

DOI:10.11918/202112138

高 QoE 的低时延智能网络数据传输调度算法

张慎文,许崇海,胡天乐,陶霜霜,李鲁群

(上海师范大学 信息与机电工程学院,上海 201418)

摘要: 面向低时延、稳定传输、高用户体验质量 (quality of experience, QoE) 的网络实时传输需求场景,提出一种低时延智能网络数据传输调度算法。该算法由数据块排队控制策略和拥塞控制策略两部分组成。数据排队控制策略提出了综合数据块的创建时间和有效时限 (effective time) 的性价比模型,有效地解决了传输时间约束下的信息传输不均衡问题;拥塞控制策略提出了基于使用耿贝尔分布 (Gumbel distribution) 采样重参数化与混合经验优先级模型改进后的深度确定性策略梯度 (deep deterministic policy gradient, DDPG) 方法,解决了深度确定性策略梯度不适用于离散网络动作空间拥塞控制的问题,并通过学习自适应调整发送参数显著提升了网络拥塞控制质量。实验结果表明,实时传输场景下使用本文提出的排队算法能够有效提升 QoE,采用改进后的 DDPG 进行拥塞控制能大幅降低传输时延。同样场景下,将提出的智能网络数据传输调度算法与排队策略及拥塞控制策略相结合,与传统的网络数据传输调度算法相比,能够更好地兼顾低时延和稳定传输,提供更高的数据传输质量。

关键词: 拥塞控制;低时延;排队算法;深度确定性策略梯度;体验质量 (QoE)

中图分类号: TP393 **文献标志码:** A **文章编号:** 0367-6234(2023)05-0132-07

Low-latency intelligent network data transmission scheduling algorithm with high QoE

ZHANG Shenwen, XU Chonghai, HU Tianle, TAO Shuangshuang, LI Luqun

(The College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 201418, China)

Abstract: This paper proposes a low-latency intelligent network data transmission scheduling algorithm for real-time network transmission demand scenarios of low latency, stable transmission, and high quality of experience (QoE). The algorithm consists of two parts: data block queuing control strategy and congestion control strategy. The data block queuing control strategy presents a cost-effective model that integrates the creation time and effective time of data blocks, effectively solving the problem of uneven information transmission under transmission time constraint. The congestion control strategy proposes a deep deterministic policy gradient (DDPG) method based on the Gumbel distribution sampling reparameterization with mixed experience prioritization model, which solves the problem that DDPG is not applicable to the congestion control of discrete network action space and significantly improves the quality of network congestion control by adaptively adjusting the sending parameters through learning. Results show that the proposed queuing algorithm could effectively improve QoE in real-time transmission scenarios, and the improved DDPG for congestion control could significantly reduce transmission delay. In the same scenario, compared with traditional network data transmission scheduling algorithms, by integrating the proposed queuing and congestion control strategies, the improved intelligent network data transmission scheduling algorithm could maintain a good balance between low latency and stable transmission and provide higher data transmission quality.

Keywords: congestion control; low latency; queuing algorithm; deep deterministic policy gradient; quality of experience (QoE)

自动驾驶、物联网、视频会议等实时应用场景要求极低的网络时延和高传输稳定性。网络传输中数据传输调度算法对网络时延、用户体验质量 (quality of experience, QoE) 有很大影响,受限于物理介质,

数据传输调度算法主要通过优化排队方法和发送策略来降低网络时延和提高体验质量。本文提出了智能的排队方法和发送策略,该策略可以满足低时延、高体验质量、稳定传输的需求。

传统的数据传输调度算法在排队方法上未考虑数据块的优先级和数据块的有效时限 (effective time, ET)。针对排队策略, 文献[1]采用滑模变结构设计一种队列管理方法, 在提高链路利用率的同时保证了 QoE, 但未考虑到不同数据块优先级和有效时限的差异。文献[2]通过数据块的有效时限设计实时调度器, 降低找到最优数据块的时间, 但此方法不适用于不同优先级的数据块调度。文献[3]提出基于数据优先级的调度算法, 能有效提高带宽利用率, 但单一考虑优先级可能会导致低优先级数据块难以被选中而错失有效期限, 产生信息传输不均衡问题。

