

基于多队列CQF的UANET业务调度机制研究

缪霏阳, 白丁月, 张策, 张, 李大鹏, 赵海涛

(南京邮电大学通信与信息工程学院, 江苏 南京 210003)

摘要: 针对无人机自组网 (Unmanned Aerial Vehicle Ad hoc Network, UANET) 中, 介质访问控制 (Medium Access Control, MAC) 层在混合业务流队列调度中服务质量 (Quality of Service, QoS) 保障机制存在的问题, 提出了一种基于多队列循环队列转发 (Cyclic Queuing and Forwarding, CQF) 的业务调度机制 (Multi-Queue CQF-Based QoS Scheduling Mechanism, MQSM)。该机制首先引入多队列 CQF 模型及相应调度策略, 实现周期流和突发流的差异化传输需求, 完成混合流的调度任务。然后, 引入了一种融合 GhostNet 和 SRResNet 结构的时间资源调度算法 (Time Resource Scheduling Algorithm Integrating GhostNet and SRResNet, GSR), 以增强队列模型的调度性能并优化网络资源分配。最后, 通过仿真验证所提出的 GSR 算法在混合数据流场景中具有良好的调度能力, 能够合理地分配时间资源, 实现低时延通信和模型的快速部署。

关键词: 服务质量; 混合流; 循环队列转发; 突发流; 卷积神经网络

中图分类号: TP391.2

文献标志码: A

doi: 10.11959/j.issn.2096-3750.XXXX.

Research on Multi-Queue CQF-Based QoS Scheduling Mechanism for UANET

MIAO Feiyang, BAI Dingyue, ZHANG Ce, ZHANG Yang, LI Dapeng, ZHAO Haitao

College of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: To address the issues in MAC-layer Quality of Service (QoS) mechanisms for hybrid traffic queue scheduling in Unmanned Aerial Vehicle Ad Hoc Networks (UANETs), a Multi-Queue CQF-Based QoS Scheduling Mechanism (MQSM) was proposed. Firstly, multi-queue CQF and corresponding scheduling strategies were introduced to meet the differentiated transmission requirements of periodic and bursty traffic, thus accomplishing the scheduling of hybrid flows. Then, a Time Resource Scheduling Algorithm Integrating GhostNet and SRResNet (GSR) was proposed to enhance the scheduling performance of the queue model and optimize network resource utilization. Finally, simulation results verify that the GSR algorithm demonstrates excellent scheduling capability in hybrid traffic scenarios, effectively allocating time resources to achieve low-latency communication and rapid model deployment.

Key words: quality of service, mixed flow, cyclic queuing and forwarding, burst flow, convolutional neural network

0 引言

随着物联网^[1]、5G/6G 通信技术^[2]、人工智能^[3]以及无人机技术^[4-5]的快速发展, 无线通信网络的应

收稿日期: XXXX-XX-XX; 修回日期: XXXX-XX-XX

通信作者: 缪霏阳, feiyangmiao@163.com

基金项目: 国家自然科学基金项目(No.U24B201873)

Foundation Items: The National Natural Science Foundation of China (No.U24B201873)

用范围持续拓展, 实际价值显著提升。传统的通信网络在诸如自然灾害^[6]、军事作战^[7]等极端或特殊环境下, 往往存在覆盖不足、通信设备易损毁等问题, 难以满足快速、高效、可靠的通信需求。而无人机自组网^[8]凭借其高机动性^[9]、部署灵活性^[10]以

及通信覆盖范围广泛^[11]等特点, 成为构建临时应急

通信网络的理想平台^[12]。为确保各类业务在时延、可靠性和传输速率等关键性能指标上满足更高标准

准的应用需求, 构建更高效、可靠的资源分配机制已成为UANET发展的核心任务之一。在此背景下, MAC层作为连接物理层与网络层的关键桥梁, 其重要性愈发凸显^[13-16]。

在MAC层中, 队列调度机制作为管理无线信道接入与资源分配的重要手段, 在QoS保障中发挥着核心作用。然而, 面对多业务并发接入所引发的信道冲突与网络拥塞, 传统调度机制多基于尽力而为(Best-Effort, BE)策略^[17], 未对不同业务类型进行有效区分与优先级管理。因此, 面向多业务场景下的MAC层QoS调度机制逐步聚焦于更为精细的调度算法设计, 尤其是结合优先级控制与时隙划分策略的调度算法。为实现对多类业务时延与可靠性的精准保障, 调度机制的模型设计也成为研究的重要内容。

近年来, 时间敏感网络^[18](Time-sensitive Networking, TSN)提出了多项面向实时通信的关键技术, 其中时间感知调度(Time-Aware Shaper, TAS)模型^[19-21]和循环队列转发模型^[22-24]最具代表性。TAS通过为不同优先级业务配置精确的时间窗口, 使高优先级流量在特定时段内独占传输资源, 从而隔离了低优先级业务的资源竞争与干扰, 提高了时延的可控性; 而CQF则通过周期性调度缓存队列中的数据包, 配合双缓冲结构, 实现了端到端的确定性转发路径, 显著增强了网络的传输稳定性和可预测性。由于UANET具有高动态、强突发等特性, 传统无线调度机制难以提供TSN所要求的确定性服务保障。因此, 部分学者开始将TSN技术中的调度算法和队列模型引入无线网络MAC层队列调度设计, 在多业务并发环境下实现高效的QoS保障。

在模型的调度算法方面, 文献[25]提出基于统计复用的时变流调度方案, 结合启发式与强化学习提高TSN链路利用率, 但时延确定性不足且实现复杂; 文献[26]设计结合TAS与CQF的混合调度机制, 通过时隙调整和奇偶映射提升调度成功率与资源利用率, 但依赖高精度时钟, 在动态环境中受限; 文献[27]提出深度强化学习辅助的DTF算法, 联合分配时频资源, 有效提升容量利用率学习能力; 文献[28]则利用时隙感知与模拟退火思想设计

混合调度算法, 将非时间敏感流插入剩余时隙, 避免局部最优并实现全局优化。而在队列模型方面, 文献[29]提出基于CQF的调度方法, 结合参数优化与整数线性规划, 在高负载下提升周期流性能与资源利用率, 但受限于固定时隙长度, 对突发流量适应性差且调优复杂; 文献[30]提出基于优先级的CQF队列, 通过横向调整长度与纵向增加队列数实现更细粒度时延控制, 增强突发流量适应性, 但未考虑流到达时间, 高动态场景下效果有限且计算开销大。

针对上述现有模型与调度算法所存在的不足, 本文设计了MQSM机制, 通过多队列CQF模型和GSR算法实现高效的业务流调度能力。结果表明, GSR算法在混合数据流场景中具有良好的调度能力, 能够合理分配时间资源, 提升了网络的整体性能。

1 系统模型与问题分析

1.1 系统模型

图1展示了UANET中的多业务流传输场景, 场景由 R 个无人机节点组成, 包括收发两端的无人机终端节点 N_i , 以及数据流转发区域中的无人机中继节点 S_j 。为构建网络拓扑抽象模型, 采用有向图结构 $G=(V, L)$, 其中 $V=N \cup S$ 由终端节点集 $N=\{N_0, N_1, \dots, N_I\}$ 与中继节点集 $S=\{S_0, S_1, \dots, S_{R-I}\}$ 共同构成。网络中的链路集合记为 L , 每条链路连接两个邻节点 v_x 和 v_y , 并且 $L=\{(v_x, v_y) | v_x, v_y \in V, v_x \neq v_y\}$ 。发送终端节点可通过接入UANET, 将多种类型的业务数据接入网络, 即图中的尽力传输流、周期流和突发流, 并通过多次转发发送至接收终端节点, 业务流集合记为 F , $|F|$ 代表业务流总数, 其中每条业务流用 $f_i \in F$ 进行标识。CQF通过多队列隔离与全局同步的循环指针机制, 在TDMA时隙框架内实现了高优先级业务的零等待调度及中低优先级业务的有序传输。无人机节点在接收到各类业务流后, 会在其分配的TDMA发送时隙 T_i 内, 通过MAC层的队列转发机制, 对各类业务数据进行调度与发送, 从而在资源受限的环境下实现多业务协同与通信质量保障, 提升UANET在应急场景中的通信可靠性与服务能力。

本文的主要符号定义如表1所示。

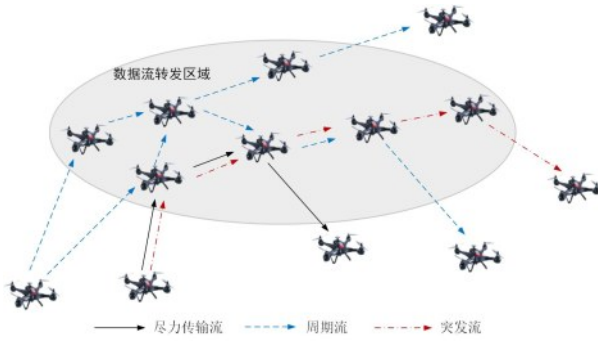


图1 UANET业务流传输场景

(1) 队列调度模型

CQF模型如图2所示，传统的CQF采用二队列模型，其中队列 Q_1 和 Q_2 共同负责业务流的接收与发送。每个队列均配备接收侧门控和发送侧门控，用于控制数据的进入与发送，门控的开关状态由门控列表统一调度。假设时隙 T_0 和时隙 T_1 组成了一个调度周期，则CQF模型需满足以下要求：

