**Computer Science and Application** 计算机科学与应用, 2025, 15(5), 592-605
Published Online May 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.155132

Hans汉斯

# 基于强化学习的低轨卫星网络动态分布式路由算法研究

**訾鑫源，刘健培*，邝　坚**

北京邮电大学计算机学院，北京

收稿日期：2025年4月12日；录用日期：2025年5月14日；发布日期：2025年5月22日

## 摘　要

**随着全球通信需求的快速增长，低轨卫星网络凭借其覆盖范围广、灵活性高的优势成为地面通信的有力补充。然而，卫星的高速运动导致网络拓扑和链路状态动态变化，为路由算法设计带来巨大挑战。为应对上述问题，本文提出一种基于强化学习的动态分布式路由算法。首先系统性建模卫星网络通信过程，涵盖网络拓扑、通信链路、通信时延及丢包等要素；其次，提出动态分布式路由架构，引入星间状态交换机制，使卫星节点能够实时感知网络状态变化并自主决策；然后，将路由决策问题建模为分布式部分可观察马尔可夫决策过程(Dec-POMDP)，提出结合Double DQN和Dueling DQN的MAD3QN路由算法，构建高效状态表示和奖励函数，有效引导智能体优化路由决策。仿真结果表明，与其他路由算法相比，MAD3QN算法在端到端时延、丢包率、吞吐量等性能指标上均表现更优，充分证明了该算法对低轨卫星网络高动态环境的适应性与有效性。**

## 关键词

**低轨卫星网络，分布式路由，强化学习**

# Research on Dynamic Distributed Routing Algorithm for Low Earth Orbit Satellite Networks Based on Reinforcement Learning

**Xinyuan Zi, Jianpei Liu*, Jian Kuang**

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing

Received: Apr. 12th, 2025; accepted: May 14th, 2025; published: May 22nd, 2025

*通讯作者。

文章引用: 訾鑫源, 刘健培, 邝坚. 基于强化学习的低轨卫星网络动态分布式路由算法研究[J]. 计算机科学与应用, 2025, 15(5): 592-605. DOI: 10.12677/csa.2025.155132

## Abstract

**With the rapid growth of global communications demands, low Earth orbit (LEO) satellite networks have become a powerful supplement to ground communications due to their wide coverage and high flexibility. However, the high-speed movement of satellites leads to dynamic changes in network topology and link status, posing significant challenges for routing algorithm design. To address these issues, this paper proposes a dynamic distributed routing algorithm based on reinforcement learning. First, we systematically model the satellite network communication process, covering factors such as network topology, communication links, communication delay, and packet loss. Second, we introduce a dynamic distributed routing architecture and a mechanism for inter-satellite state exchange, enabling satellite nodes to perceive network state changes in real time and make autonomous decisions. Next, we model the routing decision problem as a Distributed Partially Observable Markov Decision Process (Dec-POMDP) and propose the MAD3QN routing algorithm, which combines Double DQN and Dueling DQN. The algorithm constructs an efficient state representation and reward function to effectively guide the agent in optimizing routing decisions. Simulation results show that compared to other routing algorithms, the MAD3QN algorithm outperforms in performance metrics such as end-to-end delay, packet loss rate, and throughput, demonstrating its adaptability and effectiveness in the highly dynamic environment of LEO satellite networks.**

## Keywords

**Low Earth Orbit Satellite Network, Distributed Routing, Reinforcement Learning**



## 1. 引言

随着全球通信需求不断增长，低轨卫星网络作为新兴通信技术，正迅速发展并广泛应用于互联网覆盖、物联网连接等领域。相比传统地面通信，低轨卫星网络能够突破地理环境限制，为偏远和灾害地区提供持续稳定的通信服务。

低轨卫星网络具有以下特点：(1) 卫星绕地高速运行，导致网络拓扑呈现明显的时变性。(2) 受空间损耗等因素影响，网络链路状态频繁变化。(3) 网络流量分布不均，用户密集区域部分链路过载，而稀疏区域则大多链路闲置。

由于低轨卫星网络整体呈现高度动态性，传统基于固定链路配置的路由策略难以适用，亟需设计一种具备动态自适应能力的路由算法，以应对网络拓扑与链路状态变化问题，并综合考虑时延、带宽和网络负载等因素，实现高效可靠的数据传输。

根据决策节点的位置，低轨卫星网络路由算法可分为集中式和分布式两类。集中式算法依赖中心节点收集全网状态信息并统一计算路由。Jiang 等人[1]将动态拓扑划分为多个静态快照，收集各快照内的全局链路信息，结合 Dijkstra 算法计算最优路径。Zhu 等人[2]在集中式路由计算中引入拥塞感知机制，动态规避拥塞链路。Li 等人[3]引入软件定义网络(SDN)架构，将链路状态汇聚至中央控制器，实现灵活路由调度。分布式算法无需依赖中心节点，由各节点基于本地信息独立决策。Pan 等人[4]提出轨道预测最短路径优先路由算法(OPSPF)，通过星座周期性生成路由表并动态更新以应对拓扑变化。Zhang 等人[5]提出基于区域的分布式路由协议(ASER)，采用层次化路由机制对卫星分组，提升了区域间通信的稳定性与效率。

近年来，强化学习凭借自适应与动态决策能力，在卫星网络路由中备受关注。Yin 等人[6]将卫星路由建模为马尔可夫决策过程(MDP)，采用 Q-learning 实现集中式决策。Huang 等人[7]提出基于 Q-learning 的动态分布式路由方案(QRLSN)，将卫星节点视为独立智能体，采用多目标优化策略选择最优下一跳。Zuo 等人[8]基于 DQN 设计分布式路由算法，使节点能根据距离、时延、带宽等因素自适应调整路由策略。

基于全局视图的集中式路由算法虽具备全局优化能力，但依赖中心节点进行全网信息收集与计算，导致通信和计算开销较大，且在动态网络中容易出现决策滞后。相比之下，分布式算法仅依赖节点本地信息进行决策，减少了通信和计算负担。

