◎网络、通信与安全◎

基于强化学习的SDN路由优化算法

车向北[1],康文倩[1],欧阳宇宏[1],杨柯涵[2],李 剑[2]
1.深圳供电局有限公司,广东 深圳 510800
2.北京邮电大学 计算机学院,北京 100876

摘 要:针对SDN控制器中网络路由的优化问题,基于强化学习中的PPO模型设计了一种路由优化算法。该算法可以针对不同的优化目标调整奖励函数来动态更新路由策略,并且不依赖于任何特定的网络状态,具有较强的泛化性能。由于采用了强化学习中策略方法,该算法对路由策略的控制相比各类基于Q-learning的算法更为精细。基于Omnet++仿真软件通过实验评估了该算法的性能,相比传统最短路径路由算法,路由优化算法在Sprint结构网络上的平均延迟和端到端最大延迟分别降低了29.3%和17.4%,吞吐率提高了31.77%,实验结果说明了基于PPO的SDN路由控制算法不仅具有良好的收敛性,而且相比静态最短路径路由算法与基于Q-learning的QAR路由算法具有更好的性能和稳定性。

关键词:软件定义网络;强化学习;SDN路由优化
文献标志码:A **中图分类号:**TP393.0 doi:10.3778/j.issn.1002-8331.2003-0423

SDN Routing Optimization Algorithm Based on Reinforcement Learning

CHE Xiangbei[1], KANG Wenqian[1], OUYANG Yuhong[1], YANG Kehan[2], LI Jian[2]
1.Shenzhen Power Supply Bureau Co., Ltd., Shenzhen, Guangdong 510800, China
2.School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: Aiming at the network routing optimization in SDN controller, a routing optimization algorithm is designed based on the PPO model in reinforcement learning. The algorithm can adjust the reward function for different optimization goals to dynamically update the routing strategy, and this algorithm does not depend on any specific network state and has very good generalization performance. Because of adopting the strategy method in reinforcement learning, the control of routing strategy is more elaborate than various Q-learning-based algorithms. Based on Omnet++ simulation software, the performance of the algorithm is evaluated through experiments. Compared with the traditional shortest path routing algorithm, the average delay and end-to-end maximum delay of this routing optimization algorithm on the Sprint structure network are reduced by 29.3% and 17.4%, respectively and throughput rate is increased by 31.77%. The experimental result shows that this PPO-based SDN routing control algorithm not only has good convergence, but also has better performance and stability than the shortest path routing algorithm and the Q-learning based QAR routing algorithm.

Key words: software-defined network; reinforcement learning; SDN routing optimization

软件定义网络(Software Defined Network,SDN)打破了传统网络模型的垂直结构,将控制与转发进行了分离,为网络的创新与演变提供了更多可能,被认为是未来网络的主要架构,但近年来,网络规模不断扩大,网络流量也呈现指数形式增长,而对网络控制的要求也变得日益精细。传统的网络路由方案主要基于最短路径算法来计算,存在收敛速度慢,难以处理网络拥塞等问题。因此,设计一种通过SDN控制器来对网络路由进

基金项目:国家自然科学基金(U1636106);北京市自然科学基金(4182006)。
作者简介:车向北(1984—),男,高级工程师,研究领域为电力监控系统网络安全,E-mail:chexiangbei@163.com;康文倩(1988—),女,工程师,研究领域为电力监控系统网络安全;欧阳宏宇(1993—),男,助理工程师,研究领域为电力监控系统网络安全;杨柯涵(1996—),男,硕士研究生,研究领域为强化学习、计算机网络;李剑(1976—),男,教授,博士生导师,CCF会员,研究领域为量子信息、人工智能、软件定义网络等。
收稿日期:2020-03-27 **修回日期:**2020-06-22 **文章编号:**1002-8331(2021)12-0093-06

行选路的高效优化算法是保证网络服务以及推动SDN发展的关键因素。

机器学习特别是深度学习技术在大规模数据处理，分类和智能决策方面的出色表现而引起了广泛关注。在SDN网络中，有许多研究使用它来解决网络运营和管理中的问题[1-2]。Li等人在文献[3]中提出了一种基于机器学习的路线预设计方案。这种方法首先使用聚类算法（例如高斯混合模型或者K-means模型）来提取网络流量特征，然后使用监督学习方法（例如极限学习机）来预测流量需求，最后使用一种基于层次分析的自适应动态算法来处理流量路由问题。Wang等人在文献[4]中提出了一种启发式算法来优化SDN路由，但当网络变化时，该算法的性能并不稳定。也有研究使用诸如蚁群算法和遗传算法等启发式算法来优化路由选择问题[5-6]。但这些算法泛化性能较差，当网络状态变化时，这些启发式算法的参数也需要进行相关调整，算法难以稳定工作。