1) 数据流的发送与接收采用交替方式进行。如图2所示，在偶数时隙 T_0 内，队列 Q_1 打开发送

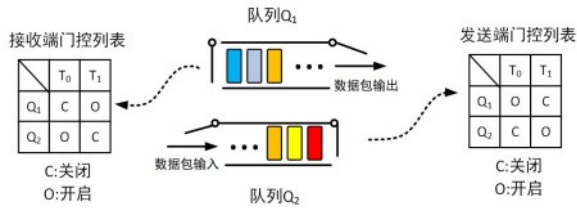


图2 传统CQF

门控但关闭接收门控，仅进行数据发送；队列 Q_2 相反，仅进行数据接收。在下一奇数时隙 T_1 ，队列状态互换。当一个周期含有多个连续时隙时，队列仍按奇偶时隙规律交替切换收发模式，以维持稳定的传输效率。

2) 为确保相邻节点间的数据传输延迟确定且受控，业务流在相邻两跳节点间的传输时长必须小于一个时隙。具体而言，CQF要求上游节点需在当前时隙 T_i 内发送完上一时隙缓存的数据包，而下游节点则利用队列缓冲机制，在相同时隙内完成数据的无丢包接收。如图3所示，节点 S_1 在时隙 T_0 通过 Q_1 队列发送数据，与此同时，下游节点同步接收，根据CQF门控策略，接收到的数据必须等到下一时隙 T_1 才能被转发至下一跳节点，以此通过严格的时隙同步确保业务流的确定性传输。

3) 业务流具有端到端确定性时延与抖动边界。在CQF机制中，业务流的端到端延迟取决于网络中继节点的跳数 H 与时隙长度 T 。当源节点在时隙 T_i 的开始时刻发送数据包时，数据包在经由各层节点时，首先被写入该节点所属的接收缓存队列，待达到时隙持续时长阈值后，方可由队列调度器执行转发指令，进行转发。因此，相邻节点之间业务流的传输延迟小于单个时隙长度 T 。由于最后一跳节点到接收终端仅需完成数据接收，因此当最后一跳节点向接收端发送数据时延恰好为时隙长度 T 时，

表1 本文主要符号定义

符号	定义	符号	定义
N_i	无人终端节点	K	传输时隙总数
S_j	无人机中继节点	$\mathfrak{U}_i$	单个周期内发送的数据包总数
V	网络节点集合	U_i	时隙资源利用率
L	链路集合	$T_i \cdot C$	时隙 T_i 的容量
F	业务流集合	L_p	调度周期
$ F $	业务流总数	T	时隙长度
f_i	业务流	$f_i.tx$	业务流 f_i 的发送时刻
f_i^j	第 i 个业务流的第 j 个数据包	$period_i$	业务流 f_i 的发送周期
T_i	第 i 个时隙	η	时钟同步精度
Q_i	CQF中的第 i 个队列	B	链路带宽
H	节点跳数	$pack_m^n$	第 m 个业务流的第 n 个数据包
T_{Dmax}	端到端最大时延	num_m	业务流在一个周期的数据包数量
T_{Dmin}	端到端最小时延	$Node_i$	传输中经过的节点总数
T_{Jmax}	抖动最大界限	$pack_m^n \cdot T_D$	数据包的端到端时延
ϕ_r^i	第 i 个业务流选择的偏移值	$delay_i^{[v_i, v_j]}$	业务流在邻节点间的传输时延
ϕ_r	队列偏移值	ε	时钟同步偏差

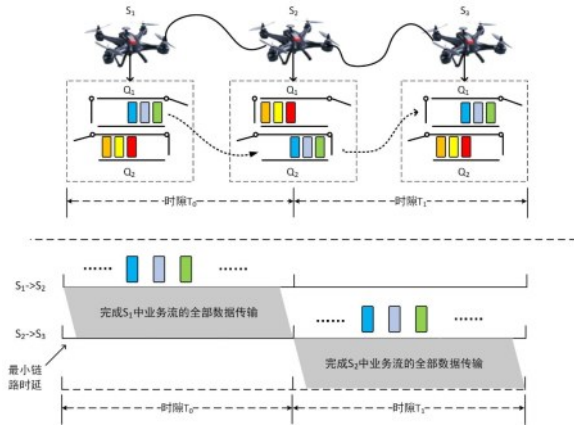


图3 业务流转发规则

即满足了CQF单时隙转发的规定，同时也确定了业务流端到端的最大延迟界限，即：

$$T_{D\max} = (H + 1) \times T \quad (1)$$

当业务流在时隙末尾由源终端发送时，由于数据包到达第一跳节点的延迟近似为0，而最后一跳节点传至接收端的时延同样近似为0，从而系统可实现端到端传输延迟的最小化。然而，受CQF时隙交替规则限制，相邻节点之间的业务流传输延迟不会低于单个时隙长度。最终，端到端最小时延公式为：

$$T_{D\min} = (H - 1) \times T \quad (2)$$

端到端时延抖动衡量了数据流在网络内传输延迟的波动范围，即不同数据包传输时延之间的差异。因此，抖动的最大界限即为数据流最大与最小端到端时延之差，表达式如下：

$$T_{J\max} = T_{D\max} - T_{D\min} = 2 \times T \quad (3)$$

(2) 业务流约束条件

在CQF模型中，一个调度周期常包含多个业务流传输时隙，为避免单时隙容量不足引起丢包，合理规划业务流的时隙映射对系统性能至关重要。

图4示例展示了CQF模型下的业务流调度约束优化问题。简化网络由三个终端节点 N_1 、 N_2 和 N_3 ，以及一个中继节点 S_1 组成。业务流1和2均为时延敏感的周期流，业务流1数据发送周期为1600 μ s，每次发送编号分别为1.1和1.2的两个数据包；业务流2的数据发送周期为3200 μ s，每周期发编号为2.1的一个数据包。为保证周期性业务流的有效调度，整个网络的调度周期设定为3200 μ s，并划分为8个传输时隙且每个时隙最多容纳2个数据包。图4(a)中，由于多流同时占用同一时隙，节点转发阶段

出现超载，导致数据包2.1被丢弃。图4(b)展示了改进的调度方案：调整 N_2 发送数据包2.1的时机，将其延迟至时隙 T_1 ，使得无人机节点 S_1 在时隙进行转发，从而避免冲突、均衡负载并提升系统可靠性与调度效率。

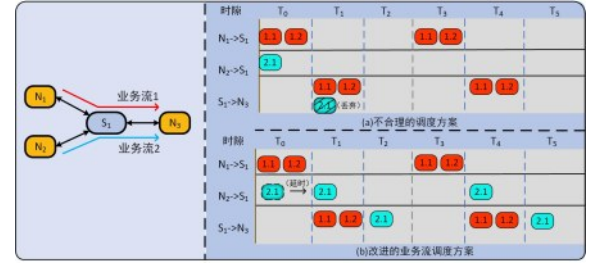


图4 业务流调度约束优化问题示意图

通过上述示例可见，在CQF模型中，为了实现业务流的合理调度，避免出现丢包现象，需要建立相应的约束优化条件。因此，首先需要对业务流的各项核心特性进行提取与精确定义，这些特性包括业务流 id 、业务流周期 $period$ 、源地址 src 、目的地址 dst 、端到端时延需求 ζ 、抖动需求 J 、数据包 $pack$ 、数据包数量 num 以及数据包大小 $size$ ，因此业务流可表示为：

$$f = \{ id, period, src, dst, \zeta, J, pack, num, size \} \quad (4)$$

为确保调度机制对网络资源的充分适配，需明确定义网络的资源属性和函数。首先，定义指示函数 $\omega_{\xi}(f_i)$ 用以判断业务流 f_i 在时隙 T_{ξ} 的传输可行性，若业务流 f_i 在时隙 T_{ξ} 内完成传输，则 $\omega_{\xi}(f_i)$ 赋值为1，否则置0。

接着，定义价值函数 $V_{\xi}(f, K, F)$ 以评估业务流的调度收益，其中 K 为调度周期内传输时隙的总数，并且 $V_{\xi}(f, K, F)$ 由业务流调度成功的数量与时隙资源平均利用率加权构成，因此定义 $A(\cdot)$ 函数用于计量成功调度的数据流数量，而 $B(\cdot)$ 函数则用于计算时隙的平均资源利用率。基于此，价值函数 $V_{\xi}(f, K, F)$ 可进一步表示为：

$$V_{\xi}(f, K, F) = \alpha_1 \times A(f, K) + \beta_1 \times B(K, F) \quad (5)$$

其中， $0 < \alpha_1, \beta_1 < 1$ 为权重因子，用于调节业务流成功调度数量与时隙资源利用率两项优化目标之间的比重，在本文中 β_1 设置为0.1，即系统模型重点关注成功调度的业务流数量。当一条业务流的所有数据包均被成功调度时，函数 $A(f, K)$ 的取值为1；反之则为0。因此，函数 $A(f, K)$ 的具体表达式可

描述如下：

$$A(f_i, K) = \begin{cases} 1, & \text{if } \sum_{j=1}^{\bar{U}_i} \sum_{t=0}^{K-1} \omega_t(f_i^j) = \bar{U}_i \\ 0, & \text{if otherwise} \end{cases} \quad (6)$$

其中， $\bar{U}_i$ 代表业务流 f_i 在一个调度周期内传输的数据包总数， f_i^j 表示第 i 个业务流的第 j 个数据包。 $B(K, F)$ 用于评估平均时隙资源利用率。每个时隙 T_i 的资源利用率 U_i 可以描述为：

$$U_i = \frac{\sum_{m=1}^{|F|} \sum_{n=1}^{\bar{U}_m} \omega_t(f_m^n) \times \text{size}_m}{T_i \cdot C} \quad (7)$$

