基于强化学习的802.11ax上行链路调度算法

黄新林 郑人华*

(同济大学电子与信息工程学院 上海 201800)

摘 要：随着物联网(IoT)时代的到来，无线网络饱和的问题已经越来越严重。为了克服终端密集接入问题，IEEE标准协会(IEEE-SA)制定了无线局域网的最新标准——IEEE 802.11ax。该标准使用正交频分多址(OFDMA)技术对无线信道资源进行了更细致的划分，划分出的子信道被称为资源单元(RU)。为解决密集用户环境下802.11ax 上行链路的信道资源调度问题，该文提出一种基于强化学习的RU调度算法。该算法使用演员-评论家(Actor-Critic)算法训练指针网络，解决了自适应RU调度问题，最终合理分配RU资源给各用户，兼具优先级和公平性的保障。仿真结果表明，该调度算法在IEEE 802.11ax上行链路中比传统的调度方式更有效，具有较强的泛化能力，适合应用在密集用户环境下的物联网场景中。

关键词：物联网；IEEE 802.11ax；强化学习；上行链路；演员-评论家

中图分类号：TN915; TP393 **文献标识码**：A **文章编号**：1009-5896(2022)05-1800-09

DOI: 10.11999/JEIT210590

802.11ax Uplink Scheduling Algorithm Based on Reinforcement Learning

HUANG Xinlin ZHENG Renhua

(*College of Electronic and Information Engineering*, *Tongji University*, *Shanghai* 201800, *China*)

Abstract: With the arrival of the Internet of Things (IoT) era, the problem of wireless network saturation has become more and more serious. In order to overcome this problem, the IEEE Standards Association (IEEE-SA) has formulated the latest standard for wireless local area networks—IEEE 802.11ax. In this standard, the Orthogonal Frequency Division Multiple Access (OFDMA) technology is utilized to divide wireless channel into several groups of tones, and the divided sub-channels are called Resource Units (RUs). In order to solve the channel resource scheduling problem of 802.11ax uplink in dense user environments, an RU scheduling algorithm based on reinforcement learning is proposed in this paper. The Actor-Critic algorithm is used to train the pointer network and solve the adaptive allocation problem of RU. Finally, RUs are allocated to each user reasonably with the guarantee of priority and fairness. The simulation results show that the scheduling algorithm is more effective than traditional scheduling methods in the IEEE 802.11ax uplink and has a strong generalization ability, which is suitable for the IoT scenario in dense user environments.

Key words: Internet of Things (IoT); IEEE 802.11ax; Reinforcement learning; Uplink; Actor-Critic

1 引言

如今，无线局域网(Wireless Local Area Network, WLAN)的用户数正不断增长。除此之外，物联网(Internet of Things, IoT)的迅猛发展也带来了大量需要接入无线网络的机器设备，导致在有限的地理区域内存在许多的接入点(Access Point, AP)和更多的站点(STAtion, STA)。同时，在物联网场景下，医疗、火警、交通等方面的传输业务相对于普通业务有更高的服务质量(Quality of Service, QoS)要求，上传数据时需要保证这些业务的优先级和实时性等。在上述的密集用户环境(dense user environments)中，来自相邻设备的干扰增加以及来自信道争用的严重冲突导致网络性能下降，无法提供良好的用户体验。因此，IEEE标准协会(IEEE Standards Association, IEEE-SA)标准委员会于2014年3月批准了802.11ax[1]协议以提高每个用户的

收稿日期：2021-06-17；改回日期：2022-01-16；网络出版：2022-02-02

*通信作者： 郑人华 471539350@qq.com

基金项目：国家自然科学基金(62071332)，上海市青年科技启明星计划(19QA1409100)，中央高校基本科研业务费专项资金

Foundation Items: The National Natural Science Foundation of China (62071332), Shanghai Rising-Star Program (19QA-1409100), The Fundamental Research Funds for the Central Universities

平均吞吐量并应对密集接入问题。802.11ax的上行链路包括两种接入方式：随机接入(Random Access, RA)和调度接入(Scheduled Access, SA)[2]。为了保证物联网设备上传数据的准确性和传输的低延时，需要减少冲突，使用调度接入是更好的选择。本文研究的重点是正交频分多址(Orthogonal Frequency Division Multiple Access, OFDMA)技术和802.11ax上行链路的调度接入问题。

调度接入算法并非新的研究方向，长期演进(Long Term Evolution, LTE)已经使用了OFDMA技术并且对调度问题进行了深入的研究[3]，调度算法包括首次最大扩展(First Maximum Expansion, FME)、递归最大扩展(Recursive Maximum Expansion, RME)等[4]。使用运营商频段的LTE，可以进行时域和频域两个维度的调度，即可以对每个资源块(Resource Block, RB)进行单独分配。802.11ax则不行，因为它还保留了载波侦听多路访问/冲突避免(Carrier Sense Multiple Access with Collision Avoid, CSMA/CA)技术。该技术已经固定了管理帧、数据帧和帧间隔的时间长短，所以无法实现时域维度的调度。除此之外，802.11ax上行链路调度的子信道资源，即资源单元(Resource Unit, RU)[5]的大小可变，使得LTE的调度算法难以迁移到802.11ax中。

802.11ax中上行链路的传输效率很大程度上取决于这些RU的调度方式。标准中提供了灵活的框架，却没有定义任何调度算法，这给本文的研究带来了可能性。Bankov等人[6]提出了一种通用的方法，可用于使现有的LTE调度器能适应802.11ax的特性，并证明OFDMA是在密集用户环境中提供高质量服务的关键技术。Wang等人[7]针对实际的上行链路RU调度问题，提出了两种实用算法：贪婪算法和递归算法，仿真结果表明递归调度非常接近最优调度。

