在不同拓扑规模下的泛化性能。

3.3.2 高层策略

高层策略 π_h 负责为当前流 f_h 选择多路径路由。在每个调度周期开始时,系统首先根据各流的端到端时延需求对流集合进行排序,优先为截止期限更近的流分配资源。然后 GNN 对网络当前状态特征进行提取,输出网络特征嵌入,结合当前网络调度状态和当前流的需求信息,构成高层策略的状态空间。具体而言,高层策略 π_h 的状态空间定义如下:

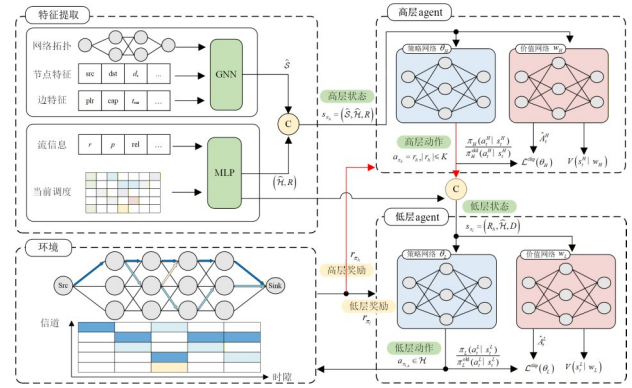


图5 HRRR算法框架图

Figure 5 Framework of the HRRR algorithm

$$\mathcal{S}_{\pi_h} = \left\{ s_{\pi_h} = \left(\hat{\mathbf{s}}, \hat{\mathcal{H}}, R \right) \right\} \quad (23)$$

其中, $\hat{\mathbf{s}} \in \mathbb{R}^{|\mathcal{E}|}$ 表示归一化的链路评分向量,由 GNN 对网络状态当前拓扑与流需求等因素综合计算后输出得到,其中每一个分量反映当前边 e 作为候选多路径路由组成部分的优劣程度,作为高层策略 π_h 动作选择的依据。 $\hat{\mathcal{H}} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{C}|}$ 表示归一化的 TSCH 调度表矩阵,其中 $|\mathcal{T}|$ 表示时隙数, $|\mathcal{C}|$ 表示信道数。虽然高层策略并不直接调度 cell 的资源分配,但是调度表提供当前资源分配的全局视图,帮助高层策略 π_h 判断哪些节点或链路的负载较高、可用资源较少,从而引导高层策略 π_h 选择更可能满足确定性传输需求的路由。 R 表示当前调度流的可靠性传输需求。每个时间步 t , 高层策略从 GNN 和环境获取状态 s_{π_h} 。

高层策略 π_h 将基于当前状态 s_{π_h} 选择多路径路由。高层策略 π_h 的动作空间 \mathcal{A} 定义为

$$\mathcal{A}_{\pi_h} = \left\{ a_{\pi_h} = r_h = \{p_1, p_2, \dots, p_i, p_j\}, |r_h| \leq K \mid \forall i \neq j, p_i \neq p_j, 0 < |p_i \cap p_j| < \min\{|p_i|, |p_j|\} \right\} \quad (24)$$

其中,每个动作 a_{π_h} 表示为流 f_h 从 src_h 到 dst_h 选择 K 条部分相交的多路径路由; $p_i \neq p_j$ 表示任意两条路径不完全重叠; $0 < |p_i \cap p_j| < \min\{|p_i|, |p_j|\}$ 保证多路径之间具有路径冗余以提升可靠性,同时允许部分重叠节省能耗且增加可调度性。低层策略 π_l 将基于高层策略 π_h 选择的动作为流 f_h 经过的链路进行时频资源的调度。

为了奖励高层策略 π_h 选择满足流 f_h 可靠性传输需求以及能耗更低的多路径路由,定义惩罚动作集 \mathcal{P}_{π_h} 。不满足可靠性需求的动作 $\mathcal{P}_{\pi_h, \text{un_rel}} \subset \mathcal{P}_{\pi_h}$ 定义为

$$\mathcal{P}_{\pi_h, \text{un_rel}} = \left\{ a_{\pi_h} \mid a_{\pi_h} \in \mathcal{A}_{\pi_h}, \text{PDR}(a_{\pi_h}) < \text{rel}_h \right\} \quad (25)$$

其中, $\text{PDR}(a_{\pi_h}) < \text{rel}_h$ 表示所选多路径路由不满足流 f_h

的可靠性传输需求。第二种惩罚动作集 $\mathcal{P}_{\pi_h, \text{dis}} \subset \mathcal{P}_{\pi_h}$ 为完全不相交的多路径路由。PREOF 支持相交的多路径路由在聚合点进行冗余数据包的删除,在满足流量确定性传输需求的同时,这一功能可降低能耗,有效支持 PREOF 在资源受限的 TSCH 网络中的应用。因此,高层策略 π_h 对不相交的多路径路由进行惩罚,从而奖励选择部分相交的多路径路由。这种惩罚动作集表示为

$$\mathcal{P}_{\pi_h, \text{dis}} = \left\{ a_{\pi_h} \mid a_{\pi_h} \in \mathcal{A}_{\pi_h}, \forall p_i, p_j \in a_{\pi_h}, i \neq j, p_i \cap p_j = \emptyset \right\} \quad (26)$$

其中, $\forall p_i, p_j \in a_{\pi_h}, i \neq j, p_i \cap p_j = \emptyset$ 表示多路径路由的路径完全不相交,这种类型的多路径路由能耗较高,因此视为惩罚动作。综上两种惩罚动作,惩罚动作集表示为 $\mathcal{P}_{\pi_h} = \{ p_{\pi_h} \mid p_{\pi_h} \in \mathcal{P}_{\pi_h, \text{un_rel}} \cup \mathcal{P}_{\pi_h, \text{dis}} \}$ 。