其中，时隙 T_i 的容量定义为 $T_i \cdot C$ ，其数值受时隙时长与链路带宽的影响。因此，在一个调度周期中，各时隙的平均资源使用效率可表示为：

$$B(K, F) = \frac{1}{K} \sum_{t=0}^{K-1} U_t \quad (8)$$

除此之外，考虑业务流周期性特点和循环队列转发特征，UANET中的调度周期可设置为各业务流周期的最小公倍数。如此一来，仅需在首个调度周期内确定业务流与时隙的对应关系，后续周期的安排便可直接复用。因此，调度周期的约束条件可以描述为：

$$L_p = \text{LCM}(F.\text{period}) \quad (9)$$

在CQF模型中，最小公倍数函数 $\text{LCM}(\cdot)$ 用于计算调度周期。调度周期的长度必须能被单个时隙长度整除，各业务流的周期也应满足能够被时隙长度整除的条件。因此，周期约束可具体表示为：

$$L_p \% T = 0, \text{period}_i \% T = 0 \quad (10)$$

由公式(10)推导可得，最大传输时隙长度应为所有业务流周期的最大公约数，因此在设计时隙时应满足如下约束条件：

$$T = \text{GCD}(F.\text{period}) \quad (11)$$

其中 $\text{GCD}(\cdot)$ 用于计算最大公约数。为避免不同周期的流量在同一接口上产生资源冲突，业务流的发送时刻应满足不超过其自身的周期长度，即：

$$0 \leq f_i.\text{tx} \leq \text{period}_i \quad (12)$$

在分组交换网络架构中，相邻中继节点间报文传输总时延由四类基础时延分量构成：接口串行化时延、物理链路传播时延、节点处理时延及队列缓冲时延。基于确定性传输需求，四类时延总和需满足时隙上限约束。此外，为贴合实际场景，还需要考虑到时钟同步精度 η ，则公式如下所示：

$$T \geq \frac{\max\{\text{size}_i\}}{B} + d_i^{\text{prop}} + d_i^{\text{proc}} + d_i^{\text{que}} + \eta, i = 1, 2, \dots, |F| \quad (13)$$

其中， B 代表链路带宽。为了保障业务流传输的高度可靠性需求，CQF模型要求在每个时隙内传输的业务流总大小不超过时隙的最大容量，即：

$$\begin{cases} 0 \leq \sum_{m=1}^{|F|} \sum_{n=1}^{\text{num}_m} \text{size}_m \times \omega_t(\text{pack}_m^n) \leq T_i C \\ K = L_p/T, t = 1, 2, \dots, K \end{cases} \quad (14)$$

其中，第 m 条业务流的第 n 个数据包记为 pack_m^n ，该业务流在一个调度周期内共包含 num_m 个数据包。当数据包 pack_m^n 在第 t 个时隙成功传输时，指示函数 $\omega_t(\text{pack}_m^n) = 1$ ，反之则为0；CQF模型在满足网络资源约束的同时，还需保障端到端业务流的确定性传输，其核心指标涵盖时延极值限定、延迟波动范围约束及无丢包保证。时延不得超过设定的最大允许值、抖动（即数据包实际时延与平均时延的偏差）应处于预设范围，以及传输过程无丢包，保证网络的高可靠性。因此，上述业务流的传输约束可具体表示为：

$$\begin{cases} (Node_i + 1) \times T \leq \zeta_i \\ \left| \text{pack}_i^j \cdot T_D - \frac{1}{\text{num}_i} \sum_{s=1}^{\text{num}_i} \text{pack}_i^s \cdot T_D \right| \leq J_i \\ \left| \sum_{s=1}^{\text{num}_i} \sum_{t=1}^K \omega_t(\text{pack}_i^s) - \text{num}_i \right| \approx 0 \\ i = 1, 2, \dots, |F| \end{cases} \quad (15)$$

其中， $Node_i$ 表示第 i 条业务流在传输中所经过的节点数量， $\text{pack}_i^j \cdot T_D$ 表示业务流数据包的端到端时延。由于每个数据包只能被分配到一个唯一的传输时隙，因此需要引入数据包与传输时隙之间的映射约束条件：

$$\sum_{t=0}^{K-1} \omega_t(f_i^j) \leq 1, \omega_t(f_i^j) \geq 0, \omega_t(f_i^j) \in Z \quad (16)$$

除周期流调度外，传统CQF还面临突发流调度问题。图5展示了其主要问题，其中 ϕ_i^t 为业务流 f_i 选择的接收队列偏移值，默认取值为1。传统CQF要求终端节点 N_i 发送的所有业务流必须在一个时间间隔内被中继节点 S_i 接收。然而，由于传输时延及发送时间的不确定性，突发流可能落入死区时间 T_d 内，导致帧无法被完整接受并引发丢包。死区时间的产生是由于突发流的到达时间晚于接收队列所附带的Rx门控的关闭时间。此外，在混合业

业务的传输场景下, 流量可能会在同一队列中聚集, 超出单个时间间隔的承载能力, 从而影响流的可靠传输。

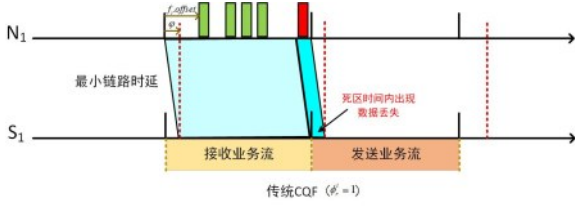


图5 传统CQF突发流调度示例

由此可见, 传统CQF在突发流调度方面存在不足。为了应对死区时间导致的数据包丢失问题, 有必要对队列模型中的偏移值设置进行约束。首先, 对于在时隙 T_i 内传输的突发业务流 f_i , 需要满足:

$$\begin{cases} f_i \cdot \text{offset} + \text{delay}_{i[v_x, v_y]} + \varepsilon \leq T \\ \text{delay}_{i[v_x, v_y]} > 0, \delta \approx 0 \end{cases} \quad (17)$$

其中, $f_i \cdot \text{offset}$ 为队列门控开启时刻与业务流 f_i 实际发送时刻之间的时钟偏移量, $\text{delay}_{i[v_x, v_y]}$ 为业务流 f_i 在两个邻节点 v_x 和 v_y 之间的传输时延, ε 为时钟同步偏差。对于在时隙 T_i 结束时刻进行发送的业务流, 则有 $f_i \cdot \text{offset} = T$, 即业务流的时间偏移量为 T , 则可以得到如下公式:

$$\begin{aligned} f_i \cdot \text{offset} + \text{delay}_{i[v_x, v_y]} + \varepsilon \\ = T + \text{delay}_{i[v_x, v_y]} + \varepsilon > T \end{aligned} \quad (18)$$

由公式 (17) 与公式 (18) 之间存在的矛盾, 可以得知, 当队列偏移值为 1, 在传输时隙内一定会出现死区时间 T_d , 从而造成数据包丢失。因此队列偏移值需满足如下公式:

$$T + \text{delay}_{i[v_x, v_y]} + \varepsilon < \phi_r^i \cdot T \quad (19)$$

1.2 问题分析

为了解决MAC层下的业务流调度问题, 需要针对每条业务流同步规划其队列资源与传输时隙资源, 从而使得网络中调度成功的业务流数量与传输时隙的整体资源利用率最大化。此外, 为避免业务流在调度过程中出现死区时间导致的丢包问题, 还需严格满足公式 (19) 中所规定的偏移值约束条件。因此本文的MAC层队列调度问题的数学模型如下所示:

$$\text{Max} \frac{1}{|F|} \sum_{i=1}^{|F|} \sum_{k=0}^{K-1} V_{\varepsilon}(f_i, K, F) \times \omega_{\varepsilon}(f_i) \quad (20)$$

$$\text{s.t. } \phi_r^i > 1 + \frac{\text{delay}_{i[v_x, v_y]}}{T} \quad (21)$$

因此, 本文通过引入多队列CQF并采取相应的业务流调度策略, 满足公式 (21) 的偏移值约束条件, 从而解决了死区时间中的业务流丢失情况。此外, 公式 (20) 的优化目标本质上是对所有业务流在不同时隙和队列资源上的调度效果进行加权求和, 进而最大化整体调度收益, 并由指示函数 $\omega_{\varepsilon}(f_i)$ 的定义和公式 (16) 可知, 该优化问题等价于0-1背包问题, 即NP难问题。本文针对其求解复杂性, 设计了一种基于深度强化学习的时间资源调度算法, 通过多种设计引导模型逐步学习并逼近最优的业务流调度决策。

2 多队列CQF及业务流调度策略

为解决突发流调度问题, 实现混合业务流的协同传输, 本文基于文献[30]和文献[31]的研究, 搭建了多队列CQF, 该结构在每个无人机节点的队列调度模型上均配置了 M 个队列, 其中 $M \geq 3$ 。每个队列配备门控机制, 按预设节奏周期性地开启或关闭。每个时隙内, 仅允许一个队列处于启用状态进行数据收发, 其他队列门控保持关闭, 从而在时隙级实现多路复用, 避免流间冲突, 提高带宽利用率, 支撑周期流与突发流的确定性共网传输。

若某队列用于数据发送, 其状态记作 $Q_i = 0, 0 < i < M - 1$, 若用于接收, 则其队列状态则设为 $Q_j = 1, i \neq j$, 具体而言, 在每个时隙 T_o 内, 队列的门控状态依据设定的调度策略进行切换, 其定义为:

$$\begin{cases} Q_{\sigma \% M} = 0, \sigma \geq 0, \sigma \% M \leq M - 1 \\ Q_j = 1, j \neq \sigma \% M, j = 0, 1, \dots, M - 1 \end{cases} \quad (22)$$

