

DOI:10.11918/202011098

低时延传输的 ERDQN 数据调度算法

常梦磊, 罗述翔, 李幸睿, 李鲁群

(上海师范大学 信息与机电工程学院, 上海 201418)

摘要: 针对车载网络、远程医疗、工业控制等领域需要低时延、高可靠性的网络传输应用场景, 提出了一种经验回放的 DQN (experience replay DQN, ERDQN) 数据传输调度算法。该算法的主要目的和任务是降低网络时延和提高网络传输的稳定性。ERDQN 算法在最后期限感知的传输协议 (deadline-aware transport protocol, DTP) 的基础上优化了发送端的排队策略, 充分考虑了数据块的优先级和截止日期 (Deadline), 将其作为计算进入等待队列顺序的重要因素, 解决了数据块丢失 Deadline 的问题, 降低了网络传输的排队延迟; 同时在拥塞控制方面以当前时刻网络传输状态为特征向量, 预测下一时刻网络传输状态参数, 并赋予不同的奖励因子进行评估, 通过 ERDQN 网络的迭代学习, 自动调整到适合当前网络传输的最优参数, 在后续的网络链路传输过程中, 平均传输速率高且稳定, 缓解了网络拥塞和传输不稳定的问题, 降低了网络传输时延。实验结果表明 ERDQN 算法的平均排队时延和传输时延远远低于传统拥塞控制算法 (Reno 算法), 在质量系数 (quality of experience, QoE) 方面远远高于传统的拥塞控制算法, 能够最大程度减少网络传输速率波动、降低丢包率, 提供稳定可靠的传输。

关键词: 拥塞控制; 传输速率; 低时延; QoE; Reno

中图分类号: TP393

文献标志码: A

文章编号: 0367-6234(2021)08-0132-05

ERDQN data scheduling algorithm for low latency transmission

CHANG Menglei, LUO Shuxiang, LI Xingrui, LI Luqun

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

Abstract: An experience replay DQN (ERDQN) data transmission scheduling algorithm was proposed for network transmission application scenarios that require low latency and high reliability in the fields of in-vehicle networks, telemedicine, and industrial control. The main purpose and task of this algorithm was to reduce network delay and improve the stability of network transmission. The ERDQN algorithm optimized the queuing strategy at the sender side based on the deadline-aware transport protocol (DTP), and gave fully consideration to the priority of data blocks and Deadline, taking them as important factors for calculating the order of entering the waiting queue, which avoided the problem of losing Deadline of data blocks and reduced the queuing delay of network transmission. Meanwhile, in the congestion control, the current network transmission state was used as the feature vector to predict the parameters of the next network transmission state and give different reward factors for evaluation. Through the iterative learning of the ERDQN network, the optimal parameters were automatically adjusted to fit the current network transmission. The average transmission rate was high and stable during the subsequent network link transmission process, which alleviated the problems of network congestion and transmission stability and reduced the network transmission delay. Experimental results show that the average queuing delay and transmission delay of the ERDQN algorithm were much lower than those of the traditional congestion control algorithm (Reno algorithm), and much higher than the traditional congestion control algorithm in terms of quality of experience (QoE), which could minimize the network transmission rate fluctuation, reduce the packet loss rate, and provide stable and reliable transmission.

Keywords: congestion control; transmission rate; low latency; QoE; Reno

车载网络、远程医疗、工业控制等领域需要低时延、高可靠性的网络传输。网络时延包括传输时延、传播时延、处理时延和排队时延。传播时延与传输

介质物理特性有关, 该部分时延很难减少。所以降低网络时延的方法, 主要从处理时延、传输时延和排队时延方面考虑。

目前, 降低网络时延的方法主要是优化排队策略和优化拥塞控制。如刘正飞等^[1]使用自动队列管理算法 (active queue management, AQM) 管理数据等待队列长度, 提高了网络链路利用率, 但可能导致重要数据无法及时到达的问题。秦唯特^[2]使用数

收稿日期: 2021-04-22

基金项目: 上海教委重点项目“Python 与机器学习应用实践”(304-AC9103-20-368414002)

