基于 DRL 的 MEC 卸载网络竞争窗口优化

詹　御,张郭健,彭麟杰,文　军*
(电子科技大学 信息与软件工程学院,四川 成都 610054)

摘　要:多接入边缘计算(MEC)是一种新兴的云计算。低算力的物联网设备可以把计算任务卸载到 MEC 上处理,以提供更高质量的服务。当 MEC 的卸载网络在面临大量设备接入时,各设备请求服务时相互竞争的网络连接会发生大量碰撞从而导致 MEC 卸载网络的性能下降。在 Wi-Fi 作为 MEC 的接入点场景中,面对较少数量的设备接入时,802.11 协议的退避算法可以合理地设置竞争窗口的值来减轻碰撞所带来的网络吞吐量下降,但默认的退避算法无法有效地应对较多的接入设备或动态变化的网络拓扑。为优化竞争窗口的设置以改善网络性能,提出两种竞争窗口优化的深度强化学习(DRL)方法,将深度 Q 网络(DQN)与深度确定性策略梯度(DDPG)方法分别用于优化 MEC 卸载网络竞争窗口大小的设置,以有效应对大量的接入设备和网络拓扑的动态变化。仿真实验结果表明,DRL 方法在不同的接入设备数量、静态网络拓扑和动态网络拓扑的条件下,均可稳定网络的吞吐量,且相比于默认的方法有较大的提升,静态条件下相对提升 46%,动态条件下相对提升 36%,且并没有破坏网络服务的公平性。

关键词:深度强化学习;多接入边缘计算;物联网;网络优化;竞争窗口
中图分类号:TP393　　**文献标识码**:A　　**文章编号**:1673-629X(2022)06-0099-07
doi:10.3969/j.issn.1673-629X.2022.06.017

Optimization of Contention Window of MEC Offloading Network Based on DRL

ZHAN Yu,ZHANG Guo-jian,PENG Lin-jie,WEN Jun*
(School of Information and Software Engineering,University of Electronic Science and Technology of China,Chengdu 610054,China)

Abstract:Multi-access edge computing (MEC) is an emerging type of cloud computing. Low computing power IoT devices can offload computing tasks to MEC for processing to provide higher quality services. When the offloading network of MEC is faced with a large number of device accesses,there are a lot of collisions among competing network connections when each device requests services,thus leading to performance degradation of the MEC offload network. In the scenario where Wi-Fi is used as the access point for MEC,the 802.11 protocol's back-off algorithm can reasonably set the value of the contention window (CW) to mitigate the network throughput degradation caused by collisions when facing a smaller number of devices,but the default back-off algorithm cannot effectively cope with a larger number of access devices or dynamically changing network topology. To optimize the setting of the contention window to improve network performance,two deep reinforcement learning (DRL) methods for CW optimization are proposed. The deep Q network (DQN) and deep deterministic policy gradient (DDPG) methods are used to optimize the setting of the contention window size for MEC offloading networks to effectively cope with a large number of access devices and dynamic changes in network topology,respectively. The simulation experimental results show that the DRL method stabilizes the network throughput under different numbers of access devices,static network topologies and dynamic network topologies,and has a large improvement over the default method,with a relative improvement of 46% under static conditions and 36% under dynamic conditions,without destroying the fairness of network services.
Key words:deep reinforcement learning;multi-access edge computing;internet of things;network optimization;contention window

0　引　言

智能手机、平板电脑、可穿戴设备等个人智能设备和物联网设备的日新月异正在推动新服务、应用的出现,如虚拟现实(virtual reality,VR)、增强现实

收稿日期:2021-07-13　　**修回日期**:2021-11-15
基金项目:四川省科技项目(SCITLAB-0013)
作者简介:詹　御(1998-),男,硕士,CCF 会员(H7208G),研究方向为云计算与大数据;通讯作者:文　军(1966-),男,博士,副教授,研导,研究方向为网络工程、大数据处理。

(augmented reality,AR)、人脸识别和移动医疗等,这些服务和应用都具有较高的计算要求。尽管现今的智能设备拥有更多的计算能力,但它们仍然无法有效地运行计算密集型的应用程序或提供稳定有效的服务。为了应对这一挑战,多接入边缘计算(multi-access edge computing,MEC)[1]被提出来,它将计算和存储资源从远程云中分出,部署在更靠近设备的一端。因此,设备的计算密集型应用任务可以被卸载到临近的 MEC 服务节点上,以实现高质量的服务(quality-of-service,QoS)。物联网设备主要通过接入点(access point,AP。例如 Wi-Fi 连接)请求 MEC 服务,如图 1 所示。AP 关联到一个或多个 MEC 服务器,多个用户设备(user m)接入并请求 MEC 服务。在物联网中,智能设备主要用于数据采集和提供简单的服务,其计算资源有限,需要 MEC 提供算力支持以达到较好的 QoS。但大量的设备接入会导致接入点的竞争冲突,尤其是在公共场所、人群聚集和大量部署物联网设备的地方,智能设备在请求 MEC 服务时必须要和该 AP 连接的其他设备竞争连接服务[2],即竞争网络资源把任务卸载到 MEC 服务器获取计算服务。