当策略 π_h 选择的动作属于惩罚动作集 $p_{\pi_h} \in \mathcal{P}_{\pi_h}$ 时, episode 将立即停止,并接收惩罚 ψ_{π_h} 。当 agent 成功完成所有流量的调度时,也会到达终止状态。因此,策略 π_h 的即时奖励 r 定义为

$$r_{\pi_h}(s_{\pi_h}, a_{\pi_h}) = \begin{cases} \psi_{\pi_h}, & \text{if } a_{\pi_h} \in \mathcal{P}_{\pi_h} \\ M \cdot \delta_{f_h} + \frac{\delta_{f_h}}{\sum_{(u,v) \in E} \sum_{c=1}^C \sum_{t=1}^T x_{(u,v)}(c,t) + \epsilon}, & \text{otherwise} \end{cases} \quad (27)$$

其中, δ_{f_h} 表示当前流 f_h 是否被成功调度,若满足则取 1,否则为 0。第二项为资源利用效率目标, ϵ 为防止分母为 0 的极小正数。常数 M (如取 100) 保证在同能效条件下,优先最大化成功调度流数量,确保第一目标优先。资源利用效率作为第二目标,仅在成功调度流数量相同时进一步优化。

高层策略 π_h 需要根据低层策略 π_l 对流 f_h 的资源分配结果,综合评估流 f_h 的需求是否满足,不仅鼓励满足端到端需求、成功调度的流量,也激励在保证确定性需求的前提下实现更高的资源利用效率。通过最大化累计奖励 \mathcal{R} 引导 agent 找到最优策略:

$$\pi_h^* = \arg \max_{\pi_h} \mathcal{R} = \arg \max_{\pi_h} \sum_{t=0}^T \lambda^t r_{\pi_h}(s_{\pi_h}, a_{\pi_h}) \quad (28)$$

其中, λ 是折扣因子,控制未来奖励的影响程度。

上述为高层策略状态空间、动作空间和奖励函数的设计。

3.3.3 低层策略

低层策略 π_l 负责依据高层策略 π_h 选择的多路径路由方案,为每条路径上的链路逐跳进行 cell 资源调度。低层策略 π_l 状态空间定义如下:

$$S_{\pi_l} = \left\{ s_{\pi_l} = (R_h, \hat{\mathcal{H}}, D) \right\} \quad (29)$$

其中, R_h 表示高层策略 π_h 为当前流 f_h 选择的多路径路由; $\hat{\mathcal{H}}$ 表示当前 TSCH 调度表的占用状态矩阵,反映网络当前 cell 的分配情况; D 表示当前流 f_h 的端时延需求。

低层策略 π_l 每个时间步从环境中获取上述状态,以链路为单位,从源节点逐跳向目的节点轮询执行 cell 分配动作。具体而言,动作空间定义为

$$\mathcal{A}_{\pi_l} = \left\{ a_{\pi_l} \mid a_{\pi_l} \in \mathcal{H}, \text{s.t. (9)(10)(11)} \right\} \quad (30)$$

其中, \mathcal{H} 是所有可能的 cell 集合 $h = (t, c)$ 。为了确保所选 cell 能够满足网络调度的基本约束,每个动作 a_{π_l} 必须满足半双工约束 (9)、资源块唯一约束 (10) 和干扰约束 (11),这些筛选操作可通过简单的集合查找与位掩码判断高效实现,并在每次资源分配后进行增量更新,其计算复杂度与候选 cell 数量呈线性关系,且不依赖于整个二维时隙帧的规模。

为提升训练早期的收敛效率,引入预期时延概念。基于目前已完成的部分链路调度,估算最短剩余传输延迟,若当前调度依然在截止时间内完成传输,则给予正奖励;反之,给予负反馈,引导低层策略规避不可达动作。对于当前动作 a_{π_l} ,其对应的时隙为 $\zeta_{a_{\pi_l}}$,对于当前调度路径 $p \in R_h$ 剩余未调度条数为 L_p^{rem} ,因此当前路径上最早到达时间为 $\hat{d}_p^{(i)} = \zeta_{\text{now}} + L_p^{\text{rem}}$,如果这个时间已经大于截止时间需求 D ,显然这个路径上的后续调度无法满足流 f_h 的确定性需求,给予其负奖励值。因此低层策略 π_l 的即时奖励 r 定义为

$$r_{\pi_l}(s_{\pi_l}, a_{\pi_l}) = \begin{cases} \psi_{\pi_l}, & \text{if } \hat{d}_p^{(i)} \leq D \\ -\psi_{\pi_l}, & \text{otherwise} \end{cases} \quad (31)$$

当低层策略 π_l 完成对多路径路由 R_h 的全部链路的 cell 调度后,会根据延迟约束给予延迟奖励,具体约束包括:聚合节点转发约束 (12) 和端到端截止时间约束 (16)。若未满足延迟约束,则给予显著的负奖励 $\alpha \psi_{\pi_l}$,否则给予适当的正向奖励,以激励策略在满足基本约束的同时优化延迟性能。因此惩罚动作集定义为

$$\mathcal{P}_{\pi_l, \text{delay}} = \left\{ a_{\pi_l} \mid a_{\pi_l} \in \mathcal{A}_{\pi_l}, a_{\pi_l} \nmid (12), (16) \right\} \quad (32)$$

延迟奖励因此定义为

$$r_{\pi_l, \text{delay}}(s_{\pi_l}, a_{\pi_l}) = \begin{cases} \alpha \psi_{\pi_l}, & \text{if } a_{\pi_l} \in \mathcal{P}_{\pi_l} \\ \alpha, & \text{otherwise} \end{cases} \quad (33)$$

其中, α 为权重参数。

最终每个时间步的总奖励包括即时奖励和延迟奖励两部分,即在 episode 未结束时只考虑即时奖励; episode 结束时,再将延迟奖励纳入总奖励中,确保奖励能够准确反馈策略的长期表现。因此,总奖励定

义为

$$r(s, a) = r_{\text{im}}(s, a) + I_{\text{episode_end}} \cdot r_{\text{delay}}(s, a) \quad (34)$$