发送策略通过拥塞控制实现, 经典的拥塞控制方法如 Reno 算法^[4] 在达到稳定状态后, 依然会有时延波动; 文献[5]提出一种基于往返时延的拥塞控制算法, 提高吞吐量的同时降低了时延, 但未考虑到 QoE, 且无法在链路吞吐量一定的情况下进行最优调整; 文献[6]提出强化学习解决互联网拥塞控制的视角, 也指出强化学习拥塞控制依然存在安全性、泛化性等一系列挑战; 文献[7]提出基于损失预测器 (loss predictor, LP) 监督学习拥塞控制方法 (loss predictor-transmission control protocol, LP-TCP) 和基于强化学习 SARSA 算法^[8] 的拥塞控制方法 (reinforcement learning-transmission control protocol, RL-TCP)。LP-TCP 能够有效权衡吞吐量和利用率, 但降低时延的效果并不理想, RL-TCP 能够有效降低时延, 提高链路利用率, 但 SARSA 为同策略的算

法^[9], 学习数据不全面, 不一定能找到最优策略, 采用回合更新神经网络参数, 学习过程不够稳定。近年来兴起深度强化学习^[10] 拥塞控制方法, 文献[11] 基于深度 Q 网络 (deep Q network, DQN) 改进经验回放模块, 降低了网络时延, 提升了体验质量, 但 DQN 采用硬更新目标网络参数, 每次更新将网络参数整体替换, 训练过程不够稳定; 文献[12] 使用深度强化学习中异策略的深度确定性策略梯度算法 (deep deterministic policy gradient, DDPG)^[13] 进行拥塞控制, 能够自行根据环境调整发送参数, 使用软更新目标网络参数, 训练过程稳定, 但 DDPG 算法常用来处理连续动作控制, 不适用于离散动作控制, 且未充分考虑到采样样本的不均衡性可能导致的过拟合问题。

针对上述问题, 提出了一种面向体验质量的低时延数据传输调度算法, 将数据块的优先级与数据块有效时限相结合, 提出数据块性价比的模型改进排队模块, 利用 Gumbel-Softmax 重参数方法^[14] 改进 DDPG, 使 DDPG 适用于离散控制, 将动作奖励与时间差分偏差综合改进 DDPG 的经验回放模块, 得到快速收敛、时延低且稳定的数据传输调度算法。

1 本文数据传输调度算法框架

为了满足实时场景下低时延、稳定传输的要求, 笔者提出的数据传输调度算法包括两部分, 数据块排队模块和深度强化学习拥塞控制模块, 算法框架见图 1。

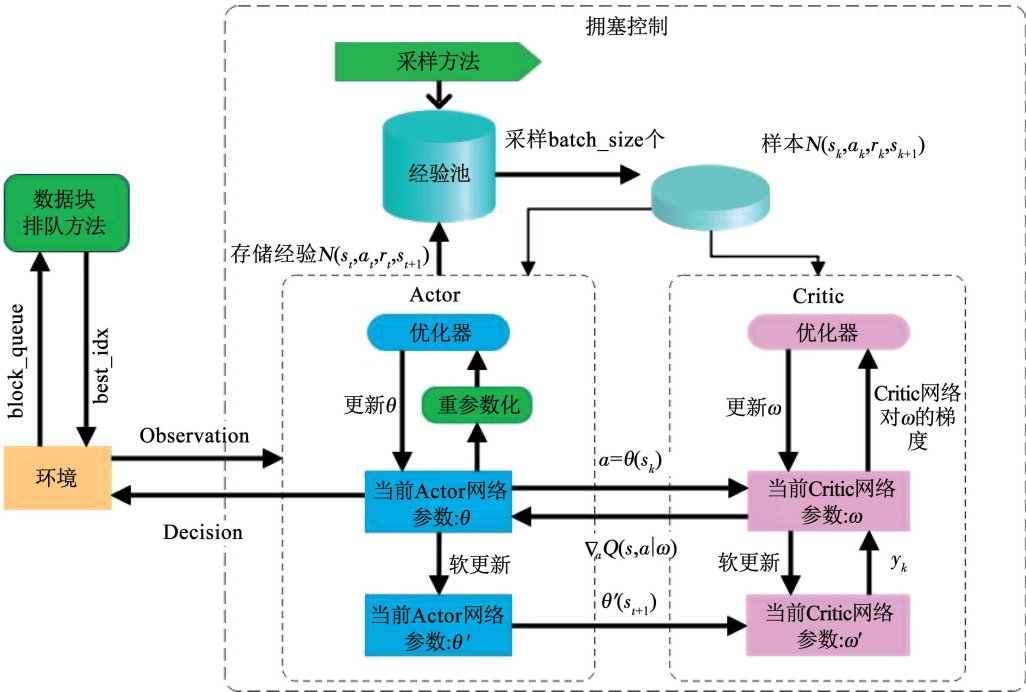


图 1 本文数据传输调度算法框架

Fig. 1 Framework of proposed data transmission scheduling algorithm

本算法分为两部分:排队策略模块和拥塞控制策略模块。数据块排队模块综合数据块优先级和有效时限 (effective time), 每次调用会返回排队结果, 即性价比最高的数据块索引。拥塞控制模块为 DDPG 算法加入 Gumbel-Softmax 重参数化和改进的优先经验回放采样方法, 经验池存储的经验样式为 $N(s_t, a_t, r_t, s_{t+1})$, s_t 为当前环境状态, a_t 为当前环境下做出的动作, r_t 为当前环境下做出 a_t 的动作奖励, s_{t+1} 为估计的下一个环境状态, 通过采样的经验样本进行学习, 实时更新参数, 调整发送速率。