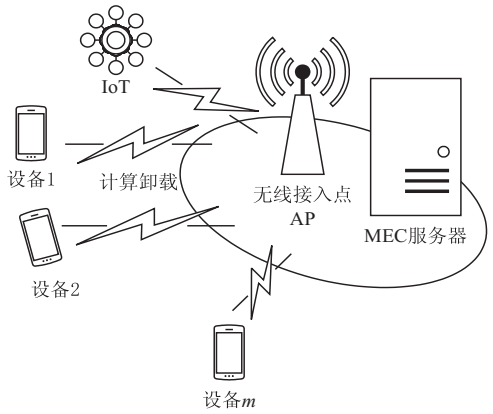

图 1 MEC 卸载网络

MEC 网络的接入主要是无线网络的方式,例如 Wi-Fi。针对 Wi-Fi 的无线访问接入协议,IEEE 在 2020 年发布的 802.11ax 协议相比于之前的 802.11 协议较好地提高了 Wi-Fi 网络效率[3]。然而,802.11ax 为了确保向后兼容,基本的网络接入传输数据方法保持不变[4]。这种无线网络接入方法被称为载波感应多路接入与避免碰撞(CSMA/CA)算法:每个待接入设备随机“退后(backs-off)”,即在接入网络前等待一定数量的时隙(back-off slots)。这个时隙被称为竞争窗口(contention window,CW)。为了减少多个设备相同的随机退避的概率,IEEE 802.11 退避算法规定每次传输超时(碰撞)后 CW 会翻倍,并定义了 CW 的最小值和最大值。尽管这种默认的方法需要很少的计算,但是会导致低效的网络接入,特别是在有密集接入设备的 MEC 网络中,无法高效、稳健地应对网络的快速变化[5]。

CW 的选取大小对网络性能有直接的影响,因此 CW 优化一直是网络优化研究分析的主题。常规的优化方法包括使用控制理论[6]和监测活跃用户的数量[7]。随着具有高计算能力的网络设备的普及,现在可以使用机器学习(machine learning,ML)方法来分析优化 CW[8-10]。但 ML 方法的选择受到网络优化问题本身性质的很大限制。如分析模型[11],可以提供最佳的 CW 值,但只在某些假设和准静态设置下,所以无法依赖它们来训练模型。无导数优化算法是较合适的选择,强化学习(reinforcement learning,RL)就是无倒数优化算法,是较为适合用来改善无线网络性能的 ML 方法[12]。因为 RL 涉及到智能软件代理(网络节点,AP)在环境(MEC 卸载网络)中采取行动(设置 CW 值),以最大化奖励(网络吞吐量)。RL 的无导数、无模型的特性,比传统的、基于模型的优化方法有更好的泛化能力。近年来,已经有在计算机无线网络中使用强化学习的研究:文献[8,13]将 RL 应用于无线局域网的干扰对策;文献[14]提出基于 RL 的通用适应性介质访问控制(MAC)算法;用于水声传感器网络的 MAC 方法[15]、无线网络资源调度算法[16]以及用于调整可重新配置无线天线的方法[17]都使用了强化学习。文献[18]也使用了 RL 用于网络传输优化。但他们大部分使用的是 Q-learning,是表格型的 RL 求解方法,这不适合应用在具有密集连接、网络拓扑动态变化的 MEC 卸载网络环境中。因为 Q-learning 存在维数灾难——查找和存储都需要消耗大量的时间和空间。文献[19]指出,通过使用深度人工神经网络(deep neural networks,DNN)构建深度强化学习(DRL),可以进一步提高 RL 的性能,使其具有更优越的可扩展性。

因此,该文提出将 DRL 改进并应用于优化物联网的 MEC 卸载网络中,通过合理预测、设置 CW 值来优化提高 MEC 卸载网络的吞吐量。改进两种 DRL 方法,深度 Q 网络(deep q-network,DQN)[19]和深度确定性策略梯度[20](deep deterministic policy gradient,DDPG),使其应用在 Wi-Fi 作为 MEC 的 AP 的环境中。仿真实验表明这两种方法都提升了 MEC 卸载网络的性能,特别是在密集设备接入的动态场景中,且没有破坏网络的公平性。