基于上述队列控制策略, 进一步制定了业务流在相邻两跳节点之间的接收逻辑。该逻辑依赖于队列的偏移量设定, 队列 Q_j 的偏移值 ϕ_j 表示在当前时隙中, 从对应的发送队列 $Q_{\sigma \% M}$ 到对应接收队列 Q_j 的逻辑单跳传输时延。发送队列 $Q_{\sigma \% M}$ 的偏移值定义为 ϕ_{base} 。对于接收业务流的队列, 其偏移值以发送队列的 ϕ_{base} 为参考点, 并在此基础上依次递增 1, 形成循环偏移模式。

结合上述队列偏移量设置, 根据各时隙中的收发状态, 通过数学表达式刻画偏移量, 即:

$$\begin{cases} \phi_j = \phi_{\text{base}} + (j - \sigma \% M + M); j - \sigma \% M < 0 \\ \phi_j = \phi_{\text{base}} + (j - \sigma \% M); j - \sigma \% M \geq 0 \\ j = 0, 1, \dots, M-1, \sigma \geq 0 \end{cases} \quad (23)$$

如图6所示，为了避免突发业务流在死区时间 T_d 内发送时出现丢包现象，突发流在选择下一跳无人机节点的接收队列时，需要优先选择偏移值不小于2的队列，即满足 $\phi_r \geq 2$ ，从而确保业务流能够被成功接收，即有以下公式成立：

$$\phi_r \times T \geq f_i \cdot \text{offset} + \text{delay}_i^{[v_r, v_s]} + \varepsilon \quad (24)$$

从而满足了公式(21)所建立的优化条件。多队列CQF通过延长接收业务流的时隙，使得在死区时间内发送的突发业务流依然能够被下游节点中偏移值较大的队列接收，有效提升其传输过程中的可靠性。

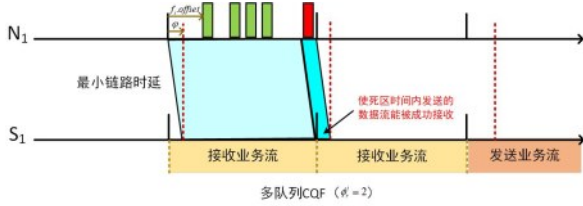


图6 多队列CQF突发流调度示例

为进一步提升混合业务流的转发效率和时隙资源利用率 U_r ，制定了以下业务流调度策略：

(1) 对来自上游节点的突发流，为避免丢包，必须为其分配具有特定偏移值的接收队列，即选择偏移值 $\phi_r \geq 2$ 的队列作为下一跳接收队列。具体的偏移值需根据突发流的流量特征、时延要求和抗抖动能力进行合理配置。

(2) 突发流在首跳被成功接收后，其后续的转发可依托循环队列机制完成。引入快速转发机制，在后续各跳选择偏移值为 $\phi_r = 1$ 的队列进行接收，从而确保每两跳节点之间的传输延迟不超过一个时隙长度，提升流量转发效率。

(3) 为合理分配队列资源并避免周期流占用突发流关键资源，根据各业务流的时延和抖动需求，为周期流配置适当的偏移值，使其所能选择的队列偏移范围为 $\phi_r \geq 1$ 。

(4) 对于网络中时延需求较低的业务流（即尽力而为流），或者当前CQF接收队列的容量已满时，可以将其存入尽力传输（Best Efforts, BE）队列^[30]。BE队列不设定队列偏移值，而是采用机会性发送策略，若当前时隙仍有剩余带宽，调度BE

队列进行发送，从而充分利用空闲时隙并提高整体利用率。

图7举例展示了多队列CQF的工作机制。在时隙 T_0 内，终端 N_1 通过UANET中的四个无人机向 N_2 同时传输周期性业务流，系统设定队列数为 $M=3$ 。为实现突发流的可靠传输，在 T_0 时隙内突发流选择偏移值 $\phi_r = 2$ 的队列，即根据公式(23)选择队列 Q_2 用于接收业务流，从而满足公式(24)。在经过两个时隙后，在 T_2 时隙内突发流需要选择 $\phi_r = 1$ 的队列，以减少后续跳数造成的累计延迟。周期流则可根据其对延迟的容忍度，选择偏移值 $\phi_r = 1$ 的队列进行传输，既满足了服务质量要求，也避免了对高优先级流量的干扰。如果周期流对时延要求不高，还可通过分配更大的偏移值，释放出更多高优先级业务所需的资源队列。

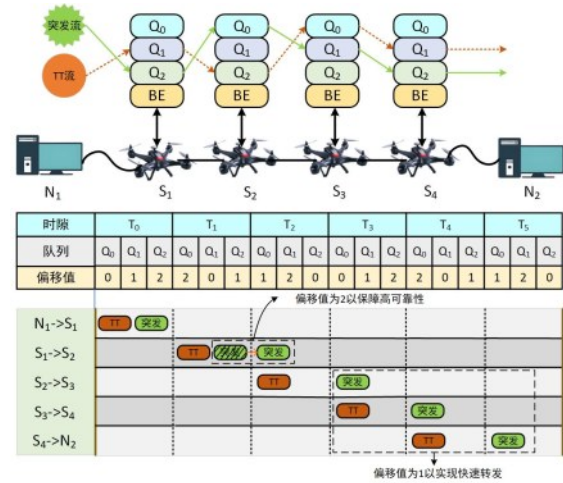


图7 多队列CQF工作机制

3 基于深度强化学习的时间资源调度算法设计

为实现公式(20)的优化目标，本节提出了一种基于深度强化学习的时间资源调度算法，即GSR算法。如图8所示，GSR算法首先构建二维时间-资源矩阵，并与动作矩阵融合形成三通道输入，通过结合本文设计的Ghost卷积和SRResNet的融合卷积网络进行特征提取并输出相应的Q值。随后，专家引导模块对选择的动作进行评估和调整，动作执行后获取环境反馈的状态和奖励信息，并通过经验回放机制持续优化网络参数，从而提升资源调度决策效率。通过上述方法，GSR算法有效增强了网络对混合业务流的支持，提升了成功调度的业务流

数量。

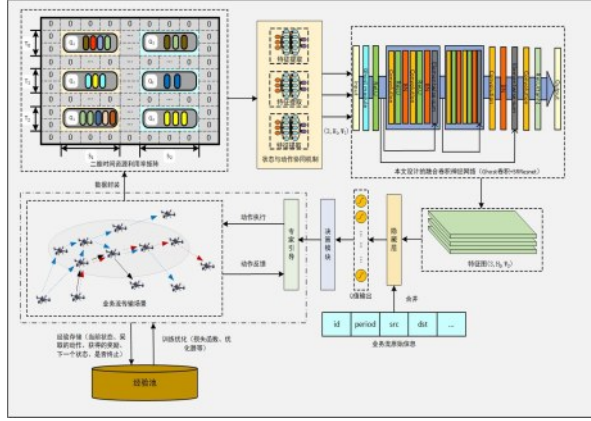


图8 GSR算法框架

3.1 二维时间-资源利用率矩阵

在多队列CQF中,时隙和队列资源间并非完全独立。如图9所示,二维时间-资源矩阵以行表示时隙,列表示每个节点的循环队列,能够动态记录不同时间下各队列的资源占用情况,并作为智能体的状态输入通道,随业务流的到达实时更新资源使用状态,实现状态空间的自适应迁移。此外,每列还可扩展引入频率、计算能力、链路占用率等附加信息,从而拓展至更广泛的调度场景。为进一步提升智能体对关键特征的提取能力,将每个矩阵元素扩展为一个 $u \times m$ 的向量,以映射不同时间-队列的资源利用关系。对于无直接关联的队列,矩阵采用零填充策略,以构建完整的 $p \times p$ 状态矩阵。

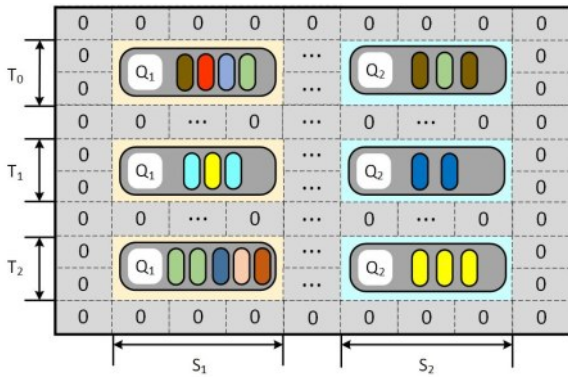


图9 二维时间-资源利用率矩阵

3.2 状态与动作协同机制

为增强智能体决策效能,采用多模态数据融合策略,将状态信息与动作信息沿通道轴拼接,通过特征提取,实现参数的协同迭代优化。状态部分由表征时间-资源利用率的二维矩阵 $S \in \mathbb{R}^{T \times Q}$ 构成。

动作部分则以决策行为张量的形式表示,矩阵包含两个子通道:其中,第一个子通道 $A_1 \in \mathbb{R}^{T \times Q}$ 用于指示各时隙中队列的操作行为(如发送或接收);第二个子通道 $A_2 \in \mathbb{R}^{T \times Q}$ 用于指示每个业务流在有效传输时隙中的行为。其中,合法动作标记为1,非法动作标记为0。通过这种融合方式,得到三通道状态-动作矩阵:

$$X = \text{concat}(S, A_1, A_2) \in \mathbb{R}^{T \times Q \times 3} \quad (25)$$