强化学习在低轨卫星网络路由中展现出巨大潜力，能够根据网络状态的变化自适应调整路由策略。然而，直接将强化学习应用于完全分布式路由时，由于节点缺乏全局视野，容易陷入局部最优解。现有分布式路由算法大多只关注如何利用局部信息进行决策，而忽视了局部状态信息的获取，缺乏清晰的状态交互流程。此外，许多路由算法未能充分建模低轨卫星网络中链路、时延、丢包等动态特性，通常只是基于静态或经验性权重选择路径，难以应对复杂环境中的优化问题。

针对上述问题，本文提出了一种基于强化学习的低轨卫星网络动态分布式路由算法，主要贡献如下：

(1) 系统性建模卫星网络通信过程，涵盖网络拓扑、通信链路、时延和丢包等要素，弥补现有算法中缺乏通信链路和丢包模型的不足。

(2) 提出动态分布式路由架构，使卫星节点能够实时感知网络状态变化并自主决策，引入星间状态交换机制，实现智能体间信息共享与协同优化。

(3) 将低轨卫星网络路由建模为分布式部分可观察马尔科夫决策过程(Dec-POMDP)，提出基于 MAD3QN (Multi-Agent Double Dueling Deep Q-Network)的动态分布式路由算法，通过高效的状态表示和奖励函数引导智能体优化决策，并结合优先经验回放、网络参数共享等优化训练策略，提高训练效率与策略性能。

## 2. 卫星网络通信过程建模

### 2.1. 网络拓扑模型

卫星网络由 M 个轨道面组成，每个轨道面包含 N 颗卫星，轨道面按升交点经度递增方向编号为 $\{1,2,\cdots,M\}$，轨道内卫星按位置编号为 $\{1,2,\cdots,N\}$。将卫星网络建模为图 $G=(V,E)$，其中 V 为卫星和地面站节点集合，E 为星间和星地链路集合。卫星间连接采用“+Grid”拓扑结构[9]，每颗卫星维持四条星间链路，分别与同轨道面的前后相邻卫星及相邻轨道面的对应卫星相连，如图 1 所示。

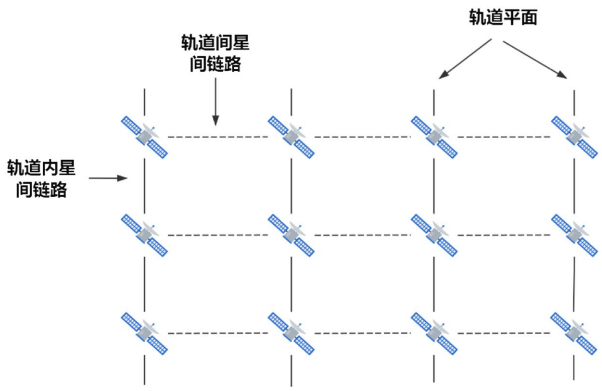


**Figure 1.** Diagram of the inter-satellite link
**图 1.** 卫星间链路示意图

为应对卫星高速运动导致的网络拓扑频繁变化，本文采用虚拟拓扑策略[1]，将连续时间轴划分为一系列离散时隙，假设每个时隙内网络拓扑保持不变，仅在时隙边界时刻更新，以减少拓扑动态变化对路由决策的影响。

## 2.2. 通信链路模型

卫星间通信主要依赖于星间链路，其性能受自由空间路径损耗等因素影响。根据 Friis 传输方程[10]，卫星 $v_i$ 和 $v_j$ 之间的路径损耗 $L_{ij}$ 计算公式如下：

$$L_{ij}=\left(\frac{4\pi d(i,j)f}{c}\right)^2$$

其中，$d(i,j)$ 为卫星间距离，$f$ 为载波频率，$c$ 为光速。

在上述链路损耗条件下，卫星 $v_i$ 和 $v_j$ 之间链路的信噪比 $SNR_{ij}$ 计算公式如下：

$$SNR_{ij}=\frac{P_tG_tG_r}{L_{ij}k_BBT}$$

其中，$P_t$ 为发射天线功率，$G_t$ 和 $G_r$ 分别为发射和接收天线增益，$k_B$ 为玻尔兹曼常数，$B$ 为信道带宽，$T$ 为系统噪声温度。

在无干扰理想环境下，假设星间链路可达到最大传输速率，根据香农定理，链路传输速率 $R_{ij}$ 计算公式如下：

$$R_{ij}=B\log_2\left(1+SNR_{ij}\right)$$

该公式表明，传输速率受信道带宽和信噪比影响，当链路质量较优(SNR 较高)时，通信速率可显著提升。

## 2.3. 通信时延模型

端到端时延 $Delay_{ij}$ 指数据包从源节点 $S_i$ 发送，经中继节点转发至目标节点 $S_j$ 所耗费的总时间，由以下三个部分组成：

$$Delay_{ij}=T_{ij}^{queue}+T_{ij}^{tran}+T_{ij}^{prop}$$

排队时延 $T_{ij}^{queue}$ 指数据包在节点接收队列中等待处理的时间，计算公式如下：

$$T_{ij}^{queue}=\frac{q_i^t\times pkt_{size}}{R_{ij}}$$

其中，$q_i^t$ 为节点 $S_i$ 在时间 $t$ 时的接收队列长度，$pkt_{size}$ 为数据包大小，$R_{ij}$ 为链路传输速率。

传输时延 $T_{ij}^{tran}$ 指数据包写入链路所需时间，与数据包大小 $pkt_{size}$ 和传输速率 $R_{ij}$ 相关，计算公式如下：

$$T_{ij}^{tran}=\frac{pkt_{size}}{R_{ij}}$$

传播时延 $T_{ij}^{prop}$ 指数据包在传播介质中完成传播所需时间，假设信号在真空中以光速传播，计算公式如下：

$$T_{ij}^{prop}=\frac{dis_{ij}}{c}$$

其中，$dis_{ij}$ 为节点间距离，$c$为真空光速。

### 2.4. 通信丢包模型

在低轨卫星网络中，数据包丢失是影响通信质量的关键因素，主要包括拥塞丢包和误码丢包。

(1) 拥塞丢包：当卫星节点的接收队列到达最大容量时，新到达的数据包将无法进入队列并被直接丢弃，丢包率(PLR)计算公式如下：

$$PLR = \begin{cases} 0, & \text{if } qlen \le qlen_{max} \\ 1, & \text{if } qlen > qlen_{max} \end{cases}$$