其中, $I_{\text{episode_end}}$ 为指示函数, 仅在 episode 结束时取 1, 否则为 0。

3.3.4 模型训练

算法 2 和算法 3 展示了 HRRA 的整体流程。考虑在大规模动作空间及动态可行动作集场景下所展现出的良好训练稳定性与优化效率, HRRA 选择采用近端策略优化 (Proximal Policy Optimization, PPO) 算法^[33]。尽管 HRRA 的结构可兼容其他 DRL 方法, 但 PPO 更适用于本任务中的层次策略建模与高效收敛需求。具体流程而言, 将分层强化学习划分为高层 (路由) 和低层 (调度) 两级策略, 并对两级策略分别优化以实现协同收敛。算法 2 描述训练阶段的数据收集与交互过程。训练开始时, 分别初始化高层和低层的网络参数与相关缓冲区。每个训练轮次, 首先按截止时间对流集合排序, 然后利用 GNN 提取流的拓扑与业务特征并构造高层状态。高层策略据此生成动作集合并选择多路径路由动作, 确定该流的路径集合。在此基础上, 低层依据高层输出的路径、当前调度状态和流需求逐跳分配 cell 资源, 直到该流所有链路完成调度。交互过程中, 高层与低层各自将轨迹存入缓冲区。每轮交互结束后, 调用算法 3 对高层与低层策略分别进行 PPO 批量更新。随着训练迭代, 两级策略协同改进, 最终在满足关键流确定性 (时延和可靠性) 约束的同时, 实现高效的 TSCH 资源利用与稳定收敛。

4 性能评估

4.1 测试环境

本节基于仿真软件模拟了关键流在工业无线网络场景中的传输调度, 对 HRRA 算法进行训练, 并对所提算法和基准方案在仿真软件中进行了调度结果的运行和评估。本节目标是通过路径选择和 cell 资源的分配, 满足关键数据流确定性传输需求, 包括端到端的时延约束和可靠性需求。因此本节对比了不同拓扑规模和流量数量下各算法的性能表现。首先, 实验采用网格拓扑, 节点规模设为 20~100 个, 其中, 单个汇聚节点负责数据收集, 其余节点既能发送数据也能中继转发。考虑到实际工业场景中普遍存在的无线干扰与环境不确定性, 本文在仿真中将每条链路的可靠性设为 0.7~0.95 范围内的随机值, 以更真实地反映链路质量的动态波动。对于关键流的传输, 单条路径通常无法满足其可靠性需求, 因此需要为其分配多条路径, 通过路径冗余传输提升传输可靠性。其次, 为模拟流量的突发性与异构性, 实验设置中 70%

算法 2 HRRA 多路路由和调度算法-训练流程

输入: 网络拓扑 G , 关键流集合 \mathcal{F} , 奖励的权重 β , 折扣因子 γ , 探索

参数 $\epsilon_{\text{max}}, \epsilon_{\text{min}}, \epsilon_{\text{decay}}$, 训练轮次 H , 批量大小 \mathcal{F}

输出: 满足关键流确定性调度的多路路由和调度的最优策略

π_h, π_l

初始化高层策略参数 θ_h 和低层策略参数 θ_l

初始化缓存, 初始化探索率 $\epsilon \leftarrow \epsilon_{\text{max}}$

1. FOR 训练轮次小于 H DO

2. 重置环境并初始化调度表 \mathcal{H} 与流集合 \mathcal{F}

3. 按照截止时间优先给流排序

$\mathcal{F} \leftarrow \text{sort_by_deadline}(\mathcal{F})$

4. FOR 流 $f \in \mathcal{F}$ DO

5. 获取高层状态 $s_{\pi_h} = (\hat{S}, \hat{\mathcal{H}}, R)$

6. 选择多路径路由动作 $a_{\pi_h} = \{p_1, p_2, \dots, p_k\}$

7. WHILE f 满足可靠性需求且调度表能满足带宽需求 DO

8. 获取低层状态 $s_{\pi_l} = (R_h, \hat{\mathcal{H}}, D)$

9. 通过动作掩码生成可行动作集合

$$\mathcal{A}_{\pi_{l_e}} = \{a_{\pi_{l_e}} \mid a_{\pi_{l_e}} \in \mathcal{H}, \text{s.t.}(9)(10)(11)\}$$

10. 选择动作 $a_{\pi_l} \in \mathcal{A}_{\pi_{l_e}}$ 进行 cell 分配

11. 获得即时奖励 r_{π_l} , 转移到下一状态 s'_{π_l}

12. 存储低层的轨迹 $(s_{\pi_l}, a_{\pi_l}, r_{\pi_l}, s'_{\pi_l})$

13. END

14. 计算高层奖励 $r_{\pi_h}(f) = (1 - \beta) \cdot I(f) + \beta \cdot EE(f)$, 转移到下一状态 s'_{π_h}

15. 存储高层的轨迹 $(s_{\pi_h}, a_{\pi_h}, r_{\pi_h}, s'_{\pi_h})$

16. END

17. 调用算法 3 进行策略更新

18. 更新探索率 $\epsilon \leftarrow \max(\epsilon_{\text{min}}, \epsilon \cdot \epsilon_{\text{decay}})$

19. END

为具有不同周期和截止时间的关键流, 剩余 30% 为背景流量。考虑到典型工业互联网场景对时延和可靠性的要求^[34-35], 实验中关键流的周期设置为 [8, 10, 15, 20] 个时隙, 可靠性指标为 0.99。背景流量包括事件驱动流量和尽力而为流量: 事件驱动流量模拟突发事件, 采用泊松过程生成 ($\lambda = 0.1$); 尽力而为流量采用随机生成方式, 模拟网络中的常规流量。实验针对关键流进行路由和调度, 模拟不同算法在资源受限情况下对关键流需求的保障能力。在 MAC 层, TSCH 网络配置如下: 时隙时长为 10 ms, 时隙帧长度由关键流的周期决定, 可用信道为 16 个。在 HRRA 算法中, 低层策略使用带有注意力机制的 MLP, 高层策略采用 3 层 GAT 作为拓扑特征编码结构。根据实验测试结果 (如表 2 所示) 和现有文献研究^[36-37], 使用 2 层 GAT 时, 由于感受野有限, 难以充分捕获多跳 TSCH 网络中长路径的结构信息; 而 4 层 GAT 则容易引发典型的