以上的研究均没有关注密集用户环境下的物联网场景，并且泛化能力不足。物联网场景具有接入量大、实时性要求高、低功耗等特点，不同业务具有不同的QoS要求。针对这个场景，本文提出一种基于强化学习的802.11ax上行链路调度算法。首先建立系统模型并将RU调度问题转化为0-1背包问题；然后引入指针网络模型并使用演员-评论家(Actor-Critic)[8]强化学习算法进行训练，增强算法的泛化能力；最后使用训练好的模型去调度RU资源，并在上行链路中进行仿真。仿真结果表明，在物联网场景下相比于经典的调度算法，本文算法具有更好的表现，能够保证各个STA的QoS要求和公平性，并且具有更好的稳定性和用户体验。

2　IEEE 802.11ax的上行链路

IEEE 802.11ax是一项WLAN标准，其标准草案由IEEE标准协会的TGax工作组制定。802.11ax制定之初所关注的就是密集用户环境，其设计思想与以往的802.11标准存在差异。由于非授权频段的资源有限，因此为了提高资源利用率从而克服密集接入问题，引入了OFDMA，双向多用户多输入多输出(Multi-User Multiple-Input Multiple-Output, MU-MIMO)等技术，并采取了最高支持1024-正交振幅调制(Quadrature Amplitude Modulation, QAM)的调制方式，基本服务集(Basic Service Set, BSS)着色等措施[9]。

802.11ax标准中的信道被划分成若干大小为78.125 kHz的子载波(tone)。一定数量的子载波构成了标准中的RU。根据子载波数的不同，RU可以分为7种，它们分别为：26 tones,52 tones,106 tones,242 tones,484 tones,996 tones和2×996 tones。因此，OFDMA将现有的802.11ax信道(大小包括20, 40, 80和160 MHz)划分为一个个包含特定数量子载波的RU。如图1所示，20 MHz信道可以被划为若干大小不同的RU，不考虑MU-MIMO的情况下最多可容纳9个STA同时上传数据。如何分配这些RU给各个STA的方案并未在标准中定义，这为找到改善频谱效率的优化调度算法提供了可能性。

如图2所示是实际系统中的802.11ax上行链路调度接入过程[10]。按照时间顺序，AP总共会向所有STA发送3个触发帧以获取特定的反馈信息或进行资源调度。首先，AP会发送类型为缓存状态报告轮询(Buffer Status Report Poll, BSRP)的触发帧1，请求各STA反馈缓存状态报告(Buffer Status Reports, BSR)信息，其中包含调度所需的缓存数据量和QoS值。然后，AP会使用调度算法计算每个STA应该如何分配RU。再发送多用户请求发送(Multi-User Request To Send, MU-RTS)帧，即触

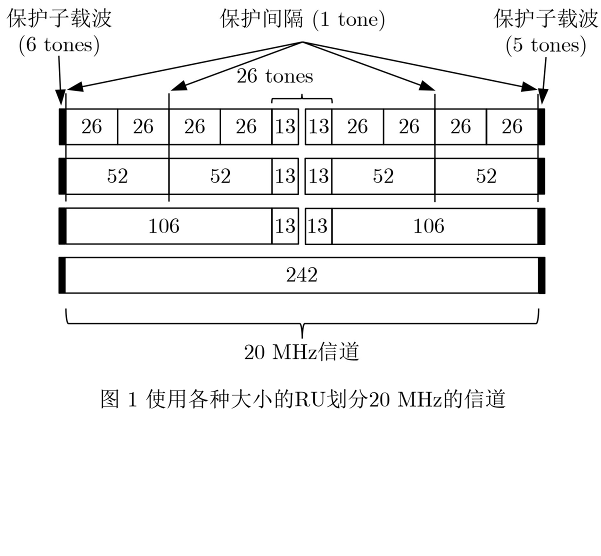


图 1 使用各种大小的RU划分20 MHz的信道

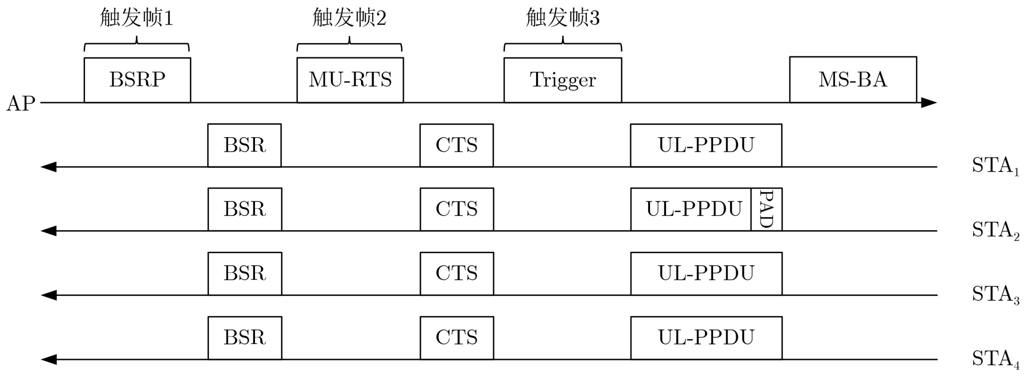


图 2 基于OFDMA的802.11ax上行链路调度接入过程

发帧2，来实际分配RU资源，从而避免上行链路冲突的发生，尽可能提高每个用户的吞吐量。各STA在接收到该帧后需要反馈准许发送(Clear To Send, CTS)帧，告知AP已知晓并认可当前的资源分配[11]。AP接收到CTS后会接着发送触发帧3，通知各STA开始在对应的RU上进行上行链路传输。值得注意的是，由于802.11ax的上行传输是基于帧的，所以当存在不同步的情况时，需要在数据帧最后添加PAD。最后，当上行链路的数据传输完成后，AP会向各STA发送多站点块确认(Multi-Station Block Acknowledgement, MS-BA)帧进行确认。