DDPG 为 Actor-Critic 结构^[15], 包含 Actor 网络和 Critic 网络, 网络参数分别为 θ 和 ω 。Actor 网络负责输出动作 a 。Critic 网络对 Actor 输出的动作 a 做评估 $Q(s, a)$, 估计出动作 a 的动作未来能有多少收益。Critic 网络根据环境的反馈来调整网络的参数 ω , Actor 网络使用策略梯度反向传播优化网络参

数 θ , 优化方程为^[16]

$$\nabla_{\theta} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a \mid \omega) \bigg|_{s=s_i, a=R(s_i)} \nabla_{\theta} R(s \mid \theta) \bigg|_{s=s_i} \quad (1)$$

式中: $\nabla_a Q(s, a \mid \omega) \big|_{s=s_i, a=R(s_i)}$ 为当前 Critic 网络输出的动作得分 $Q(s, a \mid \omega)$ 对动作 a 的梯度; $\nabla_{\theta} R(s \mid \theta) \big|_{s=s_i}$ 为当前 Actor 网络输出的当前状态 s_i 下动作策略 R 对其参数 θ 的梯度。

1.1 数据块排队

在实时通信 (real-time communication, RTC) 中传输的数据类型多样, 有不同的有效时限和优先级, 如视频电话场景的 3 种不同类型数据块如图 2 所示, 包含控制信号 control、音频 audio 和视频 video, 3 种类型数据块有各自的有效时限和优先级, 控制信号的优先级最高, 音频的优先级高于视频。

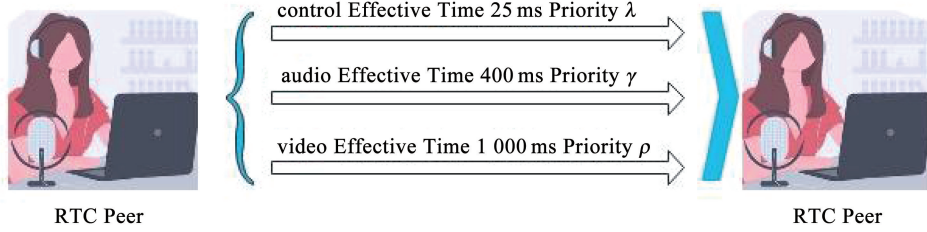


图 2 不同类型数据块传输过程

Fig. 2 Process of transferring different types of data blocks

图 2 中控制信号、音频和视频数据块的优先级 λ, γ, ρ 为互不相等的正整数, 值越小优先级越高。对创建时间小于当前时间的数据块排队处理, 计算数据块的剩余有效时限, 计算式为

$$T_{\text{rem}} = d - (T_{\text{cur}} - T_{\text{cre}}) \quad (2)$$

式中: d 为数据块有效时限 (effective time); T_{cur} 为当前时间; T_{cre} 为数据块创建时间; T_{rem} 为数据块剩余有效时限, $T_{\text{rem}} < 0$ 代表数据块失效, $T_{\text{rem}} > 0$ 代表数据块仍有效。

计算此数据块性价比, 建立的数据块性价比模型为

$$\left(\frac{T_{\text{rem}}}{d} - 1 \right)^2 + \left(\frac{Q_{\text{cp}}(\lambda + \gamma + \rho)}{\sigma} \right)^2 = 1 \quad (3)$$

式中: $\frac{T_{\text{rem}}}{d}$ 为数据块剩余有效时限比例, 值域为 $(0, 1)$, 数值越小越急迫; σ 为当前数据块优先级数值, $\frac{\sigma}{(\lambda + \gamma + \rho)}$ 为数据块优先程度, 取值为 $(0, 1)$ 范围的离散值, 数值越小优先级越高; Q_{cp} 为数据块的性价比, 数值越小性价比越高, 根据 Q_{cp} 更新数据块队列, 每次调用返回最佳性价比数据块索引。