三条独立的信息通道能够为智能体提供更丰富的状态-动作特征表达,提升策略学习能力。

3.3 卷积神经网络设计

在深度学习中,高效的特征提取,较低的计算成本和较高的收敛速度是提升模型性能的关键。针对传统卷积神经网络计算量大、参数冗余以及收敛速度较慢的缺点,本文设计了一种克服上述问题的卷积神经网络,即融合GhostNet和SRResNet结构的时隙资源调度算法。相较于其他常用的轻量化网络,如MobileNetV2/V3^[32-33]、ShuffleNetV2^[34]、EfficientNet系列^[35], GhostNet^[36-37]的核心优势在于其用更少的参数与线性运算扩展生成更多的有效通道,从而使模型复杂度更低,同时实现更高精度与更低延迟。另一方面, SRResNet^[38-39]中的残差块与跳跃连接通过恒等映射与残差学习缓解梯度消失与信息衰减问题,使得更深的卷积层能学习到更丰富的层级特征。通过结合Ghost卷积的高效性和SRResNet的强大特征提取能力, GSR算法能在降低计算成本的同时,提高模型的关键特征表达,并加快模型的收敛速度。

(1) GhostNet

Ghost卷积采用了分阶段卷积策略。如图10所示,该模块通过精简特征通道数量,并结合深度可分离卷积技术,在保证特征表达能力的同时,降低了计算量。

假设给定的输入数据为 $X \in \mathbb{R}^{c \times h \times w}$,其中 c 是输入通道的数量, h 和 w 分别是输入数据的高和宽,即输入特征的维度为 $c \times h \times w$,从而使用常规卷积经过 n 组的 $m \times m$ 卷积核处理后,输出的特征维度为 $n \times h' \times w'$,其参数量和计算量可以表示为:

$$P = ncmm \quad (26)$$

$$C = nh'w'cmm \quad (27)$$

其中,常规卷积的参数量 P 和计算量 C 是衡量模型复杂度的关键指标。

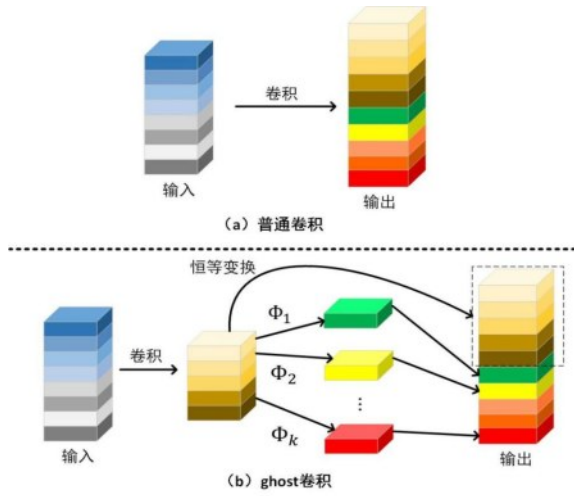


图10 普通卷积与ghost卷积对比

在 Ghost 卷积架构中，给定相同的输入输出维度约束，首先通过标准卷积生成维度为 $\frac{n}{s} \times h' \times w'$ 的本征特征输出。接着通过 $s - 1$ 次线性变换生成维度为 $(s - 1) \times \frac{n}{s} \times h' \times w'$ 的 Ghost 特征输出。最后，将本征特征与 Ghost 特征拼接，得到与常规卷积相同的输出。其参数量和计算量分别为：

$$P_G = \frac{n}{s} cmm + (s - 1) \frac{n}{s} tt \quad (28)$$

$$C_G = \frac{n}{s} h'w'cmm + (s - 1) \frac{n}{s} h'w'tt \quad (29)$$

其中， P_G 表示 Ghost 卷积的参数量， C_G 表示 Ghost 卷积的计算量， s 为远小于通道数 c 的线性变换次数， $t \times t$ 为线性变换的平均核大小，且上文的卷积核 $m \times m$ 与 $t \times t$ 大小相似。根据上述公式 (26) - (29) 可以得出如下公式：

$$R_p = \frac{P}{P_G} \approx s \quad (30)$$

$$R_c = \frac{C}{C_G} \approx s \quad (31)$$

其中， R_p 表示两者参数量的比值， R_c 则表示计算量的比值。相较于常规卷积，线性变换的计算开销可以忽略不计。因此，在生成相同尺寸的特征输出时，Ghost 卷积能够明显降低计算复杂度。（2）

SRResnet

残差学习（Residual Learning）的核心思想是在网络中引入捷径（shortcut）连接。如图 11 所示，与传统神经网络的每一层直接学习 $y = f(x)$ 的映射不同，残差学习采用 $y = f(x) + x$ 的映射关系，即将输入直接加至输出。这一捷径连接能够有效缓解

梯度消失现象，使更深的网络结构得以训练并提升整体性能。此外如果对于某个输入 x ，映射函数 $f(x)$ 近似于零，则输出 y 也接近于 x 。此时，残差块（Residual Block）能够执行近似恒等映射，使得模型可以灵活地进行非线性特征学习，而无需强制拟合过于复杂的映射，从而避免不必要的计算开销。

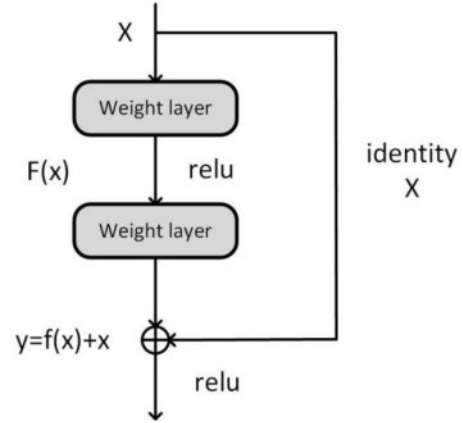


图11 残差学习

基于此，引入了 SRResNet 结构，以增强模型对关键特征的提取能力和整体性能。如图 12 所示，深度残差模块由多个残差块堆叠而成，每个残差块内部均采用跳跃连接，即前述的捷径结构。在具体实现上，每个残差块首先对输入数据进行一次卷积（Convolution, Conv）和批量归一化（Batch Normalization, BN），随后使用 ReLU 进行激活，再次经过卷积和批量归一化处理后，与输入进行逐元素求和，从而实现 $y = f(x) + x$ 的映射。

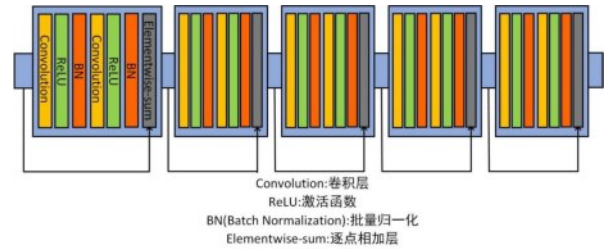


图12 深度残差块

经过残差卷积处理，卷积神经网络的最后一个卷积层后引入额外的层，并结合了额外的激活函数和批量归一化操作，这允许采用更高的学习率进行训练，并降低结果对权重初始化的敏感性。完整的融合卷积神经网络如图 13 所示，在该神经网络中，本文利用 Ghost 卷积的轻量化特性减少参数冗余，

同时结合 SRResNet 的深层残差结构来增强网络的表达能力。

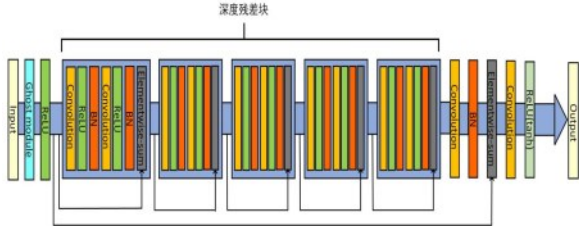


图 13 融合卷积神经网络结构

3.4 损失函数

由于传统损失函数在实际应用中存在对异常值敏感，稀疏性不足和优化困难等问题，因此本文引入了 RoBoss（Robust, Bounded, Sparse, and Smooth Loss Function）损失函数^[40]，其数学表达式如下所示：

$$\ell_{\text{RoBoss}}(u) = \begin{cases} \lambda \{1 - (au + 1) \exp(-au)\}, & u > 0 \\ 0, & u \leq 0 \end{cases} \quad (32)$$

其中， $u = 1 - y_k(\omega^T x_k + b)$ 是分类的间隔误差，并且 y_k 是样本的标签， x_k 是样本向量， ω 是权重向量， b 是偏置项； a 是形状参数，控制损失函数的形状和惩罚强度； λ 是边界参数，控制损失函数的上界，确保损失值不会无限增大。

3.5 专家引导策略

专家引导策略通过多阶段校验机制对智能体输出的动作进行评估和约束。为了保证动作选择的合理性与系统运行的稳定性，智能体首先基于当前环境状态计算各动作的 Q 值，并采用 ϵ -贪心策略完成初步动作选择。随后，专家系统同步启动多维度约束校验，依次进行时隙容量验证、周期匹配性检查和队列偏移值评估，当检测到可能影响性能的无效动作时，系统立即启动备选方案生成机制，从通过约束校验的有效动作集合中选取 Q 值次优的替代方案。该机制通过预判性干预既避免了系统性能退化，又通过提供高质量决策样本加速了智能体对环境动态特性的学习，从而提升了状态-动作映射策略的构建效率。