常用的调度算法有轮询算法和比例公平算法。Filoso等人[12]设计了一种基于比例的资源分配算法(Proportional-based Resource Allocation, PRA)，该算法利用各个STA上传的QoS信息和缓存数据量进行有效的RU分配，并同时考虑了优先级和公平性，但其算法结构固定，无法应对更为复杂的网络环境。而Bai等人[13]提出了一种自适应STA分组算法，该算法使用基于BSR的两阶段机制来克服IEEE 802.11ax面对的密集网络挑战。自适应分组算法虽然能够利用分组来有效避免冲突，减少系统能耗。但是其分组方案较为复杂，且每个分组会轮流使用信道，这导致它只能较好地保障公平性和组内优先级排布，而缺乏组与组之间优先级的保障。本文所提调度算法旨在将自适应RU调度问题抽象成背包问题，并使用指针网络模型和强化学习算法予以解决。最终让AP合理分配RU资源给各个STA，实现优先级和公平性的双重保障，并具备较强的泛化能力和调节能力。

3 基于强化学习的上行链路调度算法

在本节中，首先介绍本文建立的上行链路系统模型，再提出自适应RU调度问题，然后使用指针网络对问题进行建模，并利用Actor-Critic强化学习算法对指针网络进行训练，从而实现IEEE 802.11ax上行链路的RU调度。本文提出的基于强化学习的802.11ax上行链路调度算法，用于帮助AP给每个STA合理分配RU资源，以实现有效而公平的通信。

3.1 系统模型

本文建立的系统模型是由1个AP和n_{STA}个STA构成的物联网场景下的802.11ax网络，研究的数据传输路径是上行链路。本文的调度算法要使用到所有STA的BSR，其中包含它们的缓存数据量和QoS值。QoS值指示了STA业务的优先级，如表1所示。其数值越小对应的优先级就越高，对传输数据低延迟的要求也越高[14]。

为了便于仿真实验对系统模型进行一些简化：

(1) 仅考虑了调度接入的RU分配。

(2) 仅将1个RU分配给1个STA，不考虑MU-MIMO。

(3) 每次AP将RU分配给STA的数据流时，将占用一个完整的时间窗(T_{w})时间。

(4) 所有仿真使用相同的信道带宽。

3.2 自适应RU调度问题

对于以上系统模型，理想情况下，只要保证QoS值小(优先级高)的STA能优先获取RU，及时上传数据就能完成有效的通信。但对于实际场景并非如此，由于物联网场景下，连接数巨大，QoS值小的STA数量太多，可能会导致某些QoS值大(优先级低)的STA长期处于等待RU分配的状态，使得整个网络的公平性无法保障。

表 1 QoS值与业务类型对应关系

QoS	业务类型
1	探测请求、火灾报警、交通事故报警等
2	患者监测、工业设备监测等
3	智能家居、智慧农业、仓储管理等
4	监控视频、智能水表、智能电表等
5	信道质量指示符、无线电测量服务等

因此，为了评估调度算法能否既满足了QoS需求又保障了公平性，本文设计一种价值函数。根据STA上行数据流的数据量、QoS值和等待时间，共同计算该STA传输数据的价值，价值越高表示数据流在上行链路中的传输优先级越高。

索引为i的STA(STA_i)传输数据的价值函数如式(1)所示，式(2)、式(3)是约束条件

$$v_i = \frac{d_i}{\sum\limits_{1}^{n_{\mathrm{STA}}} d_i} \times \frac{\max(q) + 1 - q_i + \mathrm{wt}_i}{\max(q)} \tag{1}$$

$$i = 1, 2, \cdots, n_{\mathrm{STA}}; n_{\mathrm{STA}} \in \mathbb{N}_+ \tag{2}$$

$$d_i \ge 0, q_i \in \{\ 1,2,3,4,5\}\ , \mathrm{wt}_i \in \mathbb{N} \tag{3}$$

其中，d_i为STA_i数据流的数据量，q_i为STA_i的QoS值；wt_i为STA_i因没有被分配RU而等待传输的时间窗(T_{w})个数。

在系统模型中，STA需要在AP分配的RU上进行数据上传。定义$w_i \in \{26, 52, 106, 242, 484, 996\}$为AP需要分配给$\mathrm{STA}_i$的RU大小，$m_i \in \{1, 2, \cdots, 12\}$为$\mathrm{STA}_i$上传数据使用的调制与编码策略(Modulation and Coding Scheme, MCS)索引。不同RU大小，不同MCS情况下的数据传输速率如表2所示[6]。在确定MCS后可以根据d_i算出STA_i所需要的RU大小。举个例子，如果$m_i = 5$, $T_{\mathrm{w}} = 1$ ms, $d_i = 20$ kb，AP应该分配106 tones大小的RU给STA_i进行传输。

AP调度RU给接入网络的STA，以实现最有效的上行链路通信，需要同时考虑每个STA业务的价值函数和需要的RU大小。因此可以将自适应RU调度问题转化为背包问题，以便求解。