作者简介: 常梦磊(1997—), 男, 硕士研究生

通信作者: 李鲁群, liluqun@gmail.com

据中心网络算法 (data center TCP, DCTCP) 和显式拥塞指示标记方法 (explicit congestion notification, ECN) 优化排队队列过长问题, 但存在吞吐量抖动的问题。刘岩^[3]采用随机流量预测算法 (random early detection, RED) 解决拥塞问题, 但会出现网络链路参数调整不及时、拥塞控制有延迟的问题。邓仕军^[4]提出了基于 Q-learning 算法的拥塞控制算法, 但算法易出现因为样本不均衡导致的过拟合现象。

清华大学在 2019 年提出了基于快速用户数据报协议 (user datagram protocol, UDP) 互联网连接协议 (quick UDP internet connections protocol, QUIC) 的最后期限感知的传输协议^[5] (deadline-aware transport protocol, DTP), 该协议继承了 QUIC 协议的多流支持并且具有数据块传输、Deadline、优先级等多个特性, 明确指出在 DTP 中实现低时延需要满足

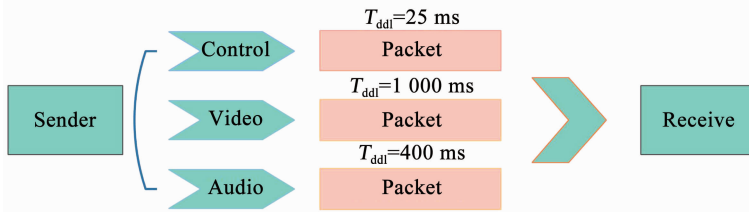


图 1 数据块发送过程

Fig. 1 Data block sending process

视频数据块包括音频 (Audio)、图像 (Video)、控制 (Control) 信号 3 种数据类型。其中 Deadline 的关系是

$$C_{ddl} < A_{ddl} < V_{ddl} \quad (1)$$

式中 C_{ddl} 、 A_{ddl} 、 V_{ddl} 分别是 Control 信号、Audio 信号、Video 信号的 Deadline。如果链路中发生拥塞, 容易造成控制信号数据等待时间过长超出 Deadline 问题, 进而导致数据重传, 增加了网络负载。

1.2 网络传输过程中不稳定且传输速率低

传统的 Reno 算法, 发送速率在慢启动阶段缓慢增加, 当出现超时重传或者丢包现象时, 将慢启动阈值 (ssthresh) 值设置为当前拥塞窗口 (cwnd) 的一半, 重新传输丢失的数据包, 并设置 cwnd 的值为 ssthresh 加上 3 个报文段 (MSS) 的长度, 计算公式为

$$W = S + 3 \times M \quad (2)$$

式中: W 为拥塞窗口大小, S 为慢启动阈值, M 为报文段长度。网络传输链路中出现丢包或者超时现象时, 发送数据窗口减半, 传输速率出现不规则波动, 导致网络链路平均传输速率降低。

2 ERDQN 算法

针对以上问题, 本文提出了 ERDQN 算法, 该算法相比于传统的拥塞控制算法在排队策略中考虑了

数据包的 Deadline。本文在 DTP 协议基础上提出了 ERDQN 算法, 该算法充分考虑了数据发送端数据块的优先级和 Deadline, 优化了排队策略, 降低了排队时延。利用强化学习网络, 自主学习调整传输参数, 降低了传输时延。实验表明本文提出的 ERDQN 算法相比于传统拥塞控制算法收敛速度更快, 排队时延和传输时延更低, QoE 更高。

1 传统拥塞控制算法存在的问题

1.1 数据块排队丢失 Deadline

传统的拥塞控制算法采用排队等待策略是先进先出策略 (first input first output, FIFO), 该策略的弊端是当数据块中包含不同的数据类型 (即包含具有不同的 Deadline), 无法满足每种数据类型的 Deadline。例如发送数据块是视频, 详细信息见图 1。