动作评估模型旨在衡量候选动作对网络资源占用、延迟和负载均衡的综合影响，并提供相应的奖励调整，如图 14。

为了考虑网络负载波动，引入动态容量因子 θ_d ：

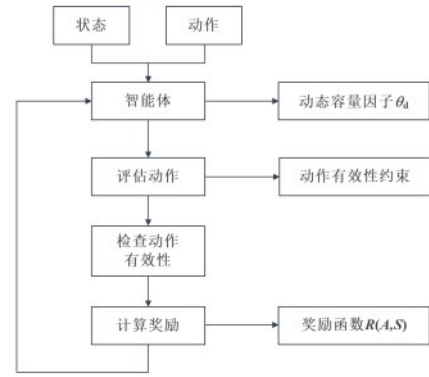


图 14 动作评估算法流程图

$$\theta_d = \begin{cases} 0.9, & \text{if } \rho > 0.8 \\ 1.1, & \text{if } \rho < 0.5 \\ 1.0, & \text{if otherwise} \end{cases} \quad (33)$$

网络负载比 ρ 定义为：

$$\rho = \frac{\sum_{t,q} s_{t,q}}{N_T \cdot R_{\text{slot}}} \quad (34)$$

$s_{t,q}$ 表示在第 t 个时隙、第 q 个队列上的资源占用量， N_T 为总时隙数， R_{slot} 为单时隙容量；通过 θ_d 的动态调整，智能体能够在高负载或低负载条件下灵活控制资源分配，避免超容量或资源浪费。动态修正后的有效时隙容量为：

$$R_{\text{eff}} = \theta_d \cdot R_{\text{slot}} \quad (35)$$

在此基础上，动作 A 的有效性需满足：

$$A \cdot T \leq \text{period}_i \quad (36)$$

即分配的目标时隙索引应位于当前业务流的传输周期 period_i 内，若智能体选择的动作超出该范围，则视为无效动作并给予惩罚。

结合负载均衡与延迟因素，奖励函数 $R(A, S)$ 定义为：

$$R(A, S) = \begin{cases} -10, & \text{if invalid}(A) \\ (1 - \alpha \Delta \rho) - \beta \frac{\sum_{t,q} 1(s_{t,q} > R_{\text{eff}})}{N_T}, & \text{if otherwise} \end{cases} \quad (37)$$

其中，负载变化率 $\Delta \rho$ 表示为：

$$\Delta \rho = \frac{\max(S_{\text{new}}) - \max(S_{\text{old}})}{\max(S_{\text{old}})} \quad (38)$$

指示函数 $1(\cdot)$ 用于计算超容量比例， α 与 β 为权重系数，平衡负载优化与动作合法性。通过该奖励函数，智能体既能优化资源利用，又能避免非法

动作，从而在大规模网络中学习高效、稳定的调度策略。

3.6 MQSM总结

本文设计的MQSM机制由多队列CQF与GSR算法协同构成，旨在实现高效的网络资源调度，满足本文优化目标。多队列CQF作为底层转发框架，通过多独立循环队列及相应的调度策略设计，确保了各类型流的有序转发与时延控制。而GSR算法作为调度决策核心，运行在多队列CQF模型之上，通过构建二维时间-资源利用率矩阵，感知各时隙与队列资源的占用情况，并借助GhostNet与ResNet融合的深度神经网络提取网络状态特征，进而基于强化学习策略动态生成时隙-队列映射的调度动作。具体而言，多队列CQF提供可调度的基础单元，GSR算法则负责在该结构上进行最优时隙调度，两者共同构成了MQSM机制的完整运行体系。

在后续仿真验证中，考虑到多队列CQF已作为MQSM业务转发队列的底层实现，实验重点将聚焦于GSR算法在不同环境下对调度性能的优化表现，包括稳定性、部署速度、业务流调度能力与时隙资源利用率，从而验证MQSM的整体性能优势。

4 仿真与分析

4.1 仿真参数设置

本实验的所有框架使用RoBoSS损失函数训练，并采用Adam优化器进行参数更新，网络模型的输入采用三通道张量，格式为 $[3, H, W]$ ，其中H和W根据环境动态确定。初始学习率设为 $1e-5$ 。若在10个episodes内模型性能没有提升，则保持当前学习率不变。训练时的梯度更新批量大小为32。当经验回放缓冲区的样本数量超过20000条后，开始更新网络参数。实验环境基于Windows 10操作系统，搭载NVIDIA RTX 4090 GPU，使用Python 3接口的PyTorch 1.13深度学习框架完成仿真实验，分析GSR算法对于混合业务流的调度性能。

表2具体描述了网络的参数设置，用于模拟不同流量场景下的调度策略与资源分配。仿真调度的混合业务流数量为1000个，突发流在每个传输时隙内随机产生。此外，本文所采用的仿真网络由若干无人机子网构成，收发两端的无人机通过组网完

成数据传输。考虑到研究重点在于混合业务流在复杂环境下的QoS保障问题，为确保仿真结果的可比性与调度策略的有效性，仿真过程中采用了相对稳定、结合了星型、树型和环型等常见拓扑结构的复合型网络拓扑进行调度性能验证，以检验所提出的MQSM的调度能力。

表2 网络参数设置

网络参数	参数值		
时隙长度	400us	800us	1600us
业务流周期	[400us,1600us,3200us]		
	[400us,1600us]	[1600us,3200us]	
链路带宽	1000Mbps		
循环队列数	3		
尽力传输队列数量	1		
数据包大小	50-1000Bytes		
调度周期	1600us,3200us		
偏移值	2(快速转发为1)		
子网节点数	5		
子网数量	4		
突发流预留带宽	50Kbit/s		

表3展示了GSR算法的具体参数设置，其中经验池大小、探索率、折扣因子、学习率等控制强化学习的经验回放和策略更新，而更新频率、目标网络更新频率等影响收敛速度和资源调度策略性能。这些参数共同作用于探索与利用的平衡、训练的稳定性及资源优化能力，确保算法高效运行。

表3 GSR算法参数设置

算法参数	参数值	算法参数	参数值
经验池大小	1e5	更新频率	50
单次训练样本数	32	奖励因子	1
初始探索率	0.4	折扣因子	0.99
最终探索率	0.1	学习率	1e-5
探索率衰减系数	1e4	目标网络更新频率	50
训练调节因子	0.95	业务调度权重	0.9
训练回合数	1000	资源利用率权重	0.1

4.2 仿真实验对比与分析

图15展示了调度算法的训练过程，其中传输时隙长度为400us，业务流周期为{400us, 1600us, 3200us}。从图中可知，GSR算法相比于TimeDRS^[31]算法最终奖励值高出35%，这是因为算法采用了改进后的SRResnet，拥有更稳定的梯度传播和更强的非线性增强；相比于Tabu^[41]算法，算法在迭代200次后已经开始收敛，这是因为它可以更快地

从历史经验中提取关键特征；而DeepCQF^[42]由于仅用单个全连接层输入一维队列资源信息，难以捕捉时隙与队列之间的关联，因此不利于探索时隙和队列之间的相关性，这导致后续奖励值的下降。

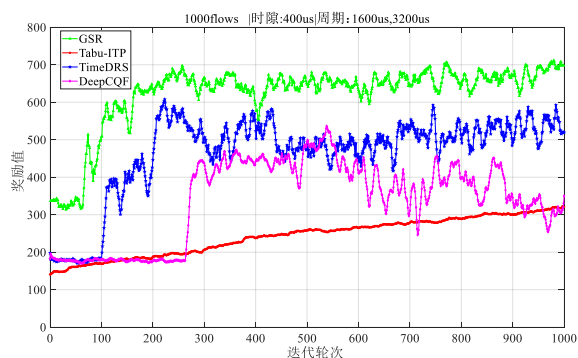


图 15 调度算法训练过程

图 16 展示了测试过程中训练模型的总执行时间，在训练过程中每 50 次迭代保存一个训练模型。Tabu 作为非训练型的传统启发式算法，不具备可比性。从图中可知，GSR 算法的平均执行时间为 0.44 秒，波动范围控制在 ± 0.32 秒内，相比于 TimeDRS，随着迭代次数的增加，在迭代 250 次后，GSR 算法的平均执行时间低出 41%，且始终保持在较低水平，且没有出现较大的波动，表现出了较强的计算稳定性，并避免了计算开销过高的问题；而 DeepCQF 由于仅用单个全连接层输入一维队列资源信息，虽然执行时间更低，但计算能力较弱，无法处理复杂任务。

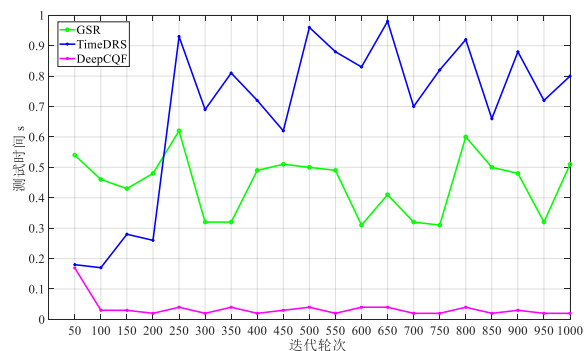


图 16 模型总测试时间对比

图 17 展示了测试过程中各算法完成训练所需的总执行步数。从图中可知，随着迭代次数的增加，在迭代 250 次后，GSR 算法完成收敛平均需要 181 步，相比于 TimeDRS，GSR 执行次数减少减少约 66%，能够在更少的步数内完成任务，避免了

TimeDRS 中多残差块堆积所造成的冗余计算问题，确保了计算稳定性；同时，GSR 在执行步数上接近 DeepCQF（相差不足 5%），说明 GSR 算法具有合理的学习效率，适合复杂环境。