对于STA_i，所需RU大小分别为w_i，数据价值为v_i，而带宽大小为W。则调度问题转化为0-1背包问题，如式(4)—式(6)所示。目标是找到一种策略p把物品子集π放在背包里，使背包中物品的价值之和最大化，且满足重量之和小于或等于背包容量

$$\max\left(\sum_{i\in\pi} v_i = \sum_{i\in\pi} \frac{d_i}{\sum\limits_{1}^{n_{\mathrm{STA}}} d_i} \times \frac{\max(q) + 1 - q_i + \mathrm{wt}_i}{\max(q)}\right) \tag{4}$$

$$\text{s. t.} \sum_{i\in\pi} w_i \le W \tag{5}$$

$$\pi \subseteq \{1, 2, \cdots, n_{\mathrm{STA}}\} \tag{6}$$

3.3 指针网络模型

背包问题是一类经典的组合优化问题。在面对大规模的背包问题时，传统的启发式算法并不能保证得到最优解。因此本文提出使用指针网络模型来解决背包问题，并用强化学习算法来训练指针网络。

指针网络被证明是解决组合优化问题的有效方案，其结构如图3所示[15]。它由编码器和解码器两部分组成，两者都由长短期记忆网络(Long Short-Term Memory, LSTM)构成。编码时，本文将STA_i每个T_{w}内所需的RU大小w_i，数据价值v_i编码为2维的背包实例向量$\boldsymbol{c}_i = (w_i, v_i)$。然后将序列$\boldsymbol{c} = \{\boldsymbol{c}_i\}_{i=1}^{n_{\mathrm{STA}}}$输入到编码器中，其中$\boldsymbol{c}$是背包实例序列训练集$C^{\mathrm{in}}$的1次采样结果。解码时，指针网络每一步都使用巴达瑙(Bahdanau)注意力机制指向输入序列中要放入背包的实例，一旦背包中的重量w之和大于限定的重量容量W就停止解码器输出。

解码器使用的Bahdanau注意力机制[16]如式(7)—式(9)所示

$$u_i^j = \boldsymbol{v}^{\mathrm{T}} \tanh(\boldsymbol{W}_1 \boldsymbol{e}_i + \boldsymbol{W}_2 \boldsymbol{d}_j), i = 1, 2, \cdots, n_{\mathrm{STA}} \tag{7}$$

$$u_i^j = \mathrm{mask}(u_i^j) = \begin{cases} u_i^j, i \notin \pi(<j) \\ -\infty, 其它 \end{cases} \tag{8}$$

表 2 不同MCS与不同RU大小情况下的数据传输速率(Mbps)

MCS索引	MCS	26 tones	52 tones	106 tones	242 tones	484 tones	996 tones
1	BPSK, 1/2	0.8	1.7	3.5	8.1	46.3	34.0
2	QPSK, 1/2	1.7	3.3	7.1	16.3	32.5	68.1
3	QPSK, 3/4	2.5	5.0	10.6	24.4	48.8	102.1
4	16-QAM, 1/2	3.3	6.7	14.2	32.5	65.0	136.1
5	16-QAM, 3/4	5.0	10.0	21.3	48.8	97.5	204.2
6	64-QAM, 2/3	6.7	13.3	28.3	65.0	130.0	272.2
7	64-QAM, 3/4	7.5	15.0	31.9	73.1	146.3	306.3
8	64-QAM, 5/6	8.3	16.7	35.4	81.3	162.5	340.3
9	256-QAM, 3/4	10.0	20.0	42.5	97.5	195.0	408.3
10	256-QAM, 5/6	11.1	22.2	47.2	108.3	216.7	453.7
11	1024-QAM, 3/4	–	–	–	121.9	243.8	510.4

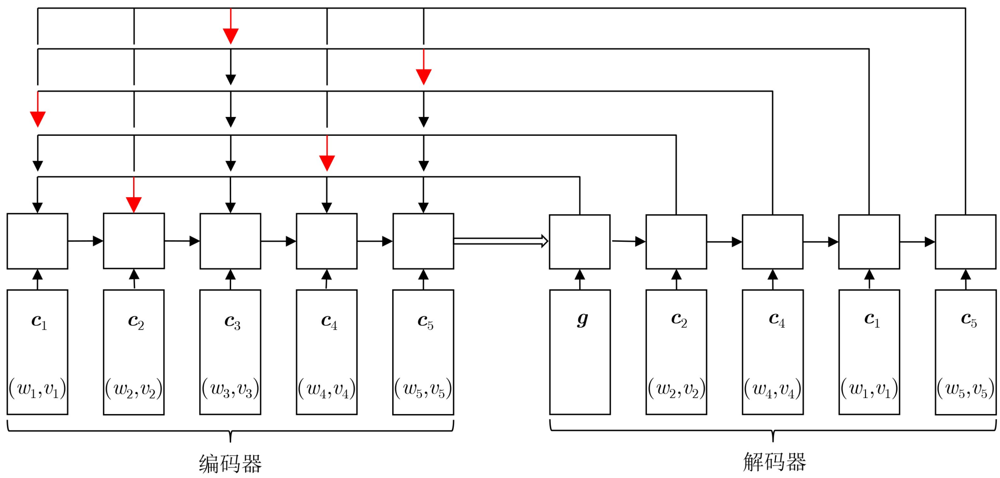


图 3 指针网络结构图

$$p(\pi(j)|\pi(<j),c)=\mathrm{softmax}(u^j) \tag{9}$$

其中，$\boldsymbol{e}_i$是编码器在第i步的隐藏状态，$\boldsymbol{d}_j$是解码器在第j步的隐藏状态，调度策略为p。而$\boldsymbol{v}$是向量，$\boldsymbol{W}_1$和$\boldsymbol{W}_2$是平方矩阵，它们都是可训练的参数。将u_i^j用作解码器在第j步指向输入序列中第i个实例的指针。同时为保证背包实例不会被重复选择，使用mask函数对u_i^j进行约束。然后通过softmax函数将长度为$n_{\rm STA}$的向量$\boldsymbol{u}^j$归一化为输入序列c上的概率分布，在按照这个概率分布在输入序列c中选择合适的背包实例作为此刻解码器的输出。换句话说输出序列完全来自输入序列，只是尽可能选择能让累计价值更大的背包实例。最后，解码器在第$j_{\rm end}$步时发现背包中的重量w之和将大于限定的重量容量W，因而停止输出。