其中， $qlen$ 为当前接收队列长度， $qlen_{max}$ 为最大容量。

(2) 误码丢包：链路质量由信噪比(SNR)来衡量，SNR 越高，信号强度相对噪声越强，误码率(BER)越低。误码率计算公式如下：

$$BER = Q\left(\sqrt{k * SNR}\right)$$

其中，$Q(x)$为高斯分布的右尾概率函数，$k$ 为与卫星通信调制方式相关的常数。

当 BER 较小时，PLR 可近似表示为：

$$PLR \approx 1 - e^{-N*BER}$$

其中，N 为数据包大小。

拥塞丢包主要由接收队列溢出导致，而误码丢包则取决于链路质量。优化队列管理与链路质量可有效降低数据包丢失，提高通信可靠性。

## 3. 提出的路由算法

### 3.1. 动态分布式路由架构

针对集中式路由算法在低轨卫星网络中的局限性，本文提出了一种动态分布式路由架构，使每个卫星节点基于局部信息自主决策，无需依赖全局网络状态。该架构包括状态信息交换、路由决策和数据包转发三个阶段，如图 2 所示。

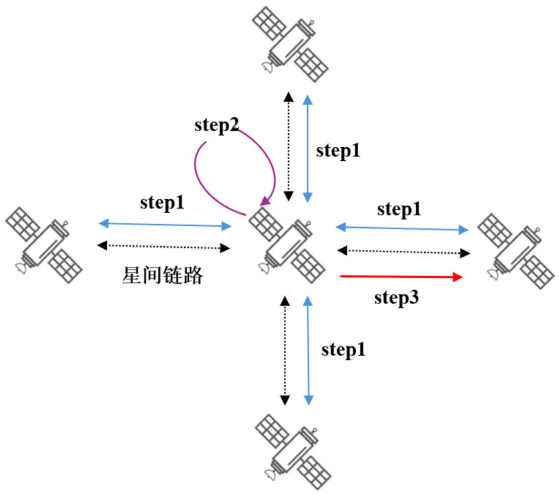


**Figure 2.** Diagram of dynamic distributed routing architecture
**图 2.** 动态分布式路由架构示意图

(1) 状态信息交换：每个卫星节点定期广播自身状态，并接收来自相邻卫星的状态信息，涵盖链路质

量(如信噪比)、流量负载(如队列长度)以及拓扑变化(如卫星相对位置)。

(2) 路由决策：卫星节点接收数据包后，对其进行解析并获取数据包相关状态，结合最新的节点相关状态，输入到路由决策模型中，选择最优的下一跳卫星。

(3) 数据包转发：完成路由决策后，卫星节点将数据包转发至跳卫星，并实时监测转发过程中的关键性能指标(如时延和丢包)，并将这些信息反馈至路由决策模块，以优化后续路由选择。

在该架构中，每颗卫星作为独立智能体，基于自身观测的局部状态信息进行路由决策，避免依赖全局状态信息，降低了通信和计算开销。此外，卫星节点能够实时感知网络状态变化，动态调整路由策略，提高在动态环境中的适应性。

### 3.2. 强化学习建模

基于 3.1 节搭建的动态分布式路由架构，本文将路由优化问题建模为分布式部分可观察马尔可夫决策过程(Dec-POMDP)，并通过多智能体强化学习(MARL)进行求解。在该模型中，每颗卫星被视为独立智能体(Agent)，基于局部观测状态自主决策数据包的下一跳转发节点。Dec-POMDP 模型的关键要素定义如下：

**1、状态空间：**智能体 i (卫星节点$v_i v_i$)的状态空间为 $S^i=\left\{s^i_{packet},s^i_{sat}\right\}$， $s^i_{packet}$ 表示数据包相关状态，$s^i_{sat}$ 表示卫星相关状态。

$$s^i_{packet}=\left\{pos_{cur},pos_{des}\right\}$$

$pos_{cur}$ 为数据包当前所在节点位置， $pos_{des}$ 为数据包目标节点位置。

$$s^i_{sat}=\left\{linkstate^{up}_i,linkstate^{down}_i,linkstate^{left}_i,linkstate^{right}_i\right\}$$

$s^i_{sat}$ 刻画了卫星 $v_i$ 与其邻近卫星间的链路信息，以 $linkstate^{up}_i$ (上方邻近卫星链路)为例，可进一步细分为：

$$linkstate^{up}_i=\left\{pos_{up},dis^{up}_i,qlen_{up},snr^{up}_i\right\}$$

其中， $pos_{up}$ 为上方邻近卫星位置， $dis^{up}_i$ 为当前卫星与上方卫星间的通信距离， $qlen_{up}$ 为上方卫星的接收队列长度(反映链路流量负载)， $snr^{up}_i$ 为链路信噪比(反映链路质量)，其余方向的链路状态定义方式同理。

**2、动作空间：**当数据包达到卫星 $v_i$ 时，智能体根据当前状态 $S^i$，从邻近节点中选择下一跳节点 $v_j$，动作空间 $A^i$ 定义如下：

$$A^i=\left\{a^i_{up},a^i_{down},a^i_{left},a^i_{right}\right\}$$

其中， $a^i_{up}$ 表示将数据包转发给上方邻近卫星，其余项同理。

**3、奖励函数：**智能体的目标是学习最优路由策略，以降低通信时延和丢包，提高网络吞吐量。为此，本文设计了综合考虑多项关键性能指标的奖励函数，用于引导智能体优化路由决策。

当数据包 $k$ 由节点 $v_i$ 转发至节点 $v_j$ 时，智能体 $i$ 获得的即时奖励 $r_{ij}$ 为：

$$r_{ij}=\begin{cases}r_{des}, & \text{if } j \text{ is } des\\ w_1T_{ij}+w_2L_{ij}+w_3C_k+w_4E_{ij}, & \text{other}\end{cases}$$