1 基于深度强化学习的 MEC 卸载网络优化方法

本节研究了强化学习、深度强化学习理论方法;然后基于 WiFi 物联网的 MEC 卸载网络模型,建立出深度强化学习优化 CW 算法。

1.1　强化学习和深度强化学习

RL 源自于马尔可夫决策过程(Markov decision process,MDP),RL 的基本理论是:使用一个可以自我学习的代理 agent 来替代模型分析的方式去寻找最优解。agent 能够通过相应的行动 action 与环境 environment 互动(采样 state,如图 2),每一步行动都会使 agent 更接近(或更远离)其目标(奖励,reward)。通过训练,代理增强了其决策策略(执行什么 action),直到代理学会(获得奖励)了在环境的每个状态下的最佳决策(动作顺序)。在 DRL 中,代理的决策是由一个 DNN 训练得到的。该文考虑了两种不同行动空间的 DRL 方法:DQN 的离散动作策略和 DDPG 的连续动作策略。DQN 试图预测每个动作的预期奖励,即是基于价值(奖励,reward)的方法。与基本的 RL 相比,DQN 附加了深度神经网络,可以更有效地推断出尚未观察到的状态的奖励。与 DQN 不同,DDPG 是基于策略(动作,action)的方法,它试图直接学习最优策略,可以产生连续的动作输出。DDPG 包括两个神经网络:一个行动者(actor)网络和一个评判者(critic)网络。行动者根据环境状态做出决定动作,而评判者是一个类似 DQN 的神经网络,试图学习行动者行动的预期奖励收益。

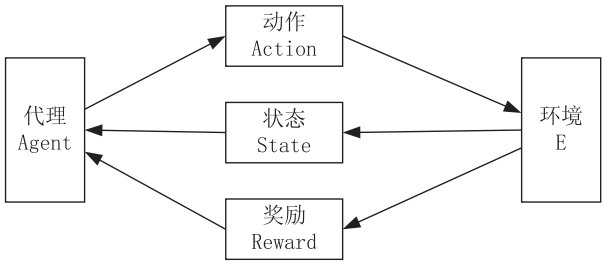

图 2　强化学习模型

为了将 DRL 应用于 MEC 计算卸载网络,该文做了一个映射,一个代理(部署在 AP 上)、环境状态(网络环境)、可用的行动(设置相应的 CW 值)和收到的奖励(网络吞吐量)。AP 上的代理是观察网络的状态,设置 CW 值(动作),以使网络性能(奖励)最大化。

1.2　MEC 卸载网络优化的 DRL 原理

作为 MDP 的改进版,RL 是在不知道可观察到的状态以及不知道每个状态的转移概率时解决 MDP 问题的方法。实际上,RL 是一个部分可观察的马尔可夫决策过程(partially observable Markov decision process,POMDP),它假定不可以完全观察到环境的所有状态,从部分状态和历史经验中推测新状态和预期的奖励。POMDP 表示为 $(S,A,T,R,\Omega,O,\lambda)$:

S: 状态(state)集合;

A: 动作(action)集合;

T: 状态之间转换的概率集合;

R: 奖励函数, $R=f(S,A)$;

Ω: 观测结果集合;

O: 观测概率集合;

λ: 折扣系数。

因为 AP 能观察到网络的全部接入,并可以通过信标数据帧(beacon frames)以集中的方式控制接入的设备,还具有处理 DRL 优化的计算要求,所以把代理 agent 部署在接入点 AP 上。此外,AP 能够与其他 AP 交换信息,可以成为基于软件定义网络(SDN)的多代理 MEC 架构的一部分[5]。

环境状态 $s \in S$ 是当前连接到网络的所有设备的确切状态。由于网络的接入设备是动态变化的,不可能收集到确切的信息。因此,该问题被表述为 POMDP 而不是 MDP。

代理的行动 $a \in A$ 直接对应于设置新的 CW 值。在基于 Wi-Fi AP 的 MEC 中,CW 有默认的最大值和最小值($CW_{min}=2^5-1=31$, $CW_{max}=2^{10}-1=1\ 023$)[21],在最大值和最小值的区间,如果设置一个较小的 CW 值,当许多设备试图在同一时间传输数据,很有可能其中一些设备有相同的退避间隔。这意味着将不断出现碰撞,对网络性能产生严重影响。另一方面,如果设置较大的 CW,当很少有设备传输数据时,它们可能会有很长的退避延迟,导致网络性能下降。该文提出的两方法的动作策略分别为离散的和连续的。离散的动作会输出正整数 $a \in \{0,1,2,3,4,5\}$,即 agent 根据环境状态在 0 到 5 取一个整数;连续的动作会输出一个 0 到 5 之间的实数 $a \in [0,5]$。动作的输出 a 值将被用来更新网络的 CW 值,根据以下公式:

$$CW=\lfloor 2^{5+a} \rfloor - 1 \tag{1}$$

即可能的动作集合 $A=[0,5]$,每个动作执行后会更新网络的 CW 值,并伴随着网络环境状态 $s \in S$ 的改变,产生新的环境 $s' \in S$,环境 $s \to s'$ 会有一个概率转移 $T(s' \mid s,a)$。