3.4 指针网络训练过程

强化学习通过与环境互动不断改善策略，使得最终获得的累计奖励最大化[17]，是解决组合优化问题的重要机器学习算法。本文使用Actor-Critic算法来优化指针网络的参数θ，将指针网络作为Actor。在输入序列为c，策略为p的情况下定义解码器输出的价值之和为$V(\pi|c)$。因此，Actor的最大化目标函数为式(10)

$$J(\theta|c)=E_{\pi\sim p_\theta(.|c)}V(\pi|c) \tag{10}$$

为获得最优的调度策略需要最大化目标函数，使用策略梯度方法优化指针网络参数θ。根据REINFORCE算法，可以得出训练目标的梯度为式(11)。为保证价值越高的实例被选择的概率越大，需要采用随机梯度上升法来更新网络参数，如果采用梯度下降法更新，则式(11)要取相反数

$$\nabla_\theta J(\theta|c)=E_{\pi\sim p_\theta(.|c)}[(V(\pi|c)-b(c))\nabla_\theta \ln p_\theta(\pi|c)] \tag{11}$$

其中，$b(c)$是不依赖策略p的基线函数，目的是减小梯度的方差。通过蒙特卡罗方法可以对梯度进行近似，即使用N批次的随机采样序列$\{c_1,c_2,\cdots,c_N\}\in C^{\rm in}$作为指针网络输入，得到近似的梯度为

$$\nabla_\theta J(\theta)\approx\frac{1}{N}\sum_{i=1}^{N}(V(\pi_i|\boldsymbol{c}_i)-b(\boldsymbol{c}_i))\nabla_\theta \ln p_\theta(\pi_i|\boldsymbol{c}_i) \tag{12}$$

其中，$b(c_i)$是改善训练结果的重要函数。因此本文引入参数为θ_v的Critic网络来估计期望的背包价值。Critic包括3个部分：编码器，处理模块和解码器。编码器与指针网络的编码器类似，将输入序列c_i编码为一系列潜在的记忆状态和隐藏状态h。然后，处理模块使用glimpse函数[16] 更新隐藏状态h。最后使用2层ReLU神经网络解码器，将获得的隐藏状态h解码为基线预测$b_{\theta_v}(\boldsymbol{c}_i)$。Critic使用随机梯度下降法进行参数更新，网络训练的损失函数为

$$L(\theta_v)=\frac{1}{N}\sum_{i=1}^{N}\|b_{\theta_v}(\boldsymbol{c}_i)-V(\pi_i|\boldsymbol{c}_i)\|_2^2 \tag{13}$$

最终，使用Actor-Critic算法训练指针网络的过程如表3所示。

3.5 算法实际调度过程

与指针网络训练过程类似，构建新的指针网络模型，将训练好的参数θ导入到新的模型中，形成调度策略$p^*(\pi^*|c)$。然后在每个时间窗中，将各个STA的数据流编码为2维向量$\boldsymbol{c}_i=(w_i,v_i)$。再将$n_{\rm STA}$个站点组成的背包实例序列$\boldsymbol{c}=\{\boldsymbol{c}_i\}_{i=1}^{n_{\rm STA}}$输入到指针网络的编码器中，通过解码器结合注意力机制输出选择的背包实例序列$\pi^*(<j_{\rm end})$，即选出当前时间窗内需要分配RU的数据流。最后在802.11ax的上行链路中，AP使用MU-RTS帧将RU调度信息发送给所有的STA，只有被指针网络解码器选中的数据流才能在相应的RU上进行上行传输。

表 3　Actor-Critic算法训练指针网络的过程

(1) 初始化超参数，初始化训练集C^{in}，设置训练总步长T，设置批次数N

(2) 初始化指针网络参数θ

(3) 初始化Critic网络参数θ_v

(4) for t = 1 to T：

(5) 从训练集中获取输入：

$c_i \sim \text{SampleInput}(C^{\text{in}})$ for $i \in \{1,2,\cdots,N\}$

(6) 使用θ选出物品子集：

$\pi_i \sim \text{SampleSolution}(p_\theta(.|c_i))$ for $i \in \{1,2,\cdots,N\}$

(7) 使用θ_v计算基线值：

$b(c_i) = b_{\theta_v}(\boldsymbol{c}_i)$ for $i \in \{1,2,\cdots,N\}$

(8) 计算Actor目标函数的梯度：

$$\nabla_\theta J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(V(\pi_i|\boldsymbol{c}_i) - b(c_i))\nabla_\theta \ln p_\theta(\pi_i|\boldsymbol{c}_i)$$

(9) 计算Critic的损失函数：

$$L(\theta_v) = \tfrac{1}{N}\sum_{i=1}^{N} \| \, b_{\theta_v}(\boldsymbol{c}_i) - V(\pi_i|\boldsymbol{c}_i) \, \|_2^2$$