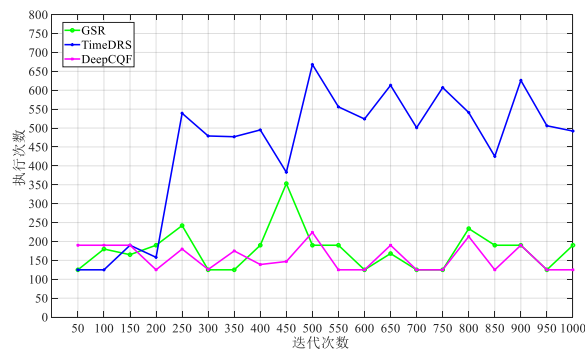


图 17 模型总执行步数对比

图 18 展示了不同时隙长度下各算法调度的业务流个数。从图中可知，随着传输时隙的增大，调度算法对于混合业务流的调度数量逐渐增加，这不仅是因为传输时隙增大导致训练过程中更易找到最优解，也由于时隙长度增大会导致其对应的业务流容量增大，可以容纳并调度更多的数据流。而 GSR 算法相比于其他三种调度算法，能够调度更多的混合业务流，具有更优秀的调度能力。

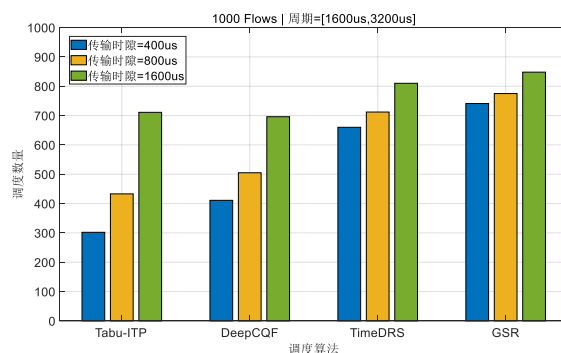


图 18 时隙长度对业务流调度数量的影响

图 19 展示了各类算法在不同时隙长度下的时隙资源使用情况。从图中可以观察到，由于本文设定的时隙资源利用率权重因子较小，为 0.1，因此随着传输时隙长度的增加，GSR 算法在资源利用率方面与其他算法的差距逐渐减小。400us 时的资源利用率达到 83%，比其他算法平均高出 20%；当时隙增大到 1600us 时，仍保持 86% 的利用率水平。同时，图 19 还表明业务流的调度数量与时隙资源使用率之间具有较强的正相关性。这是因为当时隙

资源被更充分利用时，单位时间内可传输的业务流数量随之上升，从而提升了系统的总体业务承载能力。

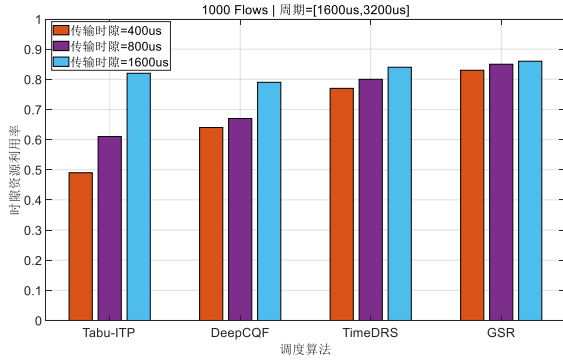


图19 时隙长度对时隙资源利用率的影响

图20展示了不同业务流周期组合下，各算法调度的业务流个数。实验将业务流划分为三类周期组合，分别为{400us, 1600us}、{400us, 1600us, 3200us}和{1600us, 3200us}。通过对比发现，在前两组周期组合中（即包含较短周期400us的情形），各种算法的调度的业务流数量差异较小，平均相差10。在仅包含较长周期（即{1600us, 3200us}）的业务流组中，各算法的调度能力表现更佳，尤其是GSR算法的调度数量接近750。这主要是因为短周期业务流在整个调度过程中出现频繁，占据了大量资源，限制了系统整体的调度效率。此外，与其他算法相比，GSR算法无论在何种业务周期条件下，

均展现出更加突出的性能，能够支持调度更多的混合流。

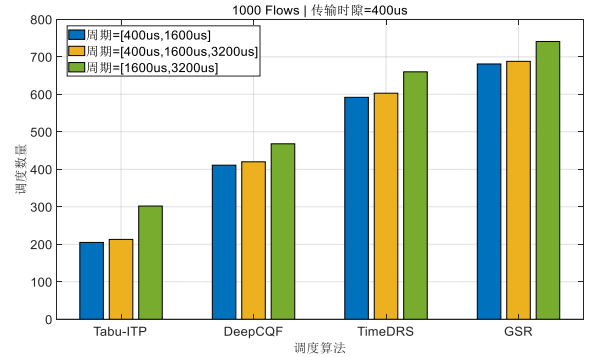


图20 业务流周期对业务流调度数量的影响

4.3 动态拓扑适应性分析

为分析网络动态性对调度算法的影响，本小节在系统模型构建中引入了动态拓扑测试环境。为保持实验条件的一致性，传输时隙长度固定为1600μs，业务流周期集合设定为{1600μs, 3200μs}，其余参数配置与前述实验保持一致。节点动态性通过随机过程进行描述：节点到达行为采用泊松过程建模，离开行为服从指数分布。为了量化拓扑波动强度，我们定义拓扑变化率 λ_{topo} ，用于衡量连续时间片内网络链路集合的差异。由于每个节点的接入与退出都会引起多条链路的变化，因此 λ_{topo} 与 $(\lambda + \mu)$ 近似成正比。具体参数设置如表5所示。

表5 动态场景参数

场景名称	节点数量变化	到达率 λ (节点/秒)	离开率 μ (节点/秒)	拓扑变化率 λ_{topo} (链路/秒)
节点扩张	20 → 35	0.065	0.015	低
快速扩张	20 → 45	0.167	0.083	中
节点流失	20 → 12	0.112	0.138	中
剧烈波动	20 → 15 → 25	0.183	0.217	高
		0.233	0.167	

图21展示了在稳定扩展场景下，GSR算法和TimeDRS算法的调度成功率。实验设置了四类典型动态场景，分别为节点扩张（20→35）、快速扩张（20→45）、节点流失（20→12）和剧烈波动（20→15→25）。在引入动态性后，两种算法的性能均出现下降。但GSR算法在四类场景下均表现出较高的调度成功率，特别是在网络剧烈波动场景中，GSR的调度成功率相比TimeDRS提升10.9%，展现出明显的性能优势。实验结果表明，虽然动态

性影响了网络拓扑稳定性，但GSR算法能够有效缓解节点移动性带来的冲击。其多队列CQF结构为业务流提供冗余传输路径，当链路因节点变化而失效时，算法可自动选择最优路径。

根据上述仿真可知，GSR算法在不同的场景下均表现出了良好的适应性，稳定性和混合业务流调度能力，并且能够在较少的训练轮次生成高性能的模型。即便在动态拓扑环境中，节点频繁加入与离开造成网络结构波动，算法仍能维持较高的调度成

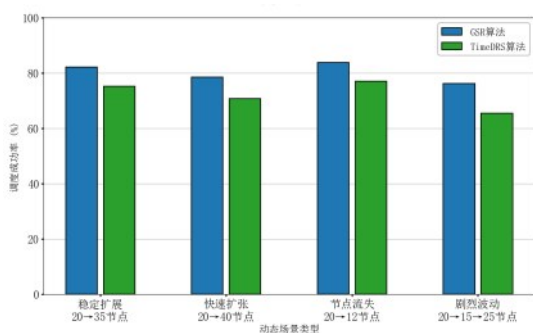


图21 不同动态场景下GSR算法与TimeDRS算法调度成功率对比

功率和资源利用率, 展现出较强的环境适应性。总而言之, GSR算法在实际应用, 如灾后救援快速部署的环境中, 有比较好的实用价值。

5 结束语

本文主要研究基于UANET的MAC层队列调度问题。针对UANET在业务流传输中的混合流调度问题, 本文设计了MQSM业务调度机制。首先通过引入并搭建多队列CQF模型及调度策略来满足混合流调度的低时延、确定性传输需求, 解决了死区时间问题。具体来说, 多队列CQF模型在混合业务流转发过程中, 对不同类型的业务流采用不同的偏移值以实现确定性传输和低时延通信, 并通过BE队列兼顾了尽力而为流的转发。其次, 提出了一种基于深度强化学习的时间资源调度算法, 即GSR算法, 并与搭建的队列模型相适配。该算法主要通过时间-资源利用率矩阵, 卷积神经网络和损失函数等设计, 提升了网络的整体性能。通过仿真实验验证, GSR算法在部署速度, 业务流调度能力与时隙资源利用率方面优于其他时间资源调度算法, 并能在动态拓扑环境下保持稳定运行, 满足了例如应急通信环境下的混合业务流调度性能要求, 实现了整体网络的低时延与可靠通信。

参考文献:

- [1] 韩东升, 郎宇航, 黄丽妍. 面向节点动态优先级的无人机信息收集优化算法[J]. 电子测量技术, 2025, 48(9): 65-74.
HAN D S, LANG Y H, HUANG L Y. Optimization algorithm for UAV information collection based on dynamic node priority[J]. Electronic Measurement Technology, 2025, 48(9): 65-74.
- [2] SATHISHKUMAR N, PALANISAMY S K, NATARAJAN R, et al. Experimental investigations of dual functional substrate integrated waveguide antenna with enhanced directivity for 5G mobile communications[J]. Heliyon, 2024, 10(17): e36929.