数据块的 Deadline 和优先级, 以及利用强化学习网络自主学习调整网络传输参数, 实现了低时延、高可靠性的网络传输。

2.1 ERDQN 算法架构

本文提出的 ERDQN 算法根据当前时刻的网络状态预估下一个时刻网络状态, 自动调整网络传输的相关参数, 从而降低了网络传输的时延, 保证数据高速稳定的传输。

ERDQN 算法中的基于经验回放的 DQN^[6-7] 网络包括两大模块 (见图 2), 分别是 Net 模块^[8] 和 Memory 模块^[9], Net 模块的作用是计算和预估网络参数调整带来的收益值, Memory 模块的作用是存储 Net 模块的训练记忆和记忆的优先级。

Net 模块包括 MainNet 和 TargetNet 两个神经网络, 网络由全连接层、DropOut 层、全连接层构成。隐藏层的单元数是 30, DropOut 层的权重设置为 0.5。两个神经网络结构相同但是参数更新频率却不相同, 其中 MainNet 是用于预测当前网络状态值, 参数是实时更新的, 当其参数更新 100 次后, 将当前网络参数传递给 TargetNet 网络, TargetNet 是用来预测目标网络状态值。在经过两大模块的运算, 输出网络链路的预估状态。在重采样学习中, Memory 模块根据优先级抽取学习记忆以供 Net 重复训练。

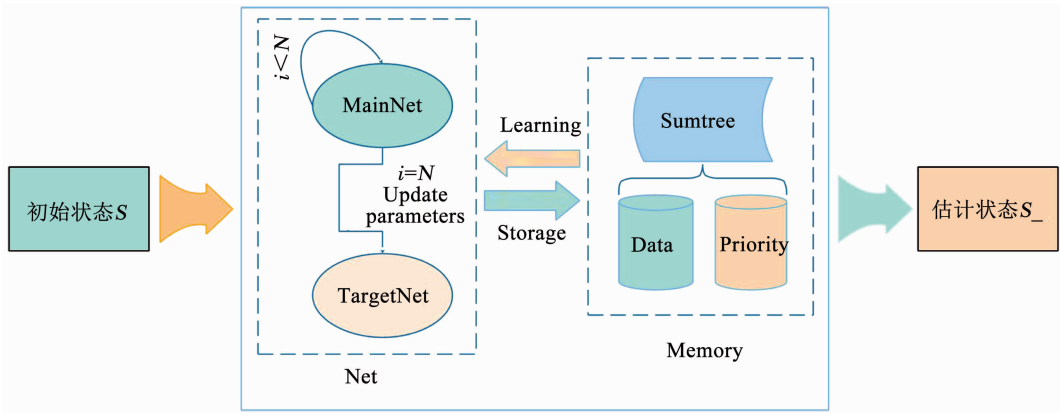


图 2 Experience replay DQN 流程

Fig.2 Flow of experience replay DQN

2.2 ERDQN 算法描述

ERDQN 算法针对传统的 FIFO 排队策略不能够满足不同数据类型的 Deadline 问题,在优化后的排队策略中考虑了数据类型的 Deadline 和优先级。首先计算当前数据块存活的时间,计算公式是

$$T_b = b_{cur} - b_{cre} \quad (3)$$

式中: T_b 为数据块的存活时间, b_{cur} 为当前时间, b_{cre} 为数据块创建时间。计算数据块紧急程度的公式为

$$u = -1 \times T_b \times D \quad (4)$$

式中: u 为数据块紧急程度, T_b 为数据块存活时间, D 为数据块的截止日期。再根据不同数据类型的优先级,决定进入排队等待队列的顺序。如在游戏视频传输过程中,包含以下 3 种数据类型,每种不同的数据类型包括不同的 Deadline 和优先级,具体见表 1。