使用椭圆形式建立数据块性价比模型, $\frac{T_{\text{rem}}}{d}$ 与 Q_{cp} 为椭圆性价比模型参数, 长半轴长为 1, 将离散值的 $\frac{\sigma}{(\lambda + \gamma + \rho)}$ 作为椭圆模型的短半轴长, 离散值数等于数据块类型数。本模型合理性在于, 只需要看椭圆模型在第一象限 $\frac{T_{\text{rem}}}{d}$ 为 $(0, 1)$ 的部分, 当数据块 $\frac{T_{\text{rem}}}{d}$ 相同时, 优先程度 $\frac{\sigma}{(\lambda + \gamma + \rho)}$ 越小, Q_{cp} 越小, 即性价比越高; 当 $\frac{\sigma}{(\lambda + \gamma + \rho)}$ 相同, $\frac{T_{\text{rem}}}{d}$ 越小, Q_{cp} 越小, 同时椭圆性价比模型避免了 $\frac{\sigma}{(\lambda + \gamma + \rho)}$ 与 $\frac{T_{\text{rem}}}{d}$ 两部分变化幅度不一致问题。

1.2 Gumbel-Softmax 重参数

为了让 DDPG 输出离散动作, 将 Actor 网络的输出层激活函数设为 softmax, 把向量归一化的同时输出动作概率分布 $\mathbf{a}(a_1, a_2, \dots, a_m)$, 用 argmax 选择动作不可导无法进行梯度回传, 为了选择确定性动作, 改进使用 Gumbel-Softmax 方法, 标准 Gumbel 分布的累积密度函数见下式^[17]

$$F(x) = e^{-e^{-x}} \quad (4)$$

通过对 Gumbel 分布求逆得到 g_i ,公式如下^[17]:

$$\begin{cases} g_i = -\log(-\log u_i) \\ u_i \sim U[0,1] \end{cases} \quad (5)$$

式中 u_i 为服从 $0 \sim 1$ 均匀分布的随机变量样本, g_i 为服从 Gumbel 分布的随机变量样本。

为动作概率向量 $\mathbf{a}$ 中的动作概率 a_i ,取对数 $\log a_i$,对 $\log a_i$ 添加 g_i ,相当于在选择动作之前加入噪声提高探索度,Actor 网络通过梯度回传更新参数,不能用不可导的 $\arg\max$ 选择采样值最大的动作,神经网络中处理离散输入的基本方法是转为 one-hot 形式,使用 softmax 对 $\text{onehot}(\arg\max)$ 光滑近似^[17],改进后为

$$\text{softmax}((\log a_i + g_i)/\tau)_{i=1}^m \quad (6)$$

式中: $\log a_i$ 为动作概率 a_i 的对数; m 是离散动作类别数; τ 是大于0的退火参数,原始的 softmax 直接进行退火只能得到最大值位置为1的 one-hot 向量,Gumbel-Softmax 通过降低 τ 的值来逼近 one-hot 形式,得到有概率取非最大值位置的 one-hot 向量,相当于对动作选择增加探索度。

Gumbel-Softmax 通过近似离散概率分布,提供了一种重参数化的方法,本算法中用来对动作概率集合进行采样以选择动作,使得模型能够进行端到端的训练。通过 Gumbel-Softmax 方法,在输出离散动作的同时保留梯度使得 Actor 网络能够更新。深度强化学习通过获取经验学习的方式,根据网络传输环境自适应调整发送参数,能够有效提升链路实时吞吐量,降低时延,且经过学习后传输时延稳定在较低的水平。

1.3 混合经验排序优先经验回放

在经验回放模块对经验进行采样,传统的经验回放方法通过计算时间差分偏差(temporal difference error)来衡量每个经验的学习价值作为经验采样依据,时间差分偏差计算公式为^[18]

$$T_d = r_{t+1} + \phi \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \quad (7)$$

式中: $Q(s_t, a_t)$ 为环境 s_t 下采取 a_t 动作实际得分,为折扣系数; $r_{t+1} + \phi \cdot Q(s_{t+1}, a_{t+1})$ 为事前估计得分; T_d 为实际得分与估计得分差值, T_d 越大,经验的学习价值越大。

本文提出的混合经验排序优先经验回放将传统方法中时间差分偏差与动作奖励结合,优化经验采样方法,本算法中动作奖励 r_t 与发送速率正相关,为保证样本选择的均衡性,分别建立基于时间差分偏差和基于动作奖励 r_t 对经验的优先级模型,计算公式如下:

$$\begin{cases} P_t(i) = |T_d| + \mu \\ P_k(i) = |r_t| + \mu \end{cases} \quad (8)$$