- [3] NWANERI I, AL-TURJMAN F. Application of artificial intelligence in mobile networks: a survey[M]//Artificial Intelligence of Things (AIoT). Cambridge:Morgan Kaufmann, 2025: 15-22.
- [4] BAI S, WANG X, GURSOY M C, et al. Deep Reinforcement Learning for Rechargeable UAV-Assisted Data Collection from Dense Mobile Sensor Nodes[J]. IEEE Access, 2025, 13: 28398-28407.
- [5] SUN C, FONTANESI G, CANBERK B, et al. Advancing uav communications: A comprehensive survey of cutting-edge machine learning techniques[J]. IEEE Open Journal of Vehicular Technology, 2024, (99): 1-31.
- [6] 马丁. 基于无人机辅助无线通信性能优化研究[D]. 兰州交通大学, 2021.
MA D. Research on performance optimization of UAV-assisted wireless communication[D]. Lanzhou: Lanzhou Jiaotong University, 2021.
- [7] 闵全, 张格鹏, 鲁晓飞. 无线通信在复杂电磁环境下的抗干扰技术研究[J]. 数字技术与应用, 2021, 39(04): 55-57.
MIN Q, ZHANG G P, LU X F. Anti-jamming technology of wireless communication in complex electromagnetic environment[J]. Digital Technology and Application, 2021, 39(04): 55-57.
- [8] GUO J, GAO H, LIU Z, et al. ICRA: An intelligent clustering routing approach for UAV ad hoc networks[J]. IEEE transactions on intelligent transportation systems, 2022, 24(2): 2447-2460.
- [9] 孙彦景, 李林, 王博文, 等. 灾后无人机自组网高动态多信道TDMA调度算法[J]. 西安电子科技大学学报, 2024, 51(02): 56-67.
SUN Y J, LI L, WANG B W, et al. High-dynamic multi-channel TDMA scheduling algorithm for UAV ad hoc networks after disaster[J]. Journal of Xidian University, 2024, 51(02): 56-67.
- [10] DASDEMIR E, BATTAR, KÖKSALAN M, et al. UAV routing for reconnaissance mission: A multi-objective orienteering problem with time-dependent prizes and multiple connections[J]. Computers & Operations Research, 2022, 145: 105882.
- [11] HAN S I. Survey on UAV deployment and trajectory in wireless communication networks: Applications and challenges[J]. Information, 2022, 13(8): 389.
- [12] MAXA J A, MAHMOUD M S B, LARRIEU N. Performance evaluation of a new secure routing protocol for UAV Ad hoc Network[C]//2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC). Piscataway: IEEE Press, 2019: 1-10.
- [13] 张子东, 宋志群, 刘玉涛, 等. 基于TDMA的空中自组网MAC层资源调度算法研究[J]. 计算机测量与控制, 2024, 32(01): 251-259.
ZHANG Z D, SONG Z Q, LIU Y T, et al. Research on TDMA-based MAC layer resource scheduling algorithm for aerial ad hoc networks[J]. Computer Measurement & Control, 2024, 32(01): 251-259.
- [14] 周佑昊. 无线自组网资源调度方法研究[D]. 北京: 北京邮电大学, 2022.
ZHOU J H. Research on resource scheduling methods for wireless

- ad hoc networks[D]. Beijing: Beijing University of Posts and Telecommunications, 2022.
- [15] SEAH W K G, LEE C H, LIN Y D, et al. Combined communication and computing resource scheduling in sliced 5G multi-access edge computing systems[J]. IEEE Transactions on Vehicular Technology, 2021, 71(3): 3144-3154.
- [16] LIANG B, OBaidat M S, LIU X, et al. Resource scheduling based on priority ladders for multiple performance requirements in wireless body area networks[J]. IEEE Transactions on Vehicular Technology, 2021, 70(7): 7027-7036.
- [17] ABBOU A N, TALEB T, SONG J S. A software-defined queuing framework for QoS provisioning in 5G and beyond mobile systems[J]. IEEE Network, 2021, 35(2): 168-173.
- [18] DOCQUIER T, SONG Y Q, CHEVRIER V. On the relevance of TSN for Substation Communication Networks[C]//2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). Piscataway: IEEE Press, IEEE, 2022: 1-8.
- [19] WU Y, PAN W, CHEN Z, et al. Optimized Mapping and Hybrid Transmission Integrating TAS and CQF in TSN[C]//2024 8th International Conference on Communication and Information Systems (ICCIS). Piscataway: IEEE Press, 2024: 82-88.
- [20] GHOSH A, KUNZ T, YOUSEFI S. Importance of realistic considerations of time synchronization for TAS in TSN networks[C]//2023 International Symposium on Networks, Computers and Communications (ISNCC). Piscataway: IEEE Press, 2023: 1-6.
- [21] NIE H, LI S, LIU Y. TAMCQF: Hybrid traffic scheduling mechanism integrating tas and multi-cqf in tsn[C]//2023 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE Press, 2023: 329-334.
- [22] ALEXANDRIS K, POP P, WANG T. Configuration and evaluation of multi-CQF shapers in IEEE 802.1 time-sensitive networking (TSN)[J]. IEEE Access, 2022, 10: 109068-109081.
- [23] YANG D, CHENG Z, ZHANG W, et al. Burst-aware time-triggered flow scheduling with enhanced multi-CQF in time-sensitive networks[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 2809-2824.
- [24] CHEN Y, XU L, LI L, et al. Cfs: A cqf-based fast scheduling scheme for time-sensitive networking[C]//2023 42nd Chinese Control Conference (CCC). Piscataway: IEEE Press, 2023: 6221-6226.
- [25] 丁赛程. 基于时变流的时间敏感网络资源调度研究[D]. 成都: 电子科技大学, 2024.
- DING B C. Research on time-sensitive network resource scheduling based on time-varying flow [D]. Chengdu: University of Electronic Science and Technology of China, 2024.
- [26] 李妍珏. 时间敏感网络中的混合调度技术研究[D]. 北京: 北京邮电大学, 2023.
- LI Y J. Research on hybrid scheduling technology in time-sensitive networks [D]. Beijing: Beijing University of Posts and Telecommunications, 2023.
- [27] CHENG Z, YANG D, GUO R, et al. Joint time-frequency resource scheduling over CQF-based TSN-5G system[C]//2023 15th International Conference on Communication Software and Networks (ICCSN). Piscataway: IEEE Press, 2023: 60-65.
- [28] 于简豪. 面向CQF模型的TSN资源调度算法的研究[D]. 沈阳: 辽宁大学, 2023.
- YU J H. Research on TSN resource scheduling algorithm for CQF model [D]. Shenyang: Liaoning University, 2023.
- [29] 聂宏蕊. 时间敏感网络流量调度机制研究[D]. 北京: 北京邮电大学, 2024.
- NIE H R. Research on traffic scheduling mechanism in time-sensitive networks [D]. Beijing: Beijing University of Posts and Telecommunications, 2024.
- [30] 王文譞. 面向车内网络确定性传输的数据调度及路由协同机制研究[D]. 成都: 电子科技大学, 2024.
- WANG W X. Research on data scheduling and routing coordination mechanism for deterministic transmission in in-vehicle networks [D]. Chengdu: University of Electronic Science and Technology of China, 2024.
- [31] YANG D, CHENG Z, ZHANG W, et al. Burst-aware time-triggered flow scheduling with enhanced multi-CQF in time-sensitive networks[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 2809-2824.
- [32] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2018: 4510-4520.
- [33] HOWARD A, SANDLER M, CHU G, et al. Searching for MobileNetV3[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Seoul: IEEE Press, 2019: 1314-1324.
- [34] MA N, ZHANG X, ZHENG H-T, et al. ShuffleNet V2: Practical guidelines for efficient CNN architecture design[C]//Proceedings of the European Conference on Computer Vision (ECCV). Cham: Springer, 2018: 128-138.
- [35] TAN M, LE Q V. EfficientNet: Rethinking model scaling for convolutional neural networks[C]//Proceedings of the 36th International Conference on Machine Learning (ICML). Long Beach: Curran Associates, Inc., 2019: 10691 - 10700.
- [36] HAN K, WANG Y, TIAN Q, et al. Ghostnet: More features from cheap operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 1577-1586.
- [37] 孔令鑫, 陈紫强, 晋良念, 等. 融合可变形卷积的轻量级路面病害检测算法[J]. 科学技术与工程, 2025, 25(02): 683-694.
- KONG L X, CHEN Z Q, JIN L N, et al. Lightweight pavement disease detection algorithm fused with deformable convolution [J]. Science Technology and Engineering, 2025, 25 (02): 683-694.
- [38] YUAN B, JACOB B, CHEN W, et al. Downscaling sea surface

- height and currents in coastal regions using convolutional neural network[J]. Applied Ocean Research, 2024, 151: 104153.
- [39] 卢冰,李灿林,冯薛龙,等. 基于改进 SRResNet 深度学习的低照度图像超分辨率重建方法[J]. 科学技术与工程, 2022, 22(27): 12045-12052.
- LU B, LI C L, FENG X L, et al. Low-illumination image super-resolution reconstruction method based on improved SRResNet deep learning network[J]. Science Technology and Engineering, 2022, 22(27): 12045-12052.
- [40] AKHTAR M, TANVEER M, ARSHAD M. RoBoSS: A robust, bounded, sparse, and smooth loss function for supervised learning [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 47(1): 149-160.
- [41] YAN J, QUAN W, JIANG X, et al. Injection time planning: Making CQF practical in time-sensitive networking[C]//IEEE INFOCOM 2020-IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 616-625.
- [42] CHENG Z, YANG D, ZHANG W, et al. DeepCQF: Making CQF scheduling more intelligent and practicable[C]//ICC 2022-IEEE International Conference on Communications. Piscataway: IEEE Press, 2022: 1-6.