表 1 数据类型及相关属性

Tab.1 Data types and related attributes

类型	T_{ddl}/ms	P
Control	50	h
Video	1 000	l
Audio	500	m

优先级代表不同的增益,h 为 0.5,m 为 0.3, l 为 0.1,通过下式计算出进入排队等待队列的顺序。

$$O = u \times p \quad (5)$$

式中: O 为进入等待队列顺序的衡量值, u 为紧急程度, p 为优先级。 O 的数值越大,越早进入数据等待队列。在强化学习网络中,为了避免过拟合现象,改进了 Memory 模块采样的过程,在采样时按一定比例抽取,即抽取每个优先级一定比例样本。训练记忆的优先级的划分依据 F , F 是 Q_{eval} 与 Q_{target} 之间的距离,其计算公式为

$$F = |Q_{target} - Q_{eval}| \quad (6)$$

式中: Q_{eval} 为当前网络状态值, Q_{target} 为目标网络状态值。 F 转换成优先级的具体的公式为

$$R = (F + \varepsilon)^\eta \quad (7)$$

式中: R 为训练记忆的优先级, ε 为一个很小的常数,该参数的设置是为了避免记忆的优先级出现零,而 η 是控制高低误差之间的差异,其取值范围是 $[0,1]$ 。 η 可以控制优先级在具体的计算中的权重。式(8)用来计算不同优先级记忆被采样的概率

$$P_i = \frac{R_i}{\sum_k R_k} \quad (8)$$

式中 P_i 为训练记忆样本 i 被采样的概率, R_k 为训练记忆样本 k 的优先级。Net 网络重采样学习时,首先选择采样数据的优先级,再将优先级与根节点、左子树、右子树相互比较,选择相应优先级区间,重复上述过程直至叶子节点。不同优先级的采样的比例为

$$H = H_{R_i} \times P_i + \dots + H_{R_k} \times P_k \quad (9)$$

式中 H 为采样样本数量, H_{R_i} 为样本优先级 R_i 的样本总量。该抽样方法既保证样本的学习效率,又避免了因为样本不均衡可能出现的过拟合现象。

3 实验

3.1 数据集及实验环境

本实验中使用的数据集和仿真环境来源于 Altrans 智能网络挑战赛,数据集包括数据块(B)和日志。每一个 B 是时间长度为 20 s 的数据,其内容是日常使用中部分的 blocks 数据(b_i), B 中包括 Video 类型和 Audio 类型,其属性包括时间戳(T_s)、帧大小(F_s)和是否为关键帧(Y),详细信息见表 2。

表 2 Video 和 Audio 的 trace 格式

Tab.2 Trace format of Video and Audio

T_s/s	$F_s/Byte$	Y
00.001 3	147 800	1
00.003 7	272 301	0
...

在数据集中, B 数据中的 Video 类型和 Audio 类型拥有不同的优先级。相对于视频来说, Audio 的优先级高于 Video 的优先级, 这样的设计可以更好地满足使用者的需求。

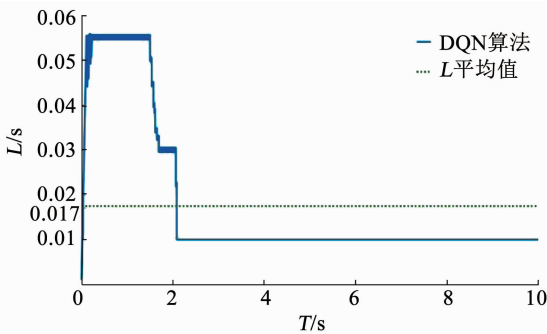
实验的仿真环境是由清华大学构建的 DTP 模拟器 (DTP-Emulator), 该系统使用 DTP 作为传输协议, 模拟了在不同网络类型下数据传输的过程。在 DTP-Emulator 中可以使用自行设计的数据传输调度算法, 并可以输出数据传输过程中的排队时延、传输时延等相关日志数据。官方给出了相关的质量系数评价模型

$$Q_{QoE} = \sum_{b_i \in B} \alpha \times P \times (1 - M_d) + (1 - \alpha) \times (1 - M_d) \quad (10)$$

式中: Q_{QoE} 为相关模型的质量系数, P 为指数数据块的优先级, M_d 代表数据块是否超过 Deadline, α 为数据块在 Deadline 内到达接收端的增益值。