该文对于奖励 $r \in R$,设置为与网络性能——网络吞吐量(network throughput,NT,每秒成功传输的比特数 bits/s),相关的变量。因为在部署了 agent 的 AP 上,可以较为容易、及时地获取到当前时间的网络吞吐量。其次,对于标准的 DRL 的奖励是在 0 到 1 之间的一个实数,因此对奖励 r 做归一化处理,用观察到的每秒比特数除以预期的最大吞吐量:

$$r=\frac{NT_{now}}{NT_{expect}}, r \in [0,1] \tag{2}$$

每次观察环境状态为 $\omega \in \Omega$,它表示为 agent 根据观察信息所得的网络当前最大可能的状态。对每个观察到的环境有一个观测概率 $o \in O$,表示在网络中观察到的当前碰撞概率 $p_{collision}$(传输失效概率),该概率

根据传输的数据帧的数量 N_t 和正确接收的数据帧的数量 N_r 来计算：

$$p_{\text{collision}} = \frac{N_t - N_r}{N_t}, p_{\text{collision}} \in [0,1] \tag{3}$$

$p_{\text{collision}}$ 的值直接反映了动作 action 选取的 CW 值的优劣性。一般的，AP 上的 agent 不能直接获取，但是 AP 作为总的接入点，作为发送方或接收方参与了所有数据帧的传输，通过数据帧的数据位绑定 N_t，AP 能够获取到设备的 N_t，对于 N_r 可以直接在 AP 端统计得到。

1.3 DRL 优化 MEC 卸载网络

对于该文的两种 DRL(DQN 和 DDPG)的理论推导和性质，就不做展开，读者可以参考文献[20-21]来辅助理解两种 DRL 的性质和异同。使用 Wi-Fi 作为 AP，默认的 CW 设置按照标准的 802.11 协议规则，即指数退避算法(exponential back-off algorithm, EBA)[21]，在[0-CW]中选择随机退避时隙，当冲突发生时，CW 在限制大小内以 2 的指数递增(CW ∈ [CW_{min}, CW_{max}])。参考标准 EBA，该文改进两个 DRL 方法——DQN 和 DDPG，用于优化 MEC 计算卸载网络。通过 DRL 优化网络的 CW 值，最大化全局网络吞吐量为奖励——优化目标，以提升网络的效率。如图 3 所示，agent 部署在 AP 上，获取网络环境状态，设置相应的 CW 值来获取最大化全局网络吞吐量。

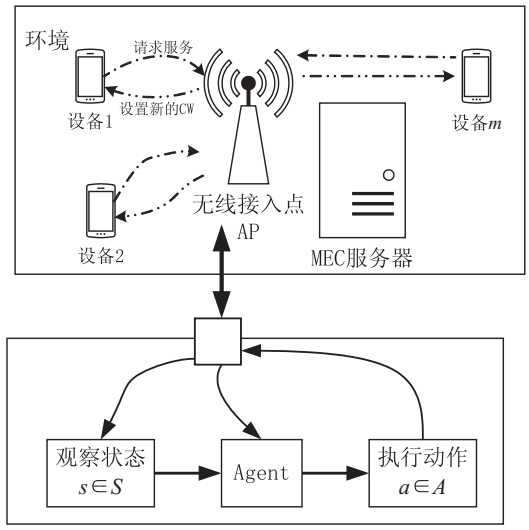

图 3 DRL 优化 MEC 卸载网络模型。

agent 分别基于 DQN 和 DDPG，它们的区别在于：DDPG 实质上是 DQN 的一种在连续动作上的扩展算法，DDPG 在 DQN 的基础上多了一些决策(policy)系列的操作。因此对于 DQN 和 DDPG，该文设计了同样的 CW 优化算法：

DRL-CW 优化算法：

1. 初始化观测 B_{obs}，长度为 h
2. 获取 agent 的动作函数参数 θ
3. 获取 agent 的动作函数 A_θ
4. 定义变量收、发帧数 N_r，N_t
5. 设置交互时间周期 Δt
6. 获取训练标志位 isTrain
7. 设置重放缓存 B
8. 最新更新时间设置为当前时间：
   $last_{update} \leftarrow current_{time}$
9. 初始化 CW ← 31
10. 初始化状态向量 s
11. for $t = 1,2,\cdots,\infty$ do：
12. 获取 N_t，N_r
13. 把 N_t，N_r 放入 B_{obs} 中
14. if ($last_{update} + \Delta t$) ≤ $current_{time}$：
15. obs ← getObs(B_{obs}) #生成观察值
16. $a \leftarrow A_\theta$(obs) #生成执行动作值
17. CW ← ($\lfloor 2^{5+a} \rfloor - 1$) #更新 CW
18. if isTrain：
19. NT ← ($N_r/\Delta t$) #计算有效网络吞吐量
20. r ← normailization(NT) #奖励
21. 把 (obs, a, r, s) 放入 B 中
22. s ← obs
23. b ← 从 B 中小批量取样
24. 用 b 训练优化 A_θ 的参数 θ