(10) 使用Adam优化器对参数θ进行更新：

$\theta = \text{Adam}(\theta, \nabla_\theta J(\theta))$

(11) 使用Adam优化器对参数θ_v进行更新：

$\theta_v = \text{Adam}(\theta_v, \nabla_{\theta_v} L(\theta_v))$

(12) end

4　仿真实验与对比分析

本文使用pytorch搭建了用于求解802.11ax上行链路自适应RU调度问题的指针网络，并使用Actor-Critic算法对其进行了训练，最终实现了基于强化学习的上行链路调度算法，在指针网络中设置输入层大小为2，隐藏层大小l为128，训练采用Adam优化器，学习率设定为10^{-4}。同时，为模拟真实场景，在MATLAB中使用WLAN工具箱搭建了802.11ax上行链路仿真模型，并按照实际系统设置参数。在仿真模型中设置信道带宽(背包大小)为20 MHz，并且采用TGax NLOS室内信道模型；时间窗大小T_{w}为1 ms；STA数量n_{STA}为120；STA的天线数为4，但未使用MU-MIMO技术；信道编码方式则采用LDPC。仿真过程中，将超参数、指针网络参数和各STA的等待时间等参数直接存储于AP中。并将指针网络模型加载到AP中。同时，在RU分配之前，调度所需的缓存数据量和QoS值会由各STA上传给AP。保证了AP在每次调度RU资源之前都能即时获得需要的所有参数。而在上行传输结束后，还可以根据AP实际接收的数据得到各STA的吞吐量和数据流价值，便于算法测试以及对比分析。

本文算法的仿真结果如图4所示。图4展示了其中5个STA在上行传输中吞吐量随时间变化的结果，用于分析QoS值在本文算法中的影响。每个STA的总缓存数据量为37 MB，MCS都为7，因为MCS相同所以吞吐量可以直接反映出占用RU资源的大小。STA_1, STA_{21}, STA_{41}, STA_{61}, STA_{81}的QoS值分别为1, 2, 3, 4, 5。由于这5个STA涵盖了所有可选的QoS值，将它们作为其余QoS值与之相同STA的代表，便于展示。从图4可以看出，QoS值较小的STA能够优先获得AP调度的RU资源，并且RU的平均大小也比QoS值较大的站点更大，所以平均吞吐量较高，因此能够在相同缓存数据量的情况下，更早地结束上行传输。同时，QoS值较大的STA虽然优先级较小，但并未产生“饿死”现象，虽然平均吞吐量较低，但也能被AP分配到适当的RU资源。并且从图4中30 s之后展示的曲线来看，QoS值较大的STA在优先级较高STA的信息传输完毕后，还能在本文算法的调度下立即获取更大的RU资源，保持了较高的信道利用率，没有造成资源浪费。因此，可以总结：本文算法既能够保证QoS值较小的STA可以优先传输大量数据、占用更大的RU资源，又能保证整个上行链路的公平性，不会造成“饿死”现象的发生。

IEEE 802.11ax中的调度算法有轮询调度算法、PRA算法和自适应分组算法。因此，接下来会将本文所提上行链路调度算法与3者相比较，分析优劣与适用场景。

如图5所示，是4种算法对于$\text{STA}_1(q_1=1, m_1=7)$和$\text{STA}_{63}(q_{63}=4, m_{63}=5)$的RU调度情况在吞吐量上的直接反映，$\text{STA}_1$和$\text{STA}_{63}$分别代表了QoS值较小和QoS值较大的两种STA。从图5(a)可以看出相比于另外3种调度算法，本文算法能够优先且稳定地分配更大的RU资源给QoS值较小的STA，保证

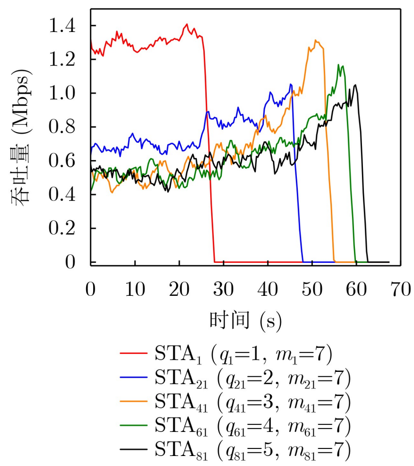
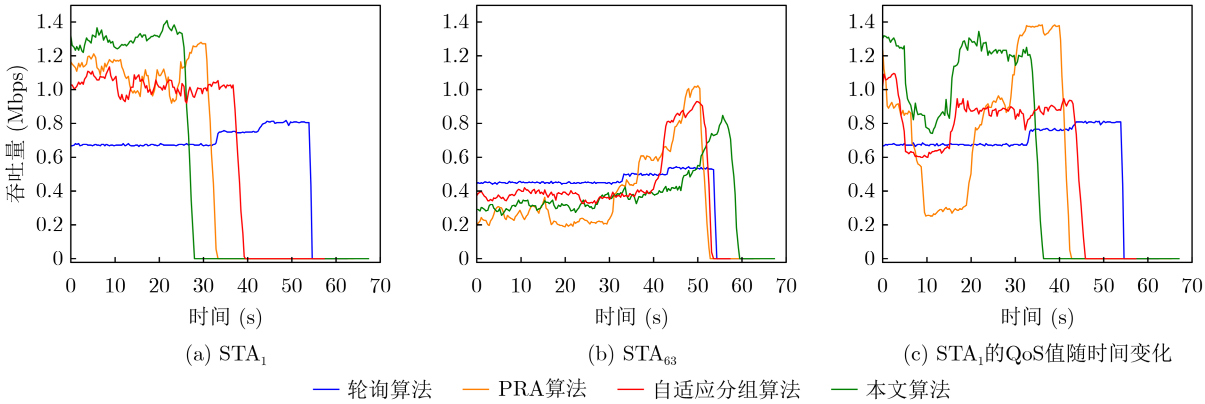