其中， $r_{des}$ 表示数据包到达目标节点时的正向奖励，用于引导数据包尽快抵达目标节点。

当下一跳节点不是目标节点时，奖励项由以下元素综合决定：

(1) 时延惩罚 $T_{ij}$：包括传输、传播和排队时延，时延越长，奖励越低，促使智能体选择低时延路径。

(2) 丢包惩罚 $L_{ij}$：丢包时赋予较大负面奖励，惩罚不可靠的路由选择，促使智能体选择稳定路径。

(3) 重复节点访问惩罚 $C_k$：当数据包 $k$ 访问到已访问过的节点时，智能体将受到惩罚，避免陷入循环路由。

(4) 趋近目标奖励因子 $E_{ij}$：若下一跳节点 $j$ 相较于当前节点 $i$ 更接近目标节点 $des$，智能体将获得正向奖励，计算方式如下：

$$E_{ij} = dis(i,des) - dis(j,des)$$

其中，$dis(i,des)$ 为节点 $i$ 与节点 $des$ 之间的距离。该奖励因子有助于缓解因智能体局部观测所导致的局部最优问题，引导智能体选择更短路径，提高路由效率。

### 3.3. 算法整体框架

基于 3.2 节建立的强化学习模型，考虑到连续状态空间和有限动作空间的特性，本文采用深度 Q 网络(Deep Q-Network, DQN)优化路由策略。DQN 通过深度神经网络来逼近状态－价值函数 $Q(s,a;\theta)$，在高维状态空间的决策问题中表现优异。智能体通过与环境交互，不断迭代更新网络参数 $\theta$，以学习最优策略 $\pi$，即选择最大化长期回报的动作。DQN 采用时序差分(Temporal Difference, TD)方法更新 $Q(s,a;\theta)$，计算公式如下：

$$Q(s,a;\theta) \leftarrow Q(s,a;\theta) + \alpha\left[r + \gamma \max_{a'} Q(s',a';\theta') - Q(s,a;\theta)\right]$$

其中，$s$ 和 $s'$ 分别为当前状态与下一状态，$a$ 和 $a'$ 分别为当前动作与下一动作，$\alpha$ 为学习率，$r$ 为即时奖励，$\gamma$ 为折扣因子。参数 $\theta$ 和 $\theta'$ 分别对应在线网络和目标网络，$Q(s,a;\theta)$ 表示在状态 $s$ 下选择动作 $a$ 的 Q 值估计，$\max_{a'} Q(s',a';\theta')$ 表示在下一状态 $s'$ 下选择最优动作 $a'$ (使得 Q 值最大的那个动作)的 Q 值。

DQN 的基本框架和训练过程如图 3 所示。为提高学习效率，DQN 引入“经验回放”(Experience Replay)机制。智能体每次与环境交互后，将经验 $(s,a,r,s')$ 存入回放缓冲池中，并在后续训练时从缓冲池中随机采样用于更新神经网络参数。该机制打破了数据的时间相关性，减少过拟合，同时使每个数据可多次用于训练，提高了样本利用率。DQN 在训练过程中还引入了“目标网络”(Target Network)来缓解训练不稳定性。目标网络 $\theta'$ 与在线网络 $\theta$ 结构相同，但参数 $\theta'$ 在一定 $\theta$ 时间内保持固定，仅在若干步后从在线网络 $\theta$ 同步更新一次。训练过程中，动作选择使用迭代更新的在线网络 $\theta$，目标值计算则依赖相对静止的目标网络 $\theta'$，从而减缓了在线网络频繁更新导致的目标值波动，促进模型稳定收敛。

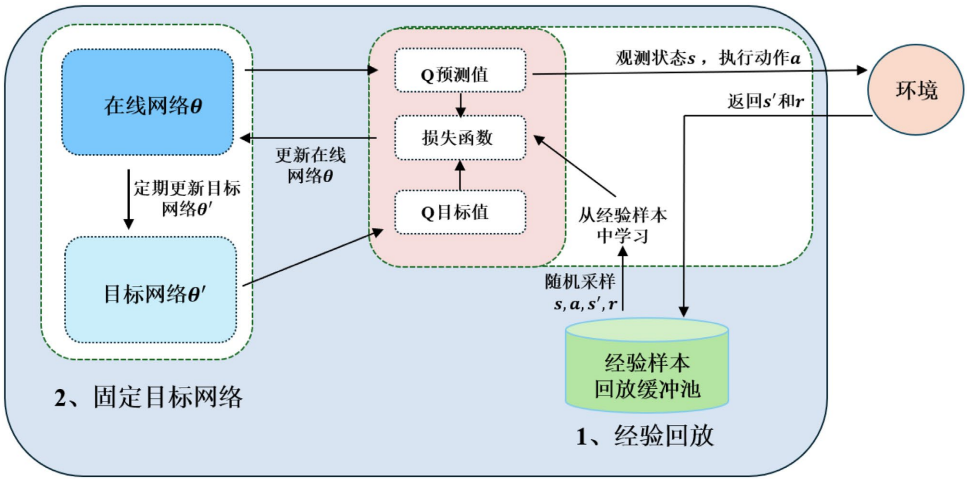


**Figure 3.** Diagram of DQN basic framework and training process
**图 3.** DQN 基本框架和训练过程示意图

为提高 DQN 在多智能体环境中的学习效率和稳定性，本文提出的基于 MAD3QN 的路由算法结合 Double DQN 和 Dueling DQN 这两种改进策略，以优化 Q 值估计，提高学习性能。

**(1) Double DQN：**传统 DQN 在同一网络中执行动作选择与 Q 值计算，易导致 Q 值过估计。针对该问题，Double DQN 通过分离动作选择与 Q 值计算，减少过估计风险。具体而言，Double DQN 使用在线网络 $\theta$ 选择最优动作，而使用目标网络 $\theta'$ 计算该动作的 Q 值，这样可以更准确地估计 Q 值，优化后的 Q 值更新公式如下：

$$Q(s,a;\theta) \leftarrow Q(s,a;\theta) + \alpha\left[r + \gamma Q\left(s', \arg\max_{a'} Q(s',a';\theta);\theta'\right) - Q(s,a;\theta)\right]$$