getObs() 根据当前网络的收发状态生成相应的状态值。

normailization() 的功能是归一化奖励值，参考公式(2)。

模块 getObs() 用于计算最近观察到的碰撞概率 $H(p_{\text{collision}})$ 历史的平均值和标准偏差，由一个固定大小($h/2$)和步长($h/4$)滑动窗口决定。滑动窗口的采样将 B_{obs} 数据从一维变为二维(滑动窗口的每一步产生两个数据点)。这样产生的数据可以被看作是一个时间序列(窗口的每一步对应一组数据，每组数据具有前后的时间顺序，且状态的观测值也在不断更新)，意味着可以用循环神经网络(RNN)分析它。因为，与使用全连接神经网络进行训练分析优化相比，RNN 的设计[22]可以更深入地学习到 agent 的行为与网络性能之间的直接和间接关系。

2 仿真实验设置

实验是基于 ns3-gym[23] 网络仿真平台实现的，ns3-gym 融合了强化学习 RL 的开发分析工具 OpenAI Gym。另外，基于 ns3-gym，使用 Python 开发实现了 DQN 和 DDPG 的神经网络。实验在 Ubuntu 18.04.5 LTS 下进行，软件版本：Python (3.6.5)；TensorFlow (1.14.0)；Torch (0.4.1)。

2.1 仿真平台的网络模型设置

在 ns3-gym 网络仿真平台，搭建如图 3 所示的 MEC 计算卸载网络模型，其相关参数如下：Radio Channels：error-free，20 MHz；Protocol：IEEE 802.11ax；

Modulation: 1024 - QAM; Coding Rate: 5/6; Transmissions:single-user;Frame-Aggregation:Disabled; Transport Layer:UDP, 1500 byte packets。

为了方便 DRL 的训练和应用,做了如下的实验条件假设:(1)连接到 AP 的设备能完整、快速地把自身网络状态信息传递给 AP,即 AP 可以立即得到 N_t, N_r;(2)agent 可以立即更新设置各个连接设备的 CW 值。这样的假设是合理的,因为基于 Wi-Fi 作为 AP 的 MEC 计算卸载网络类似小范围的局域网,传输延迟可以忽略不计。如果考虑了传输延迟,会导致 DRL 的学习周期较长,收敛较慢。因此,在该实验中,不考虑延迟的因素,但这不影响实际的应用。总的来说就是忽略延迟影响,但这样的假设不会导致在实际的网络中的应用与实验偏差较大。

2.2　DRL 的训练过程优化

该文优化应用 DQN 和 DDPG 与 MEC 计算卸载网络主要分为三个阶段:(1)预训练学习;(2)训练学习;(3)应用阶段。预学习阶段先使用标准的 EBA,让整个 MEC 网络环境运行起来,作为 DRL 训练前的热身阶段;训练学习阶段,DQN 和 DDPG 分别利用提出的 CW 优化算法来优化 CW 的设置;在训练完成后是应用阶段,即将训练好的 agent 部署到 MEC 网络中替代标准的 EBA 算法。

为了让 DQN 和 DDPG 训练稳定,学习到有效的参数,该文对这两种方法的参数更新均采用局部(local)和目标(target)神经网络的训练学习方法。action 的参数学习用两个神经网络同时学习的办法,但 CW 优化算法使用在目标神经网络的结果,目标神经网络定期获取局部神经网络的参数 w_{local} 来更新自己的网络参数 w_{target}:

$$w_{\text{target}} = \tau \times w_{\text{local}} + (1 - \tau) \times w_{\text{target}} \tag{4}$$

通过这样的训练方式,在一个时间段内固定目标网络中的参数,定期按公式(4)来更新参数,达到稳定学习目标的效果,τ 是软更新系数,确保目标网络参数的稳定更新。

对于 DQN 和 DDPG 方法的训练参数,设置的参数如表 1 所示。

这些超参数设定是根据多次实验,采用随机网格搜索和贝叶斯优化得出的训练、测试效果较好的参数,具备一定的参考价值。两种算法的神经网络使用相同的结构:一个循环的长短期记忆层(long short-term memory,LSTM),然后是两个密集层,形成 8×128×64 的网络结构用于 action 的生成,即 $a = A_\theta(s_t \mid \theta^\mu)$,在 t 时刻状态 s_t 决定了 action 的值 a;μ 是上述的网络(确定性策略网络),θ^μ 是它的参数,用于生成在 s_t 下的 action。