式中: μ 为一个极小的正常数,保证优先级不为0; $P_t(i)$ 为经验 i 基于时间差分偏差的优先级; $P_k(i)$ 为经验 i 基于动作奖励 r_t 的优先级。 $P_t(i)$ 、 $P_k(i)$ 不能保证是同一数量级,为了在统一标准下衡量两种优先级,选择重构优先级模型,将经验 i 按照 $P_t(i)$ 、 $P_k(i)$ 两种优先级值从大到小排序得到两种排名 $r_t(i)$ 与 $r_k(i)$,根据排名结果建立新的优先级模型:

$$\begin{cases} P'_t(i) = \frac{1}{r_t(i)} \\ P'_k(i) = \frac{1}{r_k(i)} \end{cases} \quad (9)$$

式中: $P'_t(i)$ 和 $P'_k(i)$ 分别为根据 $r_t(i)$ 与 $r_k(i)$ 取倒数构建的基于时间差分偏差和基于动作奖励 r_t 对经验的新优先级模型, $P'_t(i)$ 和 $P'_k(i)$ 取值范围相同,通过 $P'_t(i)$ 和 $P'_k(i)$ 构建经验的混合优先度模型

$$p_i = \alpha \cdot P'_t(i) + \beta \cdot P'_k(i) \quad (10)$$

式中: p_i 为经验 i 的混合优先度; α 、 β 是超参数,分别为两种优先级的混合权重,表征的是构建的两种优先级对于混合优先度的贡献度,取值区间都为 $[0, 1]$,为方便理解和计算,将 α 、 β 归一化输入,权重参数和为100%。

通过混合优先度进行经验采样,经验 i 采样概率公式为

$$F_i = \frac{p_i^\varepsilon}{\sum_{c=1}^h p_i^\varepsilon} \quad (11)$$

式中: F_i 为经验 i 的采样概率; ε 为 $[0, 1]$ 的实数,代表混合优先度的使用程度, ε 越大混合优先级使用的越多, ε 为0时为均匀采样; h 为经验池中的经验数量。

混合经验排序优先经验回放将时间差分偏差与动作奖励机制结合进行优先级排序,解决了样本不均衡问题导致的过拟合问题,并且更稳健,对异常值不敏感。动作奖励在本场景下与发送速率正相关,加入动作奖励的经验优先级优化经验采样方法,在经过学习后,能够进一步提升发送速率,降低时延,增加带宽利用率。

2 实验及结果

2.1 实验环境及数据集

采用的实验环境和数据集来自 ACM Multimedia 2021 Grand Challenge 系列由清华大学主办 Meet Deadline Requirements 公开挑战赛,实验环境基于确定性时延传输协议(Deadline-Aware Transport Protocol, DTP)协议^[19]。DTP 协议由清华大学 2019

年提出,基于传输层 QUIC 协议(Quick UDP Internet Connection,QUIC)^[20],DTP 协议支持数据块形式传输,并为数据块添加有效时限,数据集使用挑战赛的公开数据集。

数据集的数据块类型包含控制信号、音频和视频。以音频为例,数据集样式见表 1。

表 1 数据集样式

数据块创建时间/s	数据块大小/byte
0	106 650
0.041 708	11 430
...	...

数据集包含数据块创建时间和数据块大小,优先级与有效时限与数据块类型有关,同一种类型数据块优先级与有效时限相同。

实验环境使用的用户体验质量(QoE)评价模型为

$$Q_e = \sum_{i=1}^N P_i \cdot M_i - \sum_{i=1}^N P_i \cdot (1 - M_i) \quad (12)$$

式中: P_i 为数据块的优先级; M_i 为该数据块错过有效时限标志量,值为 0 代表错过有效时限,值为 1 代表未错过有效时限; Q_e 为体验质量,是所有未错过有效时限数据块优先级之和与所有错过有效时限数据块优先级之和的差值。

2.2 实验方法

本文采用对比消融实验法进行实验,对不同的排队算法和拥塞控制算法与本文提出算法进行对比实验,对单个算法中的多处改进,进行消融实验,分析得到改进效果。

2.2.1 排队与拥塞控制

对比消融实验所用排队算法与拥塞控制算法见表 2。

表 2 用到的排队算法和拥塞控制算法

Tab.2 Queuing and congestion control algorithms used in this paper

排队算法	拥塞控制算法
对比排队算法	Reno 算法
本文排队算法	GS-DDPG 算法
	PER-GS-DDPG 算法

表 3 对比排队算法 + 不同拥塞控制算法

Tab.3 Contrast queuing algorithm + different congestion control algorithms							
排队模块	拥塞控制模块	Q_e	Q_{avg}/s	Q_{max}/s	L_{avg}/s	L_{sta}/s	V_{avg}
	Reno 算法	442	0.003 250	0.041 250	0.022 780	—	—
对比排队算法	GS-DDPG 算法	521	0.000 850	0.041 250	0.012 452	0.008 000	1 050
	PER-GS-DDPG 算法	536	0.000 850	0.041 250	0.006 084	0.002 000	1 300