其中，$Q\left(s', \arg\max_{a'} Q(s',a';\theta);\theta'\right)$ 表示由在线网络 $\theta$ 在下一状态 $s'$ 下选择最优动作 $a'$ (使得 Q 值最大的那个动作)，然后由目标网络 $\theta'$ 计算该最优动作的 Q 值。

**(2) Dueling DQN：**Dueling DQN 网络结构如图 4 所示，将 $Q(s,a;\theta)$ 分解为两部分——状态价值函数 $V(s;\theta)$ 和动作优势函数 $A(s,a;\theta)$，$V(s;\theta)$ 衡量状态$ss$的整体价值，而 $A(s,a;\theta)$ 衡量在状态 $s$ 下选择动作 $a$ 的相对优势。这种分解结构使得网络能够更准确地学习状态的整体价值，而无需完全依赖于特定动作的值，提高了智能体在较大状态空间中的学习效率。Dueling DQN 的 Q 值计算公式如下：

$$Q(s,a;\theta) = V(s;\theta) + A(s,a;\theta) - \frac{1}{|\mathcal{A}|}\sum_{a^* \in \mathcal{A}} A(s,a^*;\theta)$$

其中，$|\mathcal{A}|$ 表示动作空间大小，$\sum_{a^* \in \mathcal{A}} A(s,a^*;\theta)$ 是对所有可能动作的优势函数求和，用于对优势函数进行归一化，以避免值函数和优势函数之间的偏差累积。

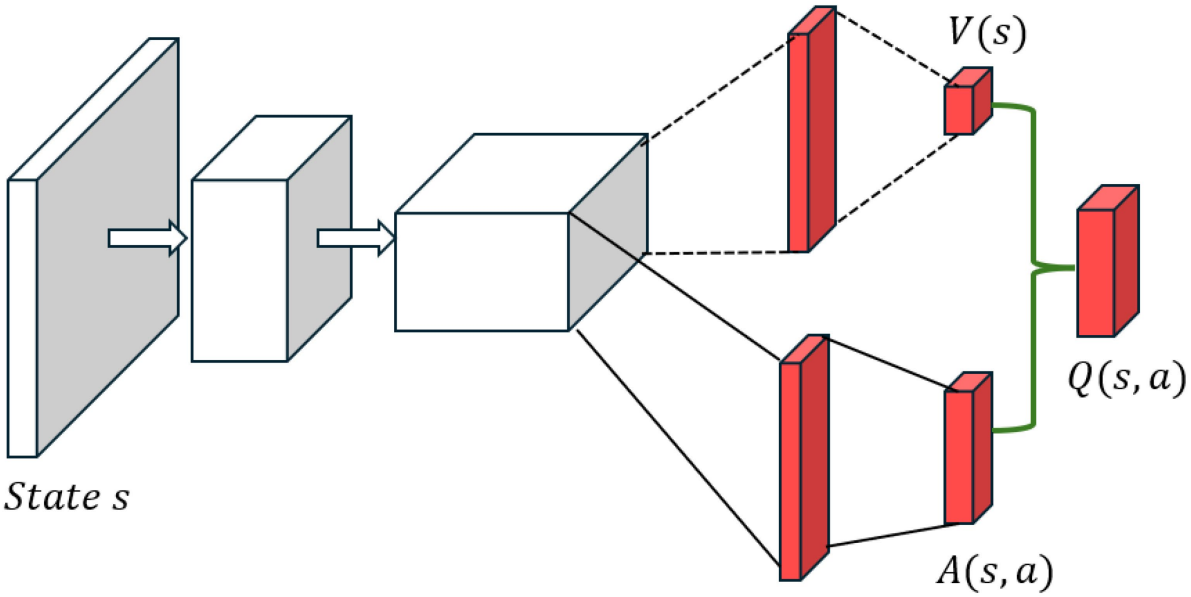


**Figure 4.** Diagram of dueling DQN network structure
**图 4.** Dueling DQN 网络结构示意图

## 3.4. 算法训练过程

基于 3.1 节提出的动态分布式路由架构，每个卫星节点作为独立智能体，采用 D3QN (Double Dueling Deep Q-Network)进行自主路由决策。MAD3QN (Multi-Agent Double Dueling Deep Q-Network) 路由算法的训练过程详见 Algorithm 1。

**Algorithm 1 MAD3QN 路由算法训练过程**

**输入：**训练轮数$N$，学习率$\alpha$，折扣因子$\gamma$，探索率$\epsilon$，批量大小$B$，在线网络更新频率$C_1$，目标网络更新频率$C_2$

**初始化：**全局经验回放池$D$，每个卫星智能体$Agent_i$的在线网络参数$\theta$和目标网络参数$\theta'$

**for** $episode = 1$ to $N$ **do**
  重置仿真环境，地面站开始生成数据包
  $finish \leftarrow false,\ step \leftarrow 0$
  **while not** $finish$ **do**
    每个卫星智能体$Agent_i$接收数据包，获得局部观测状态$s$
    根据在线网络$\theta$，输出路由决策$a$
    执行动作$a$,将数据包转发至下一跳节点
    环境返回下一时刻状态$s'$,并基于时延、丢包等信息计算即时奖励$r$
    将经验$(s,a,r,s')$存储到全局经验回放池$D$
    **if** 所有数据包请求处理完毕 **then**
      $finish \leftarrow true$
    **end if**
    $step \leftarrow step + 1$
    **if** $step$ **mod** $C_1 = 0$ **then**
      $Agent_0$从全局经验回放池$D$中随机采样批次大小为$B$的经验样本
      更新$Agent_0$在线网络参数$\theta$
    **end if**
    **if** $step$ **mod** $C_2 = 0$ **then**
      更新$Agent_0$的目标网络参数$\theta' \leftarrow \theta$
      同步$Agent_0$的网络参数至所有智能体$Agent_i$
    **end if**
  **end while**
**end for**