算法3 HRRR策略更新

输入: 高层轨迹缓存, 低层轨迹缓存, 批量阈值 B_H, B_L , 迭代次数

K_H, K_L , 学习率 α_H, α_L , 裁剪参数 ϵ

输出: 更新后的策略参数 θ_H, θ_L

1. IF 高层轨迹长度大于 B_H THEN
2. 计算高层优势 \hat{A}_H 及旧策略概率 π_H^{old}
3. FOR 迭代次数小于 K_H DO
4. 计算比值 $r = \frac{\pi_H(a_H|s_H)}{\pi_H^{\text{old}}(a_H|s_H)}$
5. 计算剪切目标损失 $\mathcal{L}_H = \mathbb{E}[\min(r\hat{A}_H, \text{clip}(r, 1-\epsilon, 1+\epsilon)\hat{A}_H)]$
6. 更新参数 $\theta_H \leftarrow \theta_H + \alpha_H \nabla_{\theta_H} \mathcal{L}_H$
7. END
8. 清空高层轨迹缓存
9. END
10. IF 低层轨迹长度大于 B_L THEN
11. 计算低层优势 \hat{A}_L 及旧策略概率 π_L^{old}
12. FOR 迭代次数小于 K_L DO
13. 计算比值 $r = \frac{\pi_L(a_L|s_L)}{\pi_L^{\text{old}}(a_L|s_L)}$
14. 计算剪切目标损失 $\mathcal{L}_L = \mathbb{E}[\min(r\hat{A}_L, \text{clip}(r, 1-\epsilon, 1+\epsilon)\hat{A}_L)]$
15. 更新参数 $\theta_L \leftarrow \theta_L + \alpha_L \nabla_{\theta_L} \mathcal{L}_L$
16. END
17. 清空低层轨迹缓存
18. END

过平滑问题, 并伴随梯度震荡, 导致节点表示趋于一致, 从而削弱路由判别能力。相比之下, 3层 GAT 在特征表达能力与训练稳定性之间实现了更优平衡。训练过程中, 使用 PPO 算法, 学习率为 3×10^{-4} , 折扣因子为 0.99, 训练共进行 10 000 个 episode, 批次大小为 64。其余网络参数和算法参数设置如表 3 所示。仿真在工作站环境进行, 硬件配置为 AMD R9 5950X CPU @ 3.40 GHz, 软件环境基于 TSCH 离散时间仿真器 6TiSCH Simulator^[38] (develop 分支)、Python3.11 和 PyTorch2.7.1。

表2 不同层数 GAT 训练结果

Table 2 Training results of GAT with different numbers of layers

GAT层数	路由准确率/%	训练稳定性
2层	86.1	稳定
3层	90.3	稳定
4层	87.5	梯度震荡、过平滑

本文基准方案设置如下:

(1) RPL+MSF^[9]。其中, RPL 是 LLN 标准单路径路由, 结合 MSF 进行分布式最小时隙分配。该方案为业界常用的分层设计基准方案。

(2) ODeSe+TASA^[14]。其中, ODeSe 基于 RPL 进行

表3 网络和模型参数设置

Table 3 Network and model parameter settings

网络参数值	
网络节点数	[20, 40, 60, 80, 100]
每场景测试次数	20
链路可靠性	[0.7, 0.95]
流量设置	
数量	[500~3 750]
周期/时隙	[8, 10, 15, 20]
截止期限/周期	0.8
可靠性需求	0.99
TSCH 设置	
时隙长度/ms	10
时隙帧长度/时隙数	120
信道数量	16
最大重传次数	3
模型参数值	
学习率	3×10^{-4}
折扣因子	0.99
episode 数量	10 000
批次大小	64

了多父节点的扩展, 为流选择多条传输路径; TASA 则是经典的集中式按流量需求分配时隙算法。实验将两者结合形成典型的多路径路由与调度的分层决策方案。

(3) EDF-MO^[15]。该方案基于最小重叠 (Minimal-Overlap, MO) 单路径路由选择以降低中继节点重叠, 配合最早截止优先 (Earliest-Deadline-First, EDF) 策略分配时隙资源, 形成启发式跨层优化方案。

(4) ILP3+TASA^[16]。该方案在 TASA 基础上改进, 将单路径路由选择与时频资源分配统一建模为一个 ILP 线性规划问题并联合求解, 从而实现跨层优化。

(5) DGRL+MWIS^[22-23]。该方案结合了 DGRL 路由器与 MWIS 调度器。其中, DGRL 利用 GCN 编码全局网络拓扑, 并基于 DDPG 框架生成集中式路由策略; MWIS 调度器则通过 GCN 对冲突图进行表征学习, 结合启发式图搜索算法, 实现干扰约束下的高效时隙分配。实验将两者结合形成分层 DRL 方案。

在实验中, 随机生成背景流与关键流需求, 各方案对关键流进行资源分配计算, 然后对所有结果进行评估。评估主要针对成功调度流数量、端到端时延和资源利用效率指标。成功调度指对关键流的资源分配同时满足其时延约束和可靠性需求; 端到端时延展示所有成功调度流 (即满足确定性需求下) 的时延传输情况; 资源利用效率指成功调度流数量与其占用 cell 数量的比值, 用于反映调度效率。