将表 2 中 2 种数据块排队算法与 3 种拥塞控制算法组合为不同的数据传输调度算法,进行对比消融实验。对实验结果进行性能分析,得到高 QoE、低时延、稳定传输的网络数据传输调度算法。

2.2.2 对比排队算法

对比排队算法使用的是挑战赛官方 Demo 中给出的排队算法,具体算法如下:

1) 优先按照数据块创建时间(T_{cur})进行排队,值越小数据块越早创建,优先选择创建时间早的数据块;

2) 创建时间相同则按照紧急程度进行排队,计算公式为

$$H = (T_{cur} - T_{cre}) \cdot d \quad (13)$$

式中: d 为数据块有效时限, T_{cur} 为当前时间, T_{cre} 为数据块创建时间,创建时间相同则($T_{cur} - T_{cre}$)相同且恒正,有效时间 d 越长越不紧急。由上可知,紧急程度 H 越小,数据块越紧急。

对比排队算法只考虑了数据块有效时限,未考虑到数据块不同优先级的差异对 QoE 和时延的影响。

2.3 对比消融实验

2.3.1 实验参数设置

DDPG 中 Actor 与 Critic 的网络结构都为 3 个全连接层,Actor 网络每层神经元数量分别为 300、200、3,Critic 网络全连接层的神经元数量分别为 400、300、1。Actor 网络与 Critic 网络的学习率分别为 0.001 和 0.005,目标网络的软更新系数 φ 0.01,折扣系数为 0.9,经验池大小为 10 000,经验回放采样的样本数为 32,混合权重超参数 α 、 β 设为 0.6 与 0.4。

2.3.2 实验结果

将对比排队算法和本文排队算法,Reno 算法与本文提出的 2 种拥塞控制算法进行对比传输实验,对拥塞控制算法的多处改进进行消融实验。在实验环境中实验后输出传输日志,不同排队算法与多种拥塞控制算法组合的数据传输调度算法实验结果分别见表 3、4。

表 4 本文排队算法 + 不同拥塞控制算法

Tab. 4 Proposed queuing algorithm + different congestion control algorithms

排队模块	拥塞控制模块	Q_e	Q_{avg}/s	Q_{max}/s	L_{avg}/s	L_{sta}/s	V_{avg}
本文排队算法	Reno 算法	470	0.003 000	0.041 250	0.021 982	—	—
	GS-DDPG 算法	565	0.000 750	0.041 250	0.010 154	0.008 000	1 250
	PER-GS-DDPG 算法	581	0.000 750	0.041 250	0.004 598	0.002 000	1 520

表 3 中排队模块采用对比排队算法,拥塞控制模块分别采用 Reno、GS-DDPG、PER-GS-DDPG 算法。拥塞控制模块使用 PER-GS-DDPG 算法和 GS-DDPG 算法用户体验质量 (Q_e)、平均传输速率 (V_{avg})、平均传输时延 (L_{avg}) 远优于 Reno 算法。其中 PER-GS-DDPG 算法最优,学习效果大大提升,在 QoE (Q_e)、平均传输速率 (V_{avg})、平均传输时延 (L_{avg}) 都优于 GS-DDPG 算法,尤其是稳定传输时延 (L_{sta}) 比 GS-DDPG 降低了 75%。可得,数据调度算法排队模块采用对比排队算法时,拥塞控制模块使用 PER-GS-DDPG 算法优于 Reno 算法和 GS-DDPG 算法。同上,由表 4 可以看出,排队模块采用本文排队算法时,拥塞控制模块采用 PER-GS-DDPG 算法最佳。

将表 4 与表 3 对比可见,排队模块采用本文排队算法优于对比排队算法,本文排队算法结合数据块优先级和有效时限保证平衡性的情况下进行排队优化,QoE (Q_e) 有较大提升,平均排队时延 (Q_{avg}) 和平均传输时延 (L_{avg}) 略有下降,平均传输速率 (V_{avg}) 也有提升。