训练过程中，地面站持续生成数据包，并在卫星网络中转发，直至抵达目标节点。该轮训练将持续进行，直到网络中所有数据包均被处理完毕。

当卫星智能体 $Agent_i$ 接收数据包时，根据局部观测状态 $s$ 和在线网络 $\theta$ 选择下一跳节点。为平衡探索和利用，智能体采用 $\epsilon$ -贪婪策略，以概率 $\epsilon$ 随机选取动作(进行探索)，或以概率 $1-\epsilon$ 选取价值最大的动作(进行利用)，如下所示：

$$a=\begin{cases}\text{随机选取 } a\in A^i, & \text{以概率 } \epsilon\\ \arg\max\limits_{a\in A^i} Q(s,a;\theta), & \text{以概率 } 1-\epsilon\end{cases}$$

完成数据包转发后，环境将反馈新状态 $s'$ 及转发过程中的时延、丢包等信息，智能体根据这些信息计算即时奖励 $r$，并将经验 $(s,a,r,s')$ 存入全局经验回放池。

每经过 $C_1$ 步，中心智能体 $Agent_0$ 从经验回放池中随机采样，通过最小化目标损失函数 $L(\theta)$ 来更新在线网络参数 $\theta$：

$$L(\theta)=\left(y-Q(s,a;\theta)\right)^2$$

其中，$Q(s,a;\theta)$ 为当前 Q 值，$yy$为目标 Q 值。

基于 Double DQN 和 Dueling DQN，目标值 $y$ 的计算公式如下：

$$y = r + \gamma \left[ V\left(s';\theta'\right) + \left( A\left(s', \arg\max_{a'} Q\left(s',a';\theta\right);\theta'\right) - \frac{1}{|\mathcal{A}|}\sum_{a^* \in \mathcal{A}} A\left(s',a^*;\theta'\right) \right) \right]$$

其中，$\theta$ 和 $\theta'$ 分别为在线网络和目标网络参数，$s'$ 为下一状态，$\arg\max_{a'} Q\left(s',a';\theta\right)$ 为在线网络 $\theta$ 在下一状态 $s'$ 下选择的最优动作 $a'$，$V$ 和 $A$ 分别为状态价值函数和动作价值函数。

计算损失函数后，通过梯度下降优化在线网络参数：

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} L\left(\theta\right)$$

其中，$\alpha$ 为学习率，控制参数更新幅度。

每经过 $C_2$ 步，同步在线网络参数 $\theta$ 至目标网络 $\theta'$，避免频繁更新导致模型震荡，提高训练稳定性。

为了在有限资源的条件下确保系统中大量智能体的训练稳定性，并加速路由策略的收敛，本文采用以下方法来优化训练策略：

**(1) 优先经验回放**

传统经验回放对所有样本赋予相同采样概率，难以有效利用关键经验。本文采用优先经验回放，为每条经验赋予优先级，提高重要样本的利用效率，经验样本 $i$ 的优先级为：

$$p_i = \left|\delta_i\right| + \epsilon_p$$

其中，$\delta_i$ 为 TD-Error，表示当前 Q 值和目标 Q 值的差异，$\epsilon_p$ 为防止优先级为零的常数。

根据优先级 $p_i$，定义该经验被采样的概率 $P_i$：

$$P_i = \frac{p_i^{\beta}}{\sum_k p_k^{\beta}}$$

其中，$\beta \geq 0$ 为控制优先级采样程度的超参数。

由于优先经验回放可能导致某些经验被过度采样，为减少采样偏差，引入加权修正因子 $\omega_i$：

$$\omega_i = \left(\frac{1}{N \cdot P_i}\right)^{\lambda}$$

其中，$N$ 为经验总数，$\lambda$ 为控制修正权重的超参数。

在更新网络参数时，将加权因子引入损失函数：

$$L\left(\theta\right) = \frac{1}{B}\sum_{i \in B} \omega_i \left(\left(y_i - Q\left(s,a;\theta\right)\right)^2\right.$$

优先经验回放能够提高采样效率，加速智能体学习最优策略。

**(2) 网络参数和经验共享**

在 MAD3QN 路由算法中，若采用完全去中心化的多智能体强化学习(每个智能体仅依据各自局部经验独立更新策略)，会导致：(1) 训练非平稳性：每个智能体的行为会影响其他智能体的观测与决策，导致环境分布不断变化，影响模型收敛性。(2) 经验不足：单个智能体收集的经验样本有限，难以支持有效策略学习。为解决上述问题，本文引入网络参数和经验共享机制，以增强训练稳定性并加快收敛速度。

在多个智能体 $Agent_i$ 的基础上，抽象出一个中心智能体 $Agent_0$，它不直接参与路由决策，而是用于更新路由策略。每个时间步，普通智能体 $Agent_i$ 与环境交互，将经验存入全局经验回放池 $D$，但不执行策略更新。每经过 $C_1$ 步，$Agent_0$ 从回放池 $D$ 随机抽取样本用于更新在线网络参数 $\theta$。每经过 $C_2$ 步，$Agent_0$ 更新目标网络参数 $\theta'$，再将网络参数同步至所有智能体 $Agent_i$。

该机制确保策略一致性，减少因策略差异导致的训练不稳定问题，同时通过经验共享整合全局数据，

弥补单个智能体经验不足的问题，加快收敛速度。

## 4. 测试和评估

### 4.1. 仿真环境设置

本文使用 Python 构建卫星网络路由仿真环境，以 Iridium 星座为仿真目标(轨道高度为 789 km，轨道倾角为 86.4˚，66 颗卫星均匀分布在 6 条轨道上)。相关的仿真参数(通信链路和模型训练参数)如表 1 所示。为了模拟卫星网络流量负载不均现象，参考文献[11]中的全球流量分布表，从全球各大洲选取了 50 个地面站来持续生成数据包用于仿真。