4.2 实验结果与分析

(1) 调度成功率。图 6 展示了不同网络规模下各算法的调度成功率。在所有拓扑规模下 HRRR 方案均取得最高值。在同一规模内, DGRL+MWIS 性能仅次于 HRRR, 其次是 ODeSe+TASA、EDF-MO 和 ILP3+TASA, RPL+MSF 性能表现最差。这是由于 HRRR 采用支持 PREOF 的多路径路由和路由与调度的跨层优化, 多路径路由在多条路径上并行传输同一数据流, 更能满足关键流对时延和可靠性的确定性需求。跨层优化协同路由和调度信息, 对路由和调度进行优化决策, 因此 HRRR 性能相比单路径路由和分层优化更具优势。DGRL+MWIS 方案中 DGRL 路由器能感知拓扑和链路队列信息, MWIS 调度器可以有效避免链路冲突, 因此效果优于 ODeSe+TASA。但是其路由与调度的分层优化, 没有反馈机制, 在路径分布过度集中的情况下, 调度仍可能因冲突而牺牲部分任务, 因此效果整体弱于 HRRR。ODeSe+TASA 基于多路径路由, 但其缺乏路由与调度的协同机制, 性能弱于 HRRR 和 DGRL+MWIS。EDF-MO 和 ILP3+TASA 均为基于单路径路由的联合优化方法, 性能表现相近。RPL+MSF 基于单路径路由, 且路由与调度分开决策, 因此在保障确定性传输能力方面表现最差。值得注意的是, 随着节点数量从 20 增至 100, HRRR 相较于基准算法的优势显著增加。这是由于网络规模的增加, 调度搜索空间与冲突组合呈指数增长, 而 HRRR 的跨层联合优化可将路由和 cell 分配进行协同计算, 在大规模场景中更能发挥优势。相比于 DGRL+MWIS 和 EDF-MO, HRRR 方案在 100 节点规模的拓扑下, 分别增加了 10.6% 和 36.6% 的调度成功率。

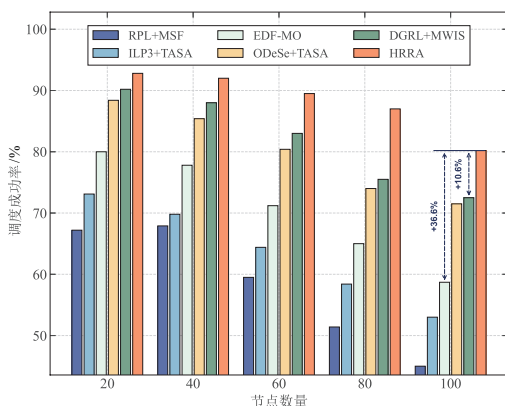


图 6 调度成功率结果

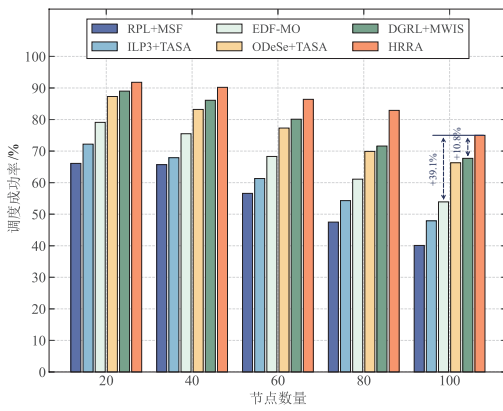
Figure 6 Scheduling success ratio results

当前实验将截止时间设置为流周期的 0.8 倍, 旨在确保所有方法在具备基本可调度性的前提下进行公平比较。我们进一步测试了将该比例依次压缩至 0.7、0.6 和 0.5, 实验结果如图 7 所示。可以看到,

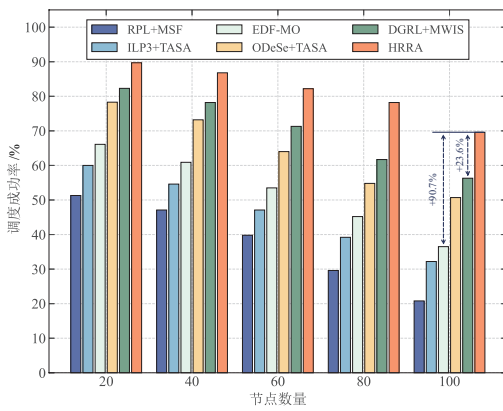
HRRR 在不同截止时间比例下始终保持性能优势, 且在约束严格的场景中(如图 7(c)的 0.5 倍), HRRR 的相对提升幅度进一步扩大, 体现出其在高约束条件下更强的资源适配能力。当截止时间与周期的比值持续减小时, 等价于绝对截止上限不断收紧, 该上限并非由算法自身决定, 而是由网络物理结构、时隙帧长度、路径跳数和可用 cell 数共同约束。因此, HRRR 的时延适配上限由可行调度的物理下界决定, 而在所有物理允许的截止时间范围内, HRRR 通过跨层马尔可夫决策过程建模和细粒度 cell 分配策略, 能够在更大的可行时间窗口内持续获得优于基准算法的性能。

(2) 端到端时延。图 8 展示了在 80 节点拓扑中, 在不同流数量下各算法的调度时延表现。随着流数量从 240 增加到 1 200, 网络资源占用与调度冲突不断增加, 所有算法的端到端时延均呈上升趋势。整体而言, HRRR 在全部流数量场景下始终保持最低时延。HRRR 通过跨层策略协同优化和多路径传输机制, 有效缓解拥塞并缩短关键流传输路径, 从而实现更优的时延控制。DGRL+MWIS 方案中, DGRL 路由器具备图感知能力, 可根据拓扑结构和队列状态优化路径选择, 避免流量集中; MWIS 调度器则通过 GCN 提升冲突图上独立集的选择质量, 从而增强并发调度能力, 降低链路排队与冲突延迟。ODeSe+TASA 同样利用多路径路由, 通过空间冗余并行转发部分关键流, 实现一定程度的时延降低。相比之下, 单路径方案(EDF-MO、ILP3+TASA 和 RPL+MSF)在丢包场景中只能依赖链路重传保障可靠性, 导致端到端时延显著上升。当流数量达到 1 200 条时, HRRR 的平均时延相比 DGRL+MWIS 和 EDF-MO 分别降低了 9.88% 与 25.5%, 显示出其跨层联合优化在高负载条件下的显著优势。