表 1　DRL 训练参数设置

参数名称	DQN 参数值	DDPG 参数值
训练时间	60 sec	60 sec
动作采样周期	0.01 sec	0.01 sec
缓存大小 h	300	300
学习率	4e-04	—
b 的大小	32	32
B 的大小	18 000	24 000
λ	0.7	0.7
Actor 学习率	—	4e-04
Critic 学习率	—	4e-03
软更新系数 τ	4e-03	4e-03

使用 LTM 这类 RNN 可以让 agent 把之前的观察训练经验考虑进去。在 agent 的动作和网络仿真中都加入了随机性。每个实验进行 15 轮(epoch),前 14 轮为训练学习阶段,最后一轮为应用测试阶段,每轮 60 秒(60/0.01=6 000 个 episodes)。每个 agent 与环境交互周期为 0.01 s(即 10 ms),在交互周期之间执行 CW 优化算法。

为了让 agent 尽可能的探索(exploration),在执行每个 action 前都会加入一个噪声因子(noise factor),该噪声因子在学习阶段中逐渐衰减。对于 DQN,噪声是指用随机的 action 替代 agent 行为的概率。对于 DDPG,噪声从高斯分布采样并加入到代理的决策中。这样做的目的是为了缓解 DRL 的本身的 exploration-exploitation 的问题——寻找新经验(exploration)和通过已知策略获得最大化奖励(exploitation)是矛盾的,不能兼得。具体实现如下:

噪声帮助 DRL 进行 exploration

DQN:　$A'_\theta(s_t) = \text{random}(A_\theta(s_t \mid \theta^\mu), \text{random}(A))$

DDPG:　$A'_\theta(s_t) = A_\theta(s_t \mid \theta^\mu) + \text{noise}$

#　$A'_\theta(s_t)$ 是由确定性策略(DPG)加噪音得到的

在 DQN 执行动作 a 之前,加入一个随机噪音,即有较小的概率选择从动作库 A 中随机选择一个动作来执行,保证了算法的 exploration。在 DDPG 中也是类似的,不过因为 DDPG 是连续动作,不适合随机选取,就在其后加上一个较小的 noise 来实现。同样保证了 agent 的 exploration 性,以适应新的环境状态。

2.3　实验基线与训练周期

该文使用 802.11 默认的退避算法作为实验基线对照:标准的 EBA。实验的 MEC 卸载网络拓扑环境分两种,固定设备数量的(静态网络拓扑)和设备数量动态变化的(动态网络拓扑)。接入设备数量为 5 到 50 个。上一节中,之所以设置 15 个 epoch 的训练周期是实验分析得到的,参考图 4:

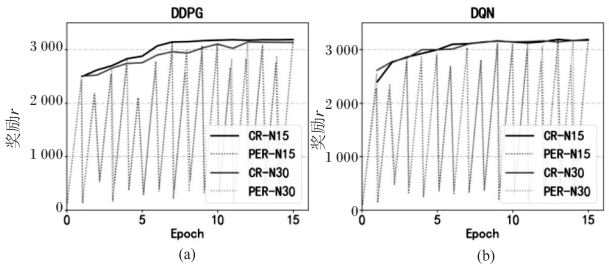

图 4 DRL 训练奖励变化

图 4 中 CR 为 cumulative-reward(最大化累积奖励),PER 为 per-episode-reward(是每个 episode 中的步进奖励)。NX 的意思是多少个设备(N15 即 15 个设备接入)。图中展示了 DQN 和 DDPG 在 15、30 个设备接入的静态网络拓扑结构下,经过 12 个 epoch 左右的训练便可获得基本趋于稳定的奖励,即在 12 个 epoch 的训练学习后可以使模型收敛。后续实验统一使用 15 个 epoch 的实验周期,即保障在 12 个以上使模型收敛,且无需更多的 epoch。

3 实验结果与分析

3.1 静态网络拓扑实验结果

在静态网络拓扑结构中,整个实验过程中都有固定数量的站点连接到 AP。从理论上讲,恒定的某个 CW 值是可以达到最佳的网络性能[5]。静态场景的实验是为了测试 DRL 算法经过训练学习是否能够实现最佳的 CW 设定,以及与标准 802.11 的 EBA 基线相比有什么差异。图 5(a)是在不同数量{5,15,30,50}连接设备下多次实验的统计平均结果,置信区间为 95%。