图 4 本文算法的吞吐量随时间变化的仿真结果



(a) STA_1 (b) STA_{63} (c) STA_1的QoS值随时间变化

图 5 4种算法下STA_1和STA_{63}的吞吐量随时间变化的仿真结果

业务优先级高数据流的低延时和高吞吐，让STA_1在30 s以内就能将数据上传完毕，早于其他算法。而在图5(b)也可以看出，本文算法相比于另外3种算法略微延长了QoS值较大的STA_{63}的传输时间。虽然STA_{63}的平均吞吐量较小，但并未出现“饿死”现象，说明本文算法保证了STA之间的公平性，且调度RU资源过程相较于PRA算法更稳定。图5(c)则展示了各算法在STA业务类型随时间变化时的适应能力。图中STA_1的QoS值在最初5 s内为1，5～15 s变为5，15 s后一直保持为2，而其余STA的业务类型保持不变。从仿真结果可以发现除轮询算法以外的3种算法都具有QoS自适应能力，且本文算法与自适应分组算法的业务类型跟踪能力较PRA算法更优秀，调节更及时，吞吐量波动更小，传输更稳定。

4种算法的仿真结果中，各个STA的平均等待时间也有所不同。如表4所示是5个STA代表在4种算法中因AP连续未分配RU资源而等待的平均时间。总体来看，轮询算法和自适应分组算法的平均等待时间较公平，即不具备优先级区分。自适应分组算法组内具有一定优先级保障，组间则没有，其平均等待时间与分组数正相关。而本文算法和PRA算法都对各STA实现了优先级划分，且本文算法平均等待时间更短。说明本文算法兼具优先级和公平性保障，既能让优先级高的业务平均等待时间更短，又能保证不同STA的平均等待时间相差不大。即实现了更稳定而公平的RU调度。

表 4 4种算法下5个STA代表的平均等待时间(ms)

算法名	STA_1	STA_{21}	STA_{41}	STA_{61}	STA_{81}
轮询算法	8.73	8.83	8.73	8.60	9.01
PRA算法	5.42	7.36	10.87	13.84	16.90
自适应分组算法	9.10	9.14	9.12	9.13	9.61
本文算法	4.49	5.65	7.97	9.31	11.56

图6展示了在4种算法的调度过程中，上行链路数据流总价值随时间变化的情况。可以发现，本文算法在调度开始的55 s内，上行数据流的总价值都是高于另外3种算法的，而轮询算法、PRA算法和自适应分组算法的总价值平均来看基本持平。除此之外，若将整个仿真时间段内的总价值取均值，可以得到对4种算法的总体评价。为了检验本文算法模型的泛化能力，将802.11ax网络的STA数量$n_{\rm STA}$依次设置为80, 100, 120, 140, 160，并进行相同的对比实验，仿真结果如图7所示。可以看出本文算法模型虽然只按照$n_{\rm STA}=120$进行训练，但是训练后的模型可以应用到STA数量更多或更少的场景中，并且本文算法在各场景中的表现依然优于另

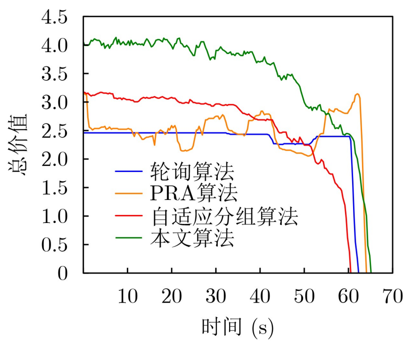


图 6 4种算法上行链路数据流总价值随时间变化的仿真结果

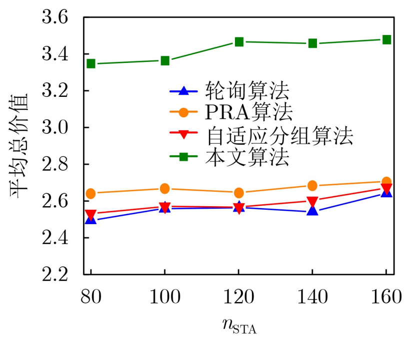


图 7 4种算法上行链路数据流平均总价值与STA数量的关系

外3种调度算法，足以证明本文算法的有效性和优秀的泛化能力。

综上所述，本文的调度算法对优先级的保障是优于轮询算法、PRA算法和自适应分组算法的，并且相较于PRA算法更加稳定和公平，有效增强了802.11ax上行链路的传输性能。虽然本文算法的复杂度相较于另外3者略高：由于需要先编码再解码所以时间复杂度为$O(2n_{\mathrm{STA}})$，另外3种算法则为$O(n_{\mathrm{STA}})$；AP需要额外存储指针网络的参数所以空间复杂度为$O(3n_{\mathrm{STA}}+2l)$，轮询和自适应分组算法则为$O(n_{\mathrm{STA}})$，PRA算法为$O(2n_{\mathrm{STA}})$。对于结构较简单、计算能力较差的AP来说不太适合。但是在密集用户环境下的物联网场景中，本文算法具有更好的表现，更能满足AP调度RU资源时对公平性和有效性的需求。