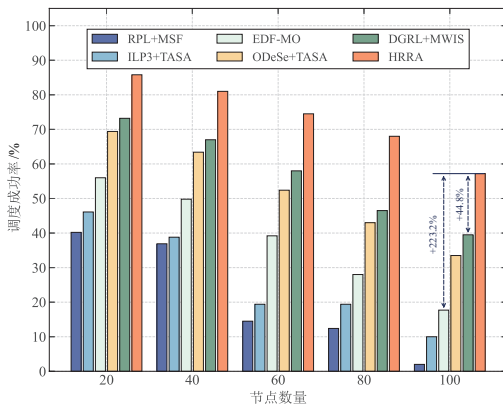
(3) 资源利用效率。图 9 展示了不同网络规模下各算法在关键流调度中的资源利用效率表现。DGRL+MWIS 通过基于图神经网络的最大独立集优化算法, 在冲突图约束下高效筛选可并发传输的链路集合, 有效减少资源碎片和时隙浪费, 从而提高资源利用率。ODeSe+TASA 依赖多路径冗余传输增强可靠性, 但未对资源冗余进行有效控制, 存在一定的重复调度开销。单路径方案(EDF-MO、ILP3+TASA 和 RPL+MSF)则需通过频繁的链路重传保障可靠性, 在流密度较高时导致大量无效或重复的 cell 分配, 资源利用效率显著下降。相比之下, HRRR 在所有网络规模下始终保持最高的资源利用率, 这是由于 HRRR 在路由和调度联合优化的框架下引入了对 PREOF 的支持, PREOF 能够有效删除冗余数据, 而联合优化则能够合理运用相交的多路径结构组成路由, 从而在充分利用和分配 cell 资源的基础上, 实现更高的资源利用效率。测试结果表明在 100 节点场景中, HRRR 平均



(a) 0.7
(a) 0.7



(b) 0.6
(b) 0.6



(c) 0.5
(c) 0.5

图7 调度成功率在不同截止时间/周期倍数情况下的结果

Figure 7 Scheduling success ratio results under different deadline/period multiples

资源利用效率相比于 DGRL+MWIS 方案和 EDF-MO 算法分别提升了 9.9% 和 35.1%。

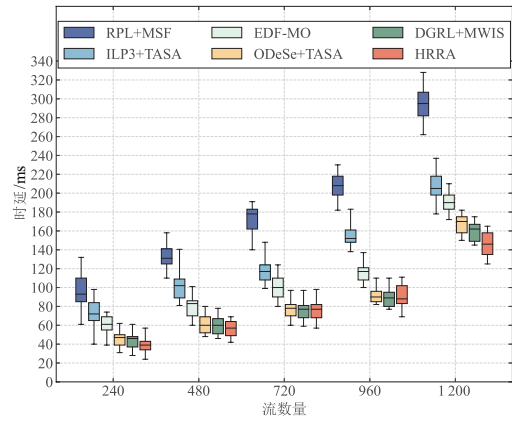


图8 时延结果

Figure 8 Delay results

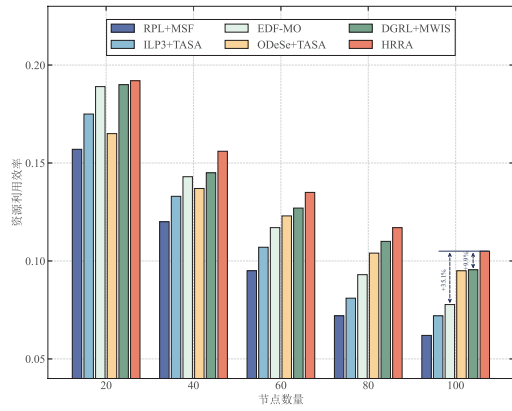
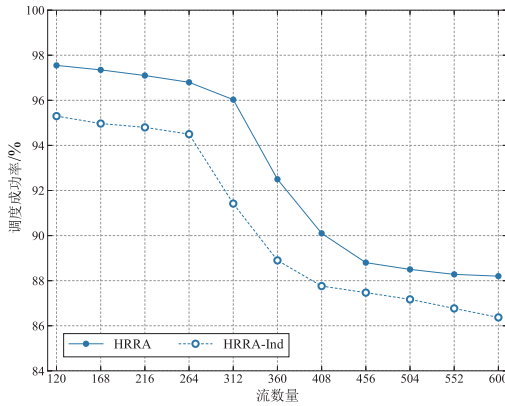


图9 资源利用效率结果

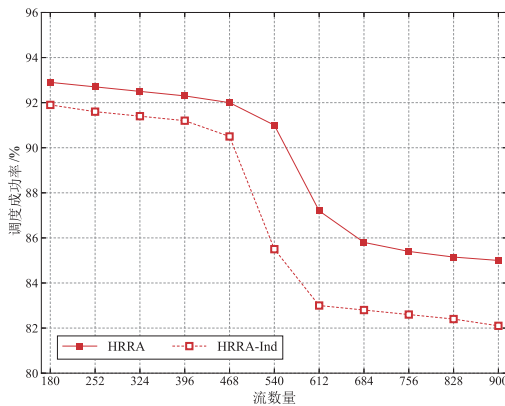
Figure 9 Resource utilization efficiency results

(4) 聚合点副本消除。PREOF 机制在多路径传输中,在分离节点进行数据包的复制和转发,在聚合节点汇聚各路径到达的包并删除冗余,从而减少 cell 浪费并提高资源利用效率。HRRR 通过跨层联合优化路由与调度,并在 cell 资源调度时支持 PREOF,能够在多路径路由下实现高效的资源利用。然而现有研究尚无面向 PREOF 的调度机制,实际实现为将同一数据流在多条路径上的传输视为若干独立数据流,为其独立分配 cell 资源。为评估 PREOF 对关键流调度成功率的影响,本文测试了“无 PREOF”和“有 PREOF”的消融实验。在 HRRR 中关闭 PREOF 功能模拟未支持 PREOF 的调度方法构成对比算法,记为 HRRR-Ind(HRRR-Independent Flow)。HRRR 和 HRRR-Ind 共在 24 个场景中进行了测试。拓扑规模为 40、60、80 节点,根据节点规模和仿真经验,流数量分别设置为 120-600、180-900、240-1 200、300-1 500。图 10 的测试结果表明,相比于 HRRR-Ind, HRRR 在全部场景中均有更高的调度成功率。而且随着流数量的增加, HRRR 的优势更明显。这是由于网络负载增加加剧了调度资源之间的竞争, PREOF 通过消除冗余为