**Table 1.** Simulation parameter settings
**表 1.** 仿真参数设置

| 链路参数 | 值 |
|---|---|
| 载波频率 | 26 GHz |
| 信道带宽 | 10 MHz |
| 发射天线功率 | 10 W |
| 发射、接收天线增益 | 34 db, 34 db |
| 接收队列最大长度 | 100 |
| 数据包大小 | 1 KB |
| 训练轮次 | 1000 |
| 学习率 | 0.001 |
| 折扣因子 | 0.99 |
| 探索率、探索率衰减率 | 0.95~0.001, 0.995 |
| 抽样批量大小 | 64 |
| 经验回放池大小 | 10000 |
| 在线网络学习频率 | 4 |
| 目标网络更新频率 | 200 |

### 4.2. 对比算法

为验证所提路由算法的性能，本文选取了以下几种算法进行对比：

(1) SPF：最短路径优先算法，直接计算当前卫星和目标卫星之间的最短传播时延路径。

(2) ELB [11]：显式负载均衡路由算法，通过在相邻卫星节点间交换链路状态信息来缓解节点拥塞问题。

(3) DQN-IR [8]：基于 DQN 的分布式路由算法，可根据周围卫星的空间位置和排队时延等信息自适应选择下一跳节点。

### 4.3. 实验结果分析

本文主要从以下三个方面评估路由算法的性能：

(1) 平均端到端时延：所有数据包从源节点传输到目标节点所经历时延的平均值。

(2) 丢包率：未能成功到达目标节点的丢失数据包占总发送数据包的比例。

(3) 吞吐量：单位时间内能够传输的最大数据量。

为模拟不同流量负载下卫星路由算法的性能，本文将地面站的流量生成速率的范围设置为 2.1 Mbps 至 3.5 Mbps，观测各算法在不同流量生成速率下的各项性能指标。

图 5 展示了四种路由算法在不同流量生成速率下的平均端到端时延变化。随着流量生成速率增加，

各算法的时延均呈上升趋势，但上升速率存在明显差异。其中，SPF 算法时延最高，ELB 次之，DQN-IR 表现较优，本文提出的 MAD3QN 算法表现最佳，这主要得益于 MAD3QN 算法在状态空间和奖励函数设计中综合考虑了节点间距离、目标趋近距离、节点负载、链路质量等因素，使其能够更有效地平衡各项链路性能指标，使算法更倾向于选择距离更短、负载更轻、链路质量更优的转发路径，最终实现端到端时延的优化。

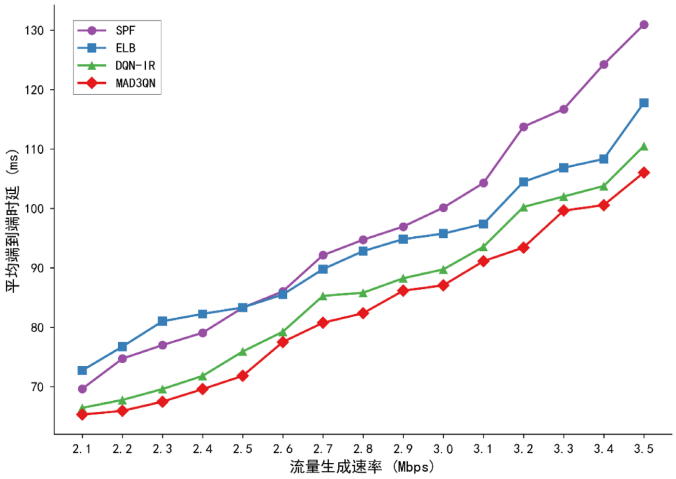


**Figure 5.** Comparison of average end-to-end delay for different algorithms
**图 5.** 不同算法的平均端到端时延对比图

图 6 展示了四种路由算法在不同流量生成速率下的丢包率变化。随着流量生成速率增加，所有算法的丢包率均有所上升。这是由于网络流量负载加重导致部分卫星节点接收队列达到上限，从而引发数据包丢弃。具体来看，SPF 算法丢包率最高，ELB 次之，DQN-IR 略优于 ELB，而 MAD3QN 在所有流量速率下均表现最佳。MAD3QN 算法不仅考虑了因链路拥塞导致的数据包丢失，还兼顾了链路质量问题，使得卫星在路由选择过程中能够根据实时链路状态自适应调整策略，从而有效避免拥塞链路和高误码链路的使用，显著降低了丢包率。

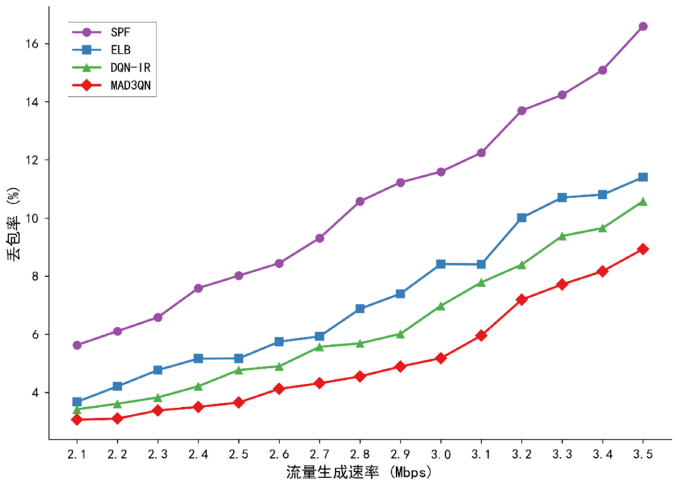


**Figure 6.** Comparison of average end-to-end delay for different algorithms
**图 6.** 不同算法的丢包率对比图

图 7 展示了四种路由算法在不同流量生成速率下的吞吐量变化，MAD3QN 算法在所有流量速率下均表现最佳。MAD3QN 算法的优势在于其能够动态感知网络状态变化(链路流量负载与链路质量)，并基于局部观测信息选择最优邻居链路进行数据转发。数据包优先选择更靠近目标节点、时延较低、负载较轻且误码率较低的链路，有效降低端到端时延与丢包率，提升系统吞吐量。