为便于比较不同排队模块和拥塞控制模块的组合的数据传输调度算法性能,建立以网络传输时延 (Latency) 为纵坐标,传输时间 (Time) 为横坐标的变化曲线见图 3。图 3 (a) ~ (b) 分别为对比排队算法、本文排队算法与多种拥塞控制算法结合为数据调度算法进行消融对比实验的网络传输时延,反映了不同组合的数据传输调度算法的传输效果。

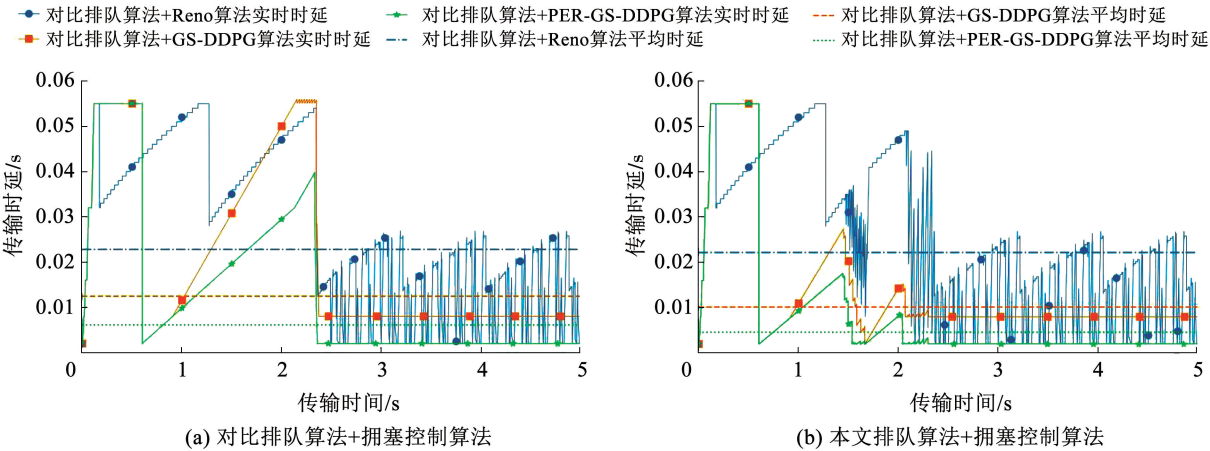


图 3 不同排队算法和拥塞控制算法对网络时延的影响

Fig. 3 Impact of different queuing algorithms and congestion control algorithms on network latency

根据图 3 (a) ~ (b) 可以看出,无论使用哪种拥塞控制算法,在 1.5 s ~ 2.3 s 之间,数据调度算法的排队模块使用本文排队算法的实时时延在波动中降低,相同时间内使用对比排队算法的实时时延一直在增加。

无论排队模块采用对比排队算法还是本文排队算法,数据调度算法的拥塞控制模块使用 PER-GS-DDPG 的平均传输时延 < GS-DDPG < Reno。同时, Reno 算法在达到稳定传输后时延仍在一定范围内波动,PER-GS-DDPG 算法和 GS-DDPG 算法通过强化学习的方式自适应调整发送参数,达到稳定传输状态后时延稳定,且优化了经验回放采样方法的 PER-GS-DDPG 算法拥塞控制的稳定传输时延能达

到 0.002 s,能够很好地进行低时延传输。 综上可得,本文数据传输调度算法排队模块采用本文提出的数据块性价比模型,拥塞控制模块采用 PER-GS-DDPG 算法能够很好地保证高 QoE、低时延、稳定的网络传输。

3 结 论

为了解决实时通信场景下用户体验不佳、高时延、传输不稳定等问题,提出了一种改进数据传输调度算法。该算法通过优化排队和拥塞控制模块,提出了一种数据块性价比模型和 PER-GS-DDPG 拥塞控制算法。经过实验验证,主要结论如下:

1) 排队模块提出的本文排队算法,通过建立数

据块性价比模型,能较大幅度提高 QoE,一定程度上降低平均传输时延。

2) 拥塞控制模块提出的 PER-GS-DDPG 算法,通过学习网络链路环境自适应调整发送参数,能够在大幅度降低传输时延的同时保持传输稳定性。

3) 实验证明,以本文排队算法 + PER-GS-DDPG 算法为主体的智能网络数据传输调度算法,能大幅提高 QoE,对降低网络时延有很好的效果,稳定后的网络时延在 0.002 s,能够满足自动驾驶、物联网、视频会议等实时通信场景的高 QoE、低时延、稳定传输等要求。