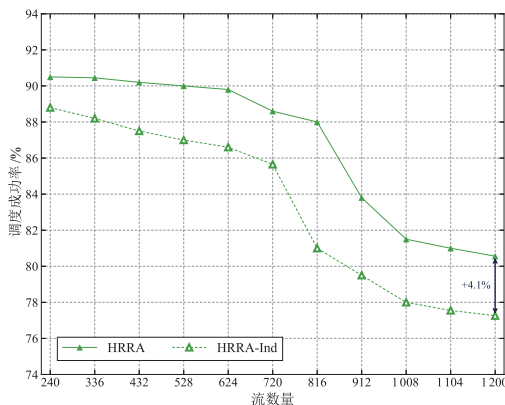
关键流释放更多可用资源,从而容纳更多关键流的确定性传输。相比于 HRR-Ind, HRR 在 80 节点 1 200 流的情况下调度成功率提高了 4.1%。



(a) 40节点拓扑
(a) 40-node topology



(b) 60节点拓扑
(b) 60-node topology



(c) 80节点拓扑
(c) 80-node topology

图 10 不同规模拓扑下 PREOF 对资源利用效率影响的结果
Figure 10 Impact of PREOF on resource utilization efficiency under different network scales

5 结束语

本文针对工业无线网络中的确定性传输需求,在 TSC 调度方法中考虑 PREOF 机制,提出了一个多路径路由与调度的联合优化问题。由于该问题具有 NP-hard 的复杂性,本文设计了一种基于 GNN 的分层强化学习算法 HRR。该算法能够为 IIoT 中关键流的确定性传输与高效资源利用提供支持。在 HRR 中,GNN 提升了对网络拓扑特征的表达能力,为高层策略的路由决策奠定基础。高层策略负责多路径路由选择,低层策略则进行无线资源块的分配,二者通过迭代优化实现最优调度。实验结果表明,HRR 在满足端到端确定性需求的同时,通过跨层联合优化显著提升了调度成功率和传输性能,并在 PREOF 支持下有效降低了冗余包转发,提升了资源利用效率。HRR 借助基于 GAT 的拓扑编码,对网络结构变化具有天然的适应能力。根据算法设计与运行机制,HRR 在动态拓扑场景中依然能够保持性能优势。针对节点增删与拓扑动态下的实验验证,将在未来工作中进一步补充。另外,将进一步探索 HRR 在分布式部署和可扩展性方面的改进,并优化其在实时运行场景下的性能表现。

参考文献

- [1] Sisinni E, Saifullah A, Han Song, et al. Industrial internet of things: Challenges, opportunities, and directions[J]. IEEE Transactions on Industrial Informatics, 2018, 14(11): 4724-4734.
- [2] Wang Quan, Jiang Jin. Comparative examination on architecture and protocol of industrial wireless sensor network standards[J]. IEEE Communications Surveys & Tutorials, 2016, 18(3): 2197-2219.
- [3] Lu Yuqian, Liu Chao, Wang K I K, et al. Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues[J]. Robotics and Computer-Integrated Manufacturing, 2020, 61: 101837.
- [4] Kotsiou V, Papadopoulos G Z, Chatzimisios P, et al. LDSF: Low-latency distributed scheduling function for industrial internet of things[J]. IEEE Internet of Things Journal, 2020, 7(9): 8688-8699.
- [5] Thubert P, Papadopoulos G, Buddenberg R. Reliable and available wireless architecture/framework[R/OL]. 2020-07-06. <https://imt-atlantique.hal.science/hal-02890366/document>.
- [6] Varga B, Farkas J, Malis A. RFC 9566: Deterministic networking (DetNet) packet replication, elimination, and ordering functions (PREOF) via MPLS over UDP/IP[R/OL]. 2024-06-26. <https://datatracker.ietf.org/doc/rfc9566/>.