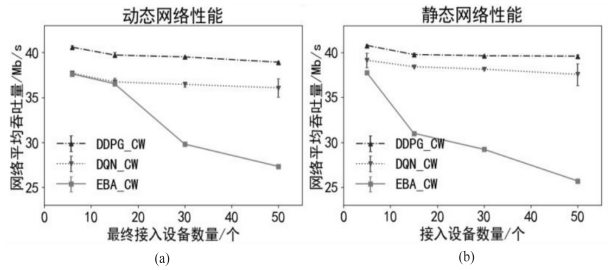

图 5 静态(a)和动态(b)网络拓扑中各方法网络性能的对比

从图中的结果可见,标准的 EBA 的性能在较多的设备接入 MEC 计算卸载网络时,网络的整体性能有明显的下降,但 DDPG 和 DQN 可以在静态网络条件下优化 CW 值以应对网络拓扑的变化。与标准的 EBA 相比,在较少的设备连接时(5 个),DRL 能提升约 6% 的性能;在较多设备连接时(50 个),约有 46% 的提升。

3.2 动态网络拓扑实验结果

在动态网络拓扑的环境下,每次实验过程中,接入设备的数量从 5 个以每次 5 个的速度递增,这样做的目的是为了增加 MEC 计算卸载网络中的碰撞率,测试算法是否能够对网络变化,做出合适的反应来稳定网络性能。

在图 6(a)中,左边的纵坐标展示的是网络性能,右边的纵坐标展示的是接入设备的数量,横坐标是实验时间。接入设备数量从 5 增加到 50,各个方法的性能变化如图中各线所示,各方法对 CW 的设定效果反映在网络的瞬时吞吐量上;两种 DRL 方法与 EBA 对比可见,有较好的稳定性,在设备数量增加到 50 时,DQN 和 DDPG 都可以保持网络性能的稳定,网络吞吐量没有明显的下降。再参考动态网络拓扑的多次实验均值结果图 5(b)(类似于图 5(a)),可见 DRL 优化 CW 值的方法可以较好地优化 MEC 卸载网络多接入情景下的网络性能,且具有优于标准 EBA 方法的效果(50 个接入设备时 DRL 约有 36% 的相对提升)。

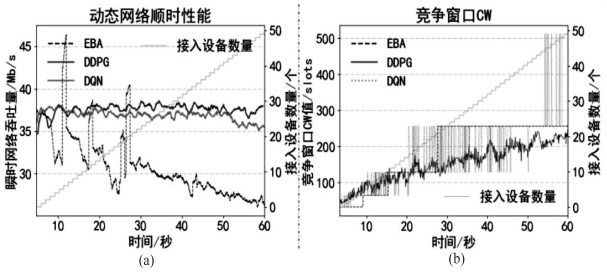

图 6 动态网络拓扑中各方法的瞬时性能对比(a)与 CW 设置(b)

对于 DRL 如何设置相应的 CW,该文给出动态网络拓扑中一个接入设备的 CW 值变化情况,如图 6(b)所示。不同的 DRL 方法所展现的 CW 不一样,因为 DQN 是离散动作类型的 RL 方法,而 DDPG 是连续动作型的,其本质原理可参考公式(1)及其说明。DDPG 的连续动作机制可以比 DQN 优化的 CW 值更快速变化,可以更好地适应网络的动态变化;标准的指数退避 EBA 作为参考,但其被 DQN 的画线所覆盖。

3.3 DRL 是否有失公平

因为设备都是相互竞争地接入 AP 请求 MEC 计算卸载服务,针对 CW 值的优化算法是否会导致接入设备的不公平竞争也是需要考虑的问题。该文使用 Jain's fairness index[24] 来评价网络的公平性,计算公式如下($0 < F_{\text{index}} \leqslant 1$):

$$F_{\text{index}} = \frac{\left(\sum_{i=0}^{n} T_i\right)^2}{n \times \sum_{i=0}^{n} T_i^2} \tag{5}$$

其中,T_i 为第 i 个设备的平均网络吞吐量,n 为设备的总个数。

实验计算结果如图 7 所示。

从图 7 的实验结果来看,DRL 方法并没有让网络失去公平性,几乎与标准的 BEA 保持一致。

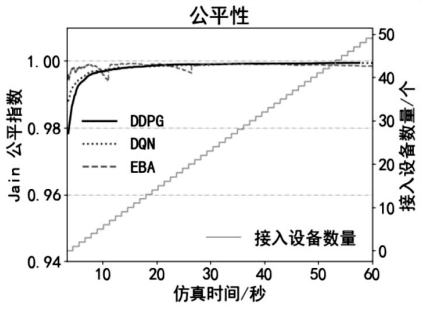

图 7　动态网络拓扑下各方法的公平性

4　结束语

该文提出两种 DRL:DQN 和 DDPG 方法,用于解决 IoT 的 MEC 卸载网络场景下面对大量接入设备时使用标准 EBA 导致网络性能下降的问题。提出针对 Wi-Fi 作为 MEC 卸载网络 AP 的 CW 值优化算法,应用 DRL 方法训练 CW 优化算法后在不同网络拓扑下可以有效地设置合理的 CW 值以稳定网络的吞吐量。实验结果表明,DRL 可以解决 CW 优化问题:即使在快速变化的网络拓扑结构下,两种算法都具有稳定网络性能的能力,DDPG 连续动作的方法略优于离散动作的 DQN。最后,实验在静态和动态网络拓扑环境下的结果都表明了 DRL 方法优于默认的退避算法,且没有破坏网络的公平性。