5　结束语

为解决密集用户环境下802.11ax 上行链路的自适应RU调度问题。针对物联网场景，本文提出了一种基于强化学习的802.11ax上行链路调度算法。该算法应用在WLAN的AP上，用于向所有接入该AP的STA调度RU资源。首先，根据物联网场景的特点建立通信系统模型；然后，在系统模型的基础上提出802.11ax上行链路的自适应RU调度问题，并将该问题转换为背包问题；之后，使用Actor-Critic强化学习算法训练用于解决背包问题的指针网络，并存储训练后的网络参数；最后，在仿真平台上使用训练好的网络参数进行上行链路RU资源调度和数据传输实验。通过与轮询算法、PRA算法和自适应分组算法对比可以发现，本文算法在密集用户环境下的物联网场景中表现更优，相比于其他调度算法更能满足优先级和公平性的需求，分配资源的过程也更加稳定而有效，同时算法所训练的模型具有很好的泛化能力，对于实际应用场景有一定的实用价值。

参 考 文 献

[1] LEE J. OFDMA-based hybrid channel access for IEEE 802.11ax WLAN[C]. 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 2018: 188–193. doi: 10.1109/IWCMC.2018.8450369.

[2] BHATTARAI S, NAIK G, and PARK J M J. Uplink resource allocation in IEEE 802.11ax[C]. ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019: 1–6. doi: 10.1109/ICC.2019.8761594.

[3] PIRO G, GRIECO L A, BOGGIA G, *et al*. Two-level downlink scheduling for real-time multimedia services in LTE networks[J]. *IEEE Transactions on Multimedia*, 2011, 13(5): 1052–1065. doi: 10.1109/TMM.2011.2152381.

[4] SAFA H and TOHME K. LTE uplink scheduling algorithms: Performance and challenges[C]. 2012 19th International Conference on Telecommunications (ICT), Jounieh, Lebanon, 2012: 1–6. doi: 10.1109/ICTEL.2012.6221230.

[5] KARTHIK R M and PALANISWAMY S. Resource unit (RU) based OFDMA scheduling in IEEE 802.11ax system[C]. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018: 1297–1302. doi: 10.1109/ICACCI.2018.8554931.

[6] BANKOV D, DIDENKO A, KHOROV E, *et al*. OFDMA uplink scheduling in IEEE 802.11ax Networks[C]. 2018 IEEE International Conference on Communications (ICC), Kansas City, USA, 2018: 1–6. doi: 10.1109/ICC.2018.8422767.

[7] WANG Kaidong and PSOUNIS K. Scheduling and Resource Allocation in 802.11ax[C]. IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, USA, 2018: 279–287. doi: 10.1109/INFOCOM.2018.8486204.

[8] 唐伦, 贺小雨, 王晓, 等. 基于迁移演员-评论家学习的服务功能链部署算法[J]. 电子与信息学报, 2020, 42(11): 2671–2679. doi: 10.11999/JEIT190542.
TANG Lun, HE Xiaoyu, WANG Xiao, *et al*. Deployment algorithm of service function chain based on transfer actor-critic learning[J]. *Journal of Electronics & Information Technology*, 2020, 42(11): 2671–2679. doi: 10.11999/JEIT190542.

[9] AFAQUI M S, GARCIA-VILLEGAS E, and LOPEZ-AGUILERA E. IEEE 802.11ax: Challenges and requirements for future high efficiency WiFi[J]. *IEEE Wireless Communications*, 2017, 24(3): 130–137. doi: 10.1109/MWC.2016.1600089WC.

[10] MACHROUH Z and NAJID A. High efficiency WLANs IEEE 802.11ax performance evaluation[C]. 2018 International Conference on Control, Automation and Diagnosis (ICCAD), Marrakech, Morocco, 2018: 1–5. doi: 10.1109/CADIAG.2018.8751296.

[11] ZHOU Hu, LI Bo, YAN Zhongjiang, *et al*. An OFDMA

based multiple access protocol with QoS guarantee for next generation WLAN[C]. 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Ningbo, China, 2015: 1–6. doi: 10.1109/ICSPCC.2015.7338845.

[12] FILOSO D G, KUBO R, HARA K, *et al*. Proportional-based resource allocation control with QoS adaptation for IEEE 802.11ax[C]. ICC 2020-2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020: 1–6. doi: 10.1109/ICC40277.2020.9149111.

[13] BAI Jiyang, FANG He, SUH J, *et al*. An adaptive grouping scheme in ultra-dense IEEE 802.11ax network using buffer state report based two-stage mechanism[J]. *China Communications*, 2019, 16(9): 31–44. doi: 10.23919/JCC.2019.09.003.

[14] DUAN Ren, CHEN Xiaojiang, and XING Tianzhang. A QoS architecture for IOT[C]. 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Dalian, China, 2011: 717–720. doi: 10.1109/iThings/CPSCom.2011.125.

[15] VINYALS O, FORTUNATO M, and JAITLY N. Pointer networks[J]. arXiv: 1506.03134, 2015.

[16] BELLO I, PHAM H, LE Q V, *et al*. Neural combinatorial optimization with reinforcement learning[J]. arXiv: 1611.09940, 2017.

[17] 李晨溪, 曹雷, 陈希亮, 等. 基于云推理模型的深度强化学习探索策略研究[J]. 电子与信息学报, 2018, 40(1): 244–248. doi: 10.11999/JEIT170347.

LI Chenxi, CAO Lei, CHEN Xiliang, *et al*. Cloud reasoning model-based exploration for deep reinforcement learning[J]. *Journal of Electronics & Information Technology*, 2018, 40(1): 244–248. doi: 10.11999/JEIT170347.

黄新林：男，1985年生，教授，博士生导师，研究方向为机器学习与智能通信.

郑人华：男，1996年生，硕士生，研究方向为强化学习与智能通信.

责任编辑：余 蓉