3.2 实验结果分析

Reno 算法是最常用的拥塞控制的算法, 在拥塞控制方面有着良好的效果, 所以本实验在 DTP-Emulator 仿真环境中测试在仿真数据集和传输数据



集中 Reno 算法、DQN 算法和本文设计的 ERDQN 算法的性能, 并输出相关的传输日志。

实验结果见表 3、4, 本文提出的 ERDQN 算法在排队时延 (Q_t)、最大排队延迟 (Q_{mt})、平均传输速率 (V_t)、QoE (Q_{QoE}) 方面明显优于 Reno 算法和 DQN 算法。实验结果说明在不同的数据集中本文提出的 ERDQN 算法可以更好地满足数据块的优先级、Deadline 的需求, 能够实现低时延、高可靠性的传输目标。

表 3 仿真数据集结果

Tab.3 Results on simulation dataset

算法	Q_t/s	Q_{mt}/s	V_t	Q_{QoE}
Reno	0.002 800	0.041 250	—	486
DQN	0.001 100	0.041 250	1 000	496
ERDQN	0.000 850	0.041 250	1 500	575

表 4 传输数据集结果

Tab.4 Results on transmission dataset

算法	Q_t/s	Q_{mt}/s	V_t	Q_{QoE}
Reno	0.003 000	0.041 250	—	476
DQN	0.001 000	0.041 250	900	480
ERDQN	0.000 750	0.041 250	1 200	556

实际网络传输过程中 ERDQN 算法、DQN 算法和 Reno 算法的时延 (L) 随着时间 (T) 变化曲线见图 3, L 包括排队时延和传播时延, 它反映了拥塞控制算法在网络传输中的实际效果。

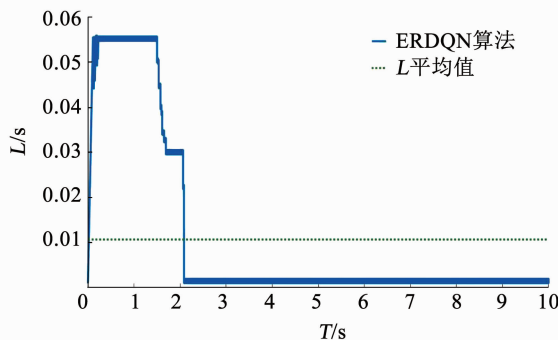
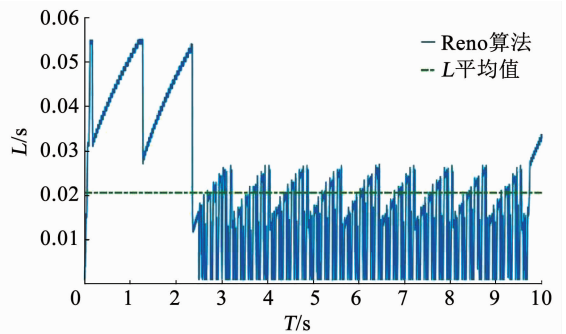


图 3 L 随时间的变化

Fig.3 Variation of L with time

在实际传输过程中,ERDQN 算法的平均时延 < DQN 网络 < Reno 算法,且在 2 s 之后 ERDQN 和 DQN 都保持稳定的传输,但 Reno 算法的时延一直在剧烈波动。相比于 DQN 算法,ERDQN 算法在达到稳定传输的时延约为 0.001 s,远远小于 DQN 的稳定传输时延。由上述数据表明本文提出的 ERDQN 算法可以提供低时延、高可靠性的网络传输。

4 结 论

针对传统拥塞控制算法中不能满足不同类型数据的 Deadline,网络传输不稳定且平均传输速率低的问题,本文提出了 ERDQN 算法,优化排队策略,满足了数据块的 Deadline 和优先级,降低了排队时延;且对网络参数调整过程不断学习,找到适宜当前网络传输的最优参数,降低了传输时延。实验证明本文所提出的 ERDQN 算法相比于 Reno 算法和 DQN 算法,具有时延低、平均传输速率高、QoE 高等优点,且本文提出的算法收敛速度快,传输更加稳定,能够满足车载互联网、远程医疗、工业控制等领域的低时延、高可靠性的传输要求。