参考文献:

[1] SABELLA D, SUKHOMLINOV V, TRANG L, et al. Developing software for multi-access edge computing[J]. ETSI White Paper, 2019, 20: 1-38.

[2] 章坚武,王路鑫,孙玲芬,等. 人工智能在 5G 系统中的应用综述[J]. 电信科学,2021,37(5):14-31.

[3] KHOROV E, KIRYANOV A, LYAKHOV A, et al. A tutorial on IEEE 802.11 ax high efficiency WLANs[J]. IEEE Communications Surveys & Tutorials, 2018, 21(1): 197-216.

[4] BELLALTA B, KOSEK-SZOTT K. AP-initiated multi-user transmissions in IEEE 802.11 ax WLANs[J]. Ad Hoc Networks, 2019, 85: 145-159.

[5] GALLO P, KOSEK-SZOTT K, SZOTT S, et al. CADWAN: a control architecture for dense WiFi access networks[J]. IEEE Communications Magazine, 2018, 56(1): 194-201.

[6] 刘宴兵,杨茜惠,孙世新. IEEE 802.11 宽带无线局域网负载均衡优化研究[J]. 计算机应用研究,2008,25(7):2135-2137.

[7] KARACA M, BASTANI S, LANDFELDT B. Modifying backoff freezing mechanism to optimize dense IEEE 802.11 networks[J]. IEEE Transactions on Vehicular Technology, 2017, 66(10): 9470-9482.

[8] WILHELMI F, BARRACHINA-MUÑOZ S, BELLALTA B, et al. A flexible machine-learning-aware architecture for future wlans[J]. IEEE Communications Magazine, 2020, 58(3): 25-31.

[9] SANDHOLM T, HUBERMAN B, HAMZEH B, et al. Learning to wait: Wi-Fi contention control using load-based predictions[J]. arXiv: 1912.06747, 2019.

[10] 刘婷婷,杨晨阳,索士强,等. 无线通信中的边缘智能[J]. 信号处理,2020,36(11):1789-1803.

[11] 李本亮,王厚军,师奕兵,等. IEEE 802.11 的 DCF 机制媒介接入延时分析与仿真[J]. 计算机应用研究,2009,26(6):2202-2204.

[12] 桂　冠,王　禹,黄　浩. 基于深度学习的物理层无线通信技术:机遇与挑战[J]. 通信学报,2019,40(2):19-23.

[13] YAO F, JIA L. A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks[J]. IEEE Wireless Communications Letters, 2019, 8(4): 1024-1027.

[14] YU Y, WANG T, LIEW S C. Deep-reinforcement learning multiple access for heterogeneous wireless networks[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(6): 1277-1290.

[15] YE X, YU Y, FU L. Deep reinforcement learning based MAC protocol for underwater acoustic networks[J]. IEEE Transactions on Mobile Computing, 2020, 39: 1-5.

[16] 朱　江,王婷婷,宋永辉,等. 无线网络中基于深度 Q 学习的传输调度方案[J]. 通信学报,2018,39(4):35-44.

[17] HUANG C, MO R, YUEN C. Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(8): 1839-1850.

[18] ALI R, SHAHIN N, ZIKRIA Y B, et al. Deep reinforcement learning paradigm for performance optimization of channel observation - based MAC protocols in dense WLANs[J]. IEEE Access, 2018, 7: 3500-3511.

[19] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[20] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. arXiv: 1509.02971, 2015.

[21] 姚　程,俞能海,王　松. SCWGF:802.11 DCF 竞争窗口增长因子自适应调整算法[J]. 电子学报,2009,37(10):2134-2138.

[22] 李焕焕,彭盛亮,陈　铮,等. 认知无线电中基于 LSTM 网络的 MAC 协议识别[J]. 信号处理,2019,35(5):837-842.

[23] GAWŁOWICZ P, ZUBOW A. Ns3-gym: extending openai gym for networking research[J]. arXiv: 1810.03943, 2018.

[24] 罗庆云,陈　敏,赵巾帼. 基于均衡算法的协作信道分配策略[J]. 计算机科学,2013,40(4):96-101.