参考文献

- [1] 尹逊和, 任丰原, 任勇, 等. 鲁棒的主动队列管理新算法[J]. 计算机学报, 2002, 25(10): 1018
YIN Xunhe, REN Fengyuan, REN Yong, et al. A novel robust algorithm for active queue management [J]. Chinese Journal of Computers, 2002, 25(10): 1018. DOI:10.3321/j.issn:0254-4164.2002.10.002.
- [2] YANG Lei, SAGDUYU Y E, ZHANG Junshan, et al. Deadline-aware scheduling with adaptive network coding for real-time traffic [J]. IEEE/ACM Transactions on Networking, 2014, 23(5): 1430. DOI: 10.1109/TNET.2014.2331018
- [3] 任浩, 王劲林, 尤佳莉. 一种高效的对等网络流媒体数据调度算法[J]. 西安交通大学学报, 2011, 45(6): 20
REN Hao, WANG Jinlin, YOU Jiali. An efficient data scheduling algorithm for peer-to-peer streaming system [J]. Journal of Xi'an Jiaotong University, 2011, 45(6): 20
- [4] 章淼, 吴建平, 林闯. 互联网端到端拥塞控制研究综述[J]. 软件学报, 2002, 13(3): 354
ZHANG Miao, WU Jianping, LIN Chuang. Overview of Internet end-to-end congestion control research [J]. Journal of Software, 2002, 13(3): 354
- [5] MITTAL R, LAM V T, DUKKIPATI N, et al. TIMELY: RTT-based congestion control for the datacenter[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 537. DOI: 10.1145/2829988.2787510
- [6] JAY N, ROTMAN N, GODFREY B, et al. A deep reinforcement learning perspective on internet congestion control[C]//International Conference on Machine Learning. [S.l.]: PMLR, 2019: 3050
- [7] KONG Yiming, HUI Zang, MA Xiaoli. Improving TCP congestion control with machine intelligence [C]//Proceedings of the 2018 Workshop on Network Meets AI & ML. Budapest: ACM, 2018: 60
- [8] THAM C K. Modular on-line function approximation for scaling up reinforcement learning [D]. Cambridge: University of Cambridge, 1994
- [9] SUTTON R S, BARTO A G. Reinforcement learning: an introduction [M]. Cambridge: MIT press, 2018: 129
- [10] LUONG N C, HOANG D T, GONG S, et al. Applications of deep reinforcement learning in communications and networking: a survey [J]. IEEE Communications Surveys & Tutorials, 2019, 21(4): 3133. DOI: 10.1109/COMST.2019.2916583
- [11] 常梦磊, 罗述翔, 李幸睿, 等. 低时延传输的 ERDQN 数据调度算法[J]. 哈尔滨工业大学学报, 2021, 53(8): 132
CHANG Menglei, LUO Shuxiang, LI Xingrui, et al. ERDQN data scheduling algorithm for low latency transmission [J]. Journal of Harbin Institute of Technology, 2021, 53(8): 132. DOI: 10.11918/202011098
- [12] 买天乐. 基于深度强化学习的路由调度机制研究 [D]. 北京: 北京工业大学, 2019
MAI Tianle. Research on routing scheduling mechanism based on deep reinforcement learning [D]. Beijing: Beijing University of Technology, 2019
- [13] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning [Z]. arXiv:1509.02971, 2015. DOI:10.48550/arXiv.1509.02971
- [14] STRYPSTEEN T, BERTRAND A. End-to-end learnable EEG channel selection for deep neural networks with Gumbel-Softmax [J]. Journal of Neural Engineering, 2021, 18(4): 0460a9. DOI: 10.1088/1741-2552/ac115d
- [15] KONDA V R, TSITSIKLIS J N. Actor-critic algorithms [C]//Advances in Neural Information Processing Systems. Cambridge: Massachusetts Institute of Technology, 2000: 1008
- [16] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms [C]//Proceedings of the 31st International Conference on Machine Learning. [S.l.]: PMLR, 2014: 387
- [17] JANG E, GU S, POOLE B. Categorical reparameterization with Gumbel-Softmax [Z]. arXiv:1611.01144, 2016. DOI:10.48550/arXiv.1611.01144
- [18] BARTO A G. Temporal difference learning [J]. Scholarpedia, 2007, 2(11): 1604. DOI: 10.1007/978-0-387-30164-8_817
- [19] SHI Hang, CUI Yong, QIAN Feng, et al. DTP: Deadline-aware transport protocol [C]//Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019. Beijing: ACM, 2019: 1
- [20] CUI Yong, LI Tianxiang, LIU Cong, et al. Innovating transport with QUIC: Design approaches and research challenges [J]. IEEE Internet Computing, 2017, 21(2): 72. DOI: 10.1109/MIC.2017.44

(编辑 苗秀芝)