参 考 文 献

- [1] 刘正飞, 孙金生, 胡斯乔. 基于模型有效性评价的主动队列管理算法[J]. 控制理论与应用, 2019, 36(4): 570
LIU Zhengfei, SUN Jinsheng, HU Siqiao. An active queue management algorithm based on model validity evaluation [J]. Control Theory and Applications, 2019, 36(4): 570. DOI: 10.7641/CTA.2018.80631
- [2] 秦唯特. 云计算环境中的流量控制算法的研究[D]. 北京: 北京邮电大学, 2018
QIN Weite. Research on traffic control algorithms in cloud computing environment [D]. Beijing: Beijing University of Posts and Telecommunications, 2018
- [3] 刘岩. 基于流量预测的 RED 拥塞控制算法研究[D]. 天津: 河北工业大学, 2011
LIU Yan. Research on RED congestion control algorithm based on traffic prediction [D]. Tianjin: Hebei University of Technology, 2011
- [4] 邓仕军. 基于深度 Q-learning 的自动 I/O 拥塞控制机制[D]. 武汉: 华中科技大学, 2018
DENG Shijun. Automatic I/O congestion control mechanism based

- on deep Q-learning [D]. Wuhan: Huazhong University of Science and Technology, 2018
- [5] SHI Hang, CUI Yong, QIAN Feng, et al. DTP: deadline-aware transport protocol [C]//Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019. New York: ACM, 2019: 1. DOI: 10.1145/3343180.3343191
- [6] CHEN Guoqing, CAI Qiong, FU Xingbao, et al. Research on algorithms of computing offloading and resource allocation based on DQN[J]. Journal of Physics: Conference Series, 2021, 1748(3): 032047. DOI: 10.1088/1742-6596/1748/3/032047
- [7] GUO Siyu, ZHANG Xiuguo, DU Yiquan, et al. Path planning of coastal ships based on optimized DQN reward function[J]. Journal of Marine Science and Engineering, 2021, 9(2): 210. DOI: 10.3390/JMSE9020210
- [8] TAN S, MOHAMAD K, MOHAMED F, et al. Balancing excitation and inhibition of spike neuron using deep Q network (DQN) [J]. Journal of Physics: Conference Series, 2021, 1755(1): 012004. DOI: 10.1088/1742-6596/1755/1/012004
- [9] YANG Leyou, JIA Jie, CHEN Jian, et al. Online reliability optimization for URLLC in HetNets: a DQN approach [J]. Neural Computing and Applications, 2020: 1. DOI: 10.1007/s00521-020-05492-4
- [10] PAN Wansu, TAN Haibo, LI Xiru, et al. Improved RTT fairness of BBR congestion control algorithm based on adaptive congestion window [J]. Electronics, 2021, 10(5): 615. DOI: 10.3390/ELECTRONICS10050615
- [11] KUMAR D M, ARTHI R, ARVINDHAN C, et al. Traffic congestion control synchronizing and rerouting using LoRa [J]. Microprocessors and Microsystems, 2021: 104048. DOI: 10.1016/J.MICPRO.2021.104048
- [12] ZEMOURI S, DJAHEL S, MURPHY J. An altruistic prediction-based congestion control for strict beaconing requirements in urban VANETs [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 49(12): 2582
- [13] PARANJOTHI A, KHAN M S, ZEADALLY S. A survey on congestion detection and control in connected vehicles [J]. Ad Hoc Networks, 2020, 108: 102277. DOI: 10.1016/j.adhoc.2020.102277
- [14] SAEDI T, EL-OCLA H. TCP CERN+: revisiting TCP congestion control in wireless networks with random loss [J]. Wireless Networks, 2020: 1. DOI: 10.1007/s11276-020-02459-0
- [15] WANG Zhiran. Research and strategy of urban traffic congestion control [J]. Urban Transportation & Construction, 2020, 6(2): 34. DOI: 10.18686/utc.v6i2.87
- [16] REN Y, LU G, SUN C. Joint congestion control and resource allocation in cache-enabled sensor networks [J]. Sensors (Basel), 2019, 19(13): 2961. DOI: 10.3390/s19132961

(编辑 苗秀芝)