- [7] De Armas J, Tuset P, Chang Tengfei, et al. Determinism through path diversity: Why packet replication makes sense[C]//Proceedings of the 2016 International Conference on Intelligent Networking and Collaborative Systems. Ostrava: IEEE, 2016: 150-154.
- [8] Bernardos C J, Papadopoulos G Z, Thubert P, et al. Reliable and available wireless (RAW) use cases: RFC 9450[S/OL]. <https://datatracker.ietf.org/doc/rfc9450/>.
- [9] Chang T, Vučinić M, Vilajosana X, et al. 6TiSCH minimal scheduling function (MSF): RFC 9033[S/OL]. <https://datatracker.ietf.org/doc/rfc9033/>.
- [10] Tanaka Y, Minet P, Vučinić M, et al. YSF: A 6TiSCH scheduling function minimizing latency of data gathering in IIoT[J]. IEEE Internet of Things Journal, 2022, 9(11): 8607-8615.
- [11] Mozaffari Ahrar E, Nassiri M, Theoleyre F. Multipath aware scheduling for high reliability and fault tolerance in low power industrial networks[J]. Journal of Network and Computer Applications, 2019, 142: 25-36.
- [12] 杨冬, 程宗荣, 田伟康, 等. 广义确定性标识网络[J]. 电子学报, 2024, 52(1): 1-18.
Yang Dong, Cheng Zongrong, Tian Weikang, et al. Generalized deterministic identification networks[J]. Acta Electronica Sinica, 2024, 52(1): 1-18. (in Chinese)
- [13] Palattella M R, Accettura N, Dohler M, et al. Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks[C]//Proceedings of the 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications. Sydney: IEEE, 2012: 327-332.
- [14] Lagos Jenschke T, Koutsiamanis R A, Papadopoulos G Z, et al. ODeSe: On-demand selection for multi-path RPL networks[J]. Ad Hoc Networks, 2021, 114: 102431.
- [15] Gaitán M G, Almeida L, Santos P M, et al. EDF scheduling and minimal-overlap shortest-path routing for real-time TSCH networks[C]//Proceedings of the Second Workshop on Next Generation Real-Time Embedded Systems (NGRES 2021). Budapest: Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021: 2.
- [16] Wijayasekara S K, Sasithong P, Hsieh H Y, et al. Optimization of packet transmission scheduling and node parent selection for 802.15.4e Time Slotted Channel Hopping (TSCH)[J]. ICT Express, 2024, 10(2): 442-450.
- [17] Chen Wuhui, Qiu Xiaoyu, Cai Ting, et al. Deep reinforcement learning for internet of things: A comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2021, 23(3): 1659-1692.
- [18] Li Jieling, Xiao Liang, Qi Xuchen, et al. Reinforcement learning based energy-efficient fast routing for FANETs[J]. IEEE Transactions on Communications, 2024, 72(11): 7063-7076.
- [19] Jurado-Lasso F F, Orfanidis C, Jurado J F, et al. HRL-TSCH: A hierarchical reinforcement learning-based TSCH scheduler for IIoT[J]. IEEE Transactions on Cognitive Communications and Networking, 2024, 10(6): 2102-2118.
- [20] Jiang Weiwei, Han Haoyu, Zhang Yang, et al. Graph neural networks for routing optimization: Challenges and opportunities[J]. Sustainability, 2024, 16(21): 9239.
- [21] Dong Guimin, Tang Mingyue, Wang Zhiyuan, et al. Graph neural networks in IoT: A survey[J]. ACM Transactions on Sensor Networks, 2023, 19(2): 1-50.
- [22] Huang Ru, Guan Wenfan, Zhai Guangtao, et al. Deep graph reinforcement learning based intelligent traffic routing control for software-defined wireless sensor networks[J]. Applied Sciences, 2022, 12(4): 1951.
- [23] Zhao Zhongyuan, Verma G, Rao C, et al. Link scheduling using graph neural networks[J]. IEEE Transactions on Wireless Communications, 2023, 22(6): 3997-4012.
- [24] Kim M, Park J, Kim J. Learning collaborative policies to solve NP-hard routing problems[PP/OL]. V1. arXiv(2021-10-26)[2025-10-10]. <https://arxiv.org/abs/2110.13987>.
- [25] Kodialam M, Nandagopal T. Characterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem[C]//Proceedings of the 9th annual international conference on mobile computing and networking. San Diego: ACM, 2003: 42-54.
- [26] Kool W, Van Hoof H, Welling M. Attention, learn to solve routing problems! [C/OL]//Proceedings of the 7th International Conference on Learning Representations, ICLR, 2019: <https://openreview.net/forum?id=ByxBFsRqYm>.
- [27] Bello I, Pham H, Le Q V, et al. Neural combinatorial optimization with reinforcement learning[C]//Proceedings of the 5th International Conference on Learning Representations. Toulon: OpenReview.net, 2017.
- [28] Chen Rongjun, Zhang Weiting, Wang Hongchao, et al. Enhancing energy efficiency in multipath routing for industrial Internet of Things[J]. IEEE Internet of Things Journal, 2025, 12(16): 33714-33730.
- [29] De Vos A, Brighente A, Conti M. Threat sensitive networking: On the security of IEEE 802.1CB and (un)effectiveness of existing security solutions[C]//Proceedings of the Computer Security. ESORICS 2022 International

Workshops. Copenhagen: Springer, 2022: 67-80.

- [30] IEEE Std 802.1CB-2017 IEEE standard for local and metropolitan area networks—frame replication and elimination for reliability[S].
- [31] Yang Sijin, Zhuang Lei, Zhang Jianhui, et al. A multipolicy deep reinforcement learning approach for multiobjective joint routing and scheduling in deterministic networks[J]. IEEE Internet of Things Journal, 2024, 11(10): 17402-17418.
- [32] Yang Liu, Wei Yifei, Yu F R, et al. Joint routing and scheduling optimization in time-sensitive networks using graph-convolutional-network-based deep reinforcement learning[J]. IEEE Internet of Things Journal, 2022, 9(23): 23981-23994.
- [33] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[PP/OL]. V2. arXiv(2017-08-28) [2025-11-12]. <https://arXiv.org/abs/1707.06347>.

- [34] Varga P, Peto J, Franko A, et al. 5G support for industrial IoT applications: Challenges, solutions, and research gaps[J]. Sensors, 2020, 20(3): 828.
- [35] Zubair Islam M, Shahzad, Ali R, et al. IoTactileSim: A virtual testbed for tactile industrial internet of things services[J]. Sensors, 2021, 21(24): 8363.
- [36] Valsesia D, Fracastoro G, Magli E. RAN-GNNs: Breaking the capacity limits of graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(8): 4610-4619.
- [37] Cai Chen, Wang Yusu. A note on over-smoothing for graph neural networks[PP/OL]. V1. arXiv(2020-06-23) [2025-10-11]. <https://arXiv.org/abs/2006.13318>.
- [38] Municio E, Daneels G, Vučinić M, et al. Simulating 6TiSCH networks[J]. Transactions on Emerging Telecommunications Technologies, 2019, 30(3): e3494.

作者简介



陈荣均 男,1997年9月生,湖北十堰人。现为北京交通大学博士研究生。主要研究方向为无线自组织网络、确定性网络传输技术、资源调度等。
E-mail: rongjun.chen@bjtu.edu.cn



乔凯 男,1997年7月生,山西吕梁人。现为北京交通大学博士研究生。主要研究方向为无线网络、资源分配、强化学习等。
E-mail: kaiqiao@bjtu.edu.cn



王洪超 男,1982年12月生,河北衡水人。现为北京交通大学电子信息工程学院副教授、硕士生导师。主要研究方向为新一代信息网络关键理论与技术、工业互联网、空天地信息网络技术等。
E-mail: hcwang@bjtu.edu.cn



田伟康 男,1999年3月生,重庆人。现为北京交通大学博士研究生。主要研究方向为未来确定性网络统一资源调度技术、异构网络人工智能优化算法等。
E-mail: weikangtian@bjtu.edu.cn



王钦定 男,1992年8月生,甘肃白银人。现为北京交通大学博士研究生。主要研究方向为未来网络体系架构、算力网络、算网融合等。中国电子学会会员编号:E190131238A。
E-mail: 22110022@bjtu.edu.cn



杨冬 男,1980年12月生,山西大同人。现为北京交通大学电子信息工程学院教授、博士生导师。主要研究方向为新一代信息网络关键理论与技术以及工业互联网、网络智能化技术等。中国电子学会会员编号:E190035787M。
E-mail: dyang@bjtu.edu.cn