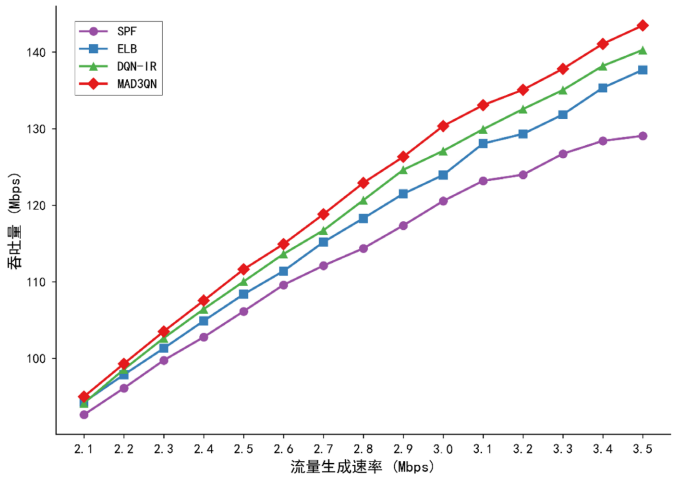


**Figure 7.** Comparison of average end-to-end delay for different algorithms
**图 7.** 不同算法的吞吐量对比图

## 5. 总结

本文提出了一种基于强化学习的低轨卫星网络动态分布式路由算法。首先系统性构建了涵盖网络拓扑、通信链路、通信时延和通信丢包等要素的卫星网络通信过程模型，之后提出了动态分布式路由架构，使卫星节点能够实时感知网络状态变化并自主决策。然后将卫星路由问题建模为分布式部分可观察马尔可夫决策过程(Dec-POMDP)，提出结合 Double DQN 和 Dueling DQN 的 MAD3QN 路由算法，设计了高效状态表示和有效奖励函数并引入了优先经验回放、网络参数和经验共享机制，以增强算法的稳定性与训练效率。仿真结果表明，MAD3QN 算法在端到端时延、丢包率和吞吐量方面均显著优于其他路由算法，验证了其对低轨卫星网络高动态环境的适应性与性能优势。在未来的研究中，将进一步综合考虑卫星能耗管理、故障检测与恢复等多个关键因素，以提升路由算法的适应性和实用性，为低轨卫星网络的高效运行提供更加全面的解决方案。

## 基金项目

本文由项目(D040304)资助。

## 参考文献

[1] Jiang, W. and Zong, P. (2011) A Discrete-Time Traffic and Topology Adaptive Routing Algorithm for LEO Satellite Networks. *International Journal of Communications, Network and System Sciences*, **4**, 42-52. https://doi.org/10.4236/ijcns.2011.41005

[2] Zhu, Y., Qian, L., Ding, L., Yang, F., Zhi, C. and Song, T. (2017) Software Defined Routing Algorithm in LEO Satellite Networks. 2017 *International Conference on Electrical Engineering and Informatics* (*ICELTICs*), Banda Aceh, 18-20 October 2017, 257-262. https://doi.org/10.1109/iceltics.2017.8253282

[3] Li, C., He, W., Yao, H., Mai, T., Wang, J. and Guo, S. (2023) Knowledge Graph Aided Network Representation and Routing Algorithm for LEO Satellite Networks. *IEEE Transactions on Vehicular Technology*, **72**, 5195-5207.

https://doi.org/10.1109/tvt.2022.3225666

[4] Pan, T., Huang, T., Li, X., Chen, Y., Xue, W. and Liu, Y. (2019) OPSPF: Orbit Prediction Shortest Path First Routing for Resilient LEO Satellite Networks. 2019 *IEEE International Conference on Communications* (*ICC*), Shanghai, 20-24 May 2019, 1-6. https://doi.org/10.1109/icc.2019.8761611

[5] Zhang, X., Yang, Y., Xu, M. and Luo, J. (2021) ASER: Scalable Distributed Routing Protocol for LEO Satellite Networks. 2021 *IEEE* 46*th Conference on Local Computer Networks* (*LCN*), Edmonton, 4-7 October 2021, 65-72. https://doi.org/10.1109/lcn52139.2021.9524989

[6] Yin, Y., Huang, C., Wu, D., Huang, S., Ashraf, M.W.A. and Guo, Q. (2021) Reinforcement Learning-Based Routing Algorithm in Satellite-Terrestrial Integrated Networks. *Wireless Communications and Mobile Computing*, **2021**, Article 3759631. https://doi.org/10.1155/2021/3759631

[7] Huang, Y., Wu, S., Kang, Z., Mu, Z., Huang, H., Wu, X., *et al*. (2023) Reinforcement Learning Based Dynamic Distributed Routing Scheme for Mega LEO Satellite Networks. *Chinese Journal of Aeronautics*, **36**, 284-291. https://doi.org/10.1016/j.cja.2022.06.021

[8] Zuo, P., Wang, C., Yao, Z., Hou, S. and Jiang, H. (2021) An Intelligent Routing Algorithm for LEO Satellites Based on Deep Reinforcement Learning. 2021 *IEEE* 94*th Vehicular Technology Conference* (*VTC*2021-*Fall*), Norman, 27-30 September 2021, 1-5. https://doi.org/10.1109/vtc2021-fall52928.2021.9625325

[9] Chen, Q., Giambene, G., Yang, L., Fan, C. and Chen, X. (2021) Analysis of Inter-Satellite Link Paths for LEO Mega-Constellation Networks. *IEEE Transactions on Vehicular Technology*, **70**, 2743-2755. https://doi.org/10.1109/tvt.2021.3058126

[10] Leyva-Mayorga, I., Soret, B., Matthiesen, B., Röper, M., Wübben, D., Dekorsy, A., *et al*. (2022) Non-Geostationary Orbit Constellation Design for Global Connectivity. In: Leyva-Mayorga, I., Soret, B., *et al*., Eds., *Non-Geostationary Satellite Communications Systems*, Institution of Engineering and Technology, 237-267. https://doi.org/10.1049/pbte105e_ch10

[11] Taleb, T., Mashimo, D., Jamalipour, A., Kato, N. and Nemoto, Y. (2009) Explicit Load Balancing Technique for NGEO Satellite IP Networks with On-Board Processing Capabilities. *IEEE/ACM Transactions on Networking*, **17**, 281-293. https://doi.org/10.1109/tnet.2008.918084