强化学习通过不断与环境交互，能够进行动态的决策管理，因此也常被用来解决路由优化问题。文献[7]中的工作针对网络服务质量指标建立奖励函数使用Q-Learning方法进行优化。文献[8]中的工作提出了一种端到端的自适应HTTP流媒体智能传输架构，该架构基于部分可观测的马尔可夫决策过程进行建模，也采用了基于Q-Learning的决策算法。这些基于Q-Learning[9]的强化学习算法需要对Q表求解以进行控制决策。而SDN网络状态空间十分巨大，基于Q-Learning的算法并不能很好地对状态进行描述。同时，Q-Learning作为强化学习中的值方法，输出的控制动作仅在离散动作空间内工作，决策动作空间十分有限。因此，设计一种能对SDN网络进行细粒度的分析和控制的动态路由策略是一个巨大的挑战。

针对SDN控制平面中的路由策略，本文将引入近端最优算法（Proximal Policy Optimization，PPO），根据此算法来对SDN控制平面中的路由方案进行决策。该算法具有以下优点：首先，该算法使用神经网络来对SDN网络状态的Q值进行精确计算，避免了Q表带来的局限性和低效性问题。同时，该算法属于强化学习中的策略方法，能够对网络决策输出更加细粒度的控制方案。最后，该算法可以根据不同的优化目标调整强化学习奖励函数来动态优化路由策略。由于这种算法不依赖于任何特定的网络状态，具有非常好的泛化性能，这种算法也有效地实现了黑盒优化。基于Omnet++仿真软件通过实验评估了该算法的性能。实验结果表明，本文提出的基于PPO的SDN路由控制算法不仅具有良好的收敛性，而且比传统的基于最短路径静态路由算法与文献[7]中提出的QAR算法具有更好的性能和稳定性。

1 背景及相关技术

1.1 SDN与知识平面网络

传统互联网架构主要以OSI七层或者TCP/IP四层协议模型为主，各层网络设备之间通过相应的网络协议（交换、路由、标签、安全等协议）来进行数据传递。大体工作流程都是按照：邻居建立—信息共享—路径选择三个步骤来实现。另外，网络设备之间传递信息采用典型的分布式架构，设备之间以“接力棒”的形式交互信息，然后建立数据库信息，再依据相关路径算法（如Dijkstra最短路径算法）传递数据。各层设备独立计算，有独立的控制器和转发硬件，通过协议来进行沟通。

这种分布式架构在协议规范不完整的过去促进了互联网的蓬勃发展。但随着现今通信设备协议等逐步统一完善，分布式架构已逐渐到达瓶颈，凸显出诸多问题，例如传输表信息冗余、流量难以控制、设备无法自定义传输等。出现这些问题的根本原因在于传统架构中网络设备数据与控制相耦合，且设备不具有开放性的可编程接口，从而无法将数据转发与数据控制进行分离。

软件定义网络这一理念正是为了解决这一问题被提出，其核心思想是将网络上的所有信息集中到一个核心控制器（Controller）上，从而控制器能够采用集中式的方法直接操纵下层基础设备，从而处理整体网络的信息传输逻辑，并对应用软件提供可编程接口，这种方法能够为数据传递提供极大的灵活性，因此近年来也得到广泛应用[10-11]。

软件定义网络将控制平面与数据平面相分离，控制平面中的SDN控制器能够全感知到整个网络中的信息，从理论上可以为整体网络提供更加高效快速的路由方案。文献[12]中提出了一种知识平面网络的范式，在传统SDN架构的基础上添加了知识平面（Knowledge Plane），如图1所示，知识平面需要处理由下层平面收集得到的信息并利用机器学习方法来对网络管理进行决策，从而提高SDN的整体效率。

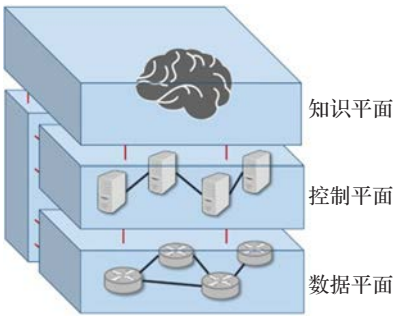

图1 KDN架构图

本文提出的路由策略优化算法也是基于此控制模型架构进行展开，将在知识平面引入强化学习方法来对数据平面的路由方案进行集中动态管理。

1.2 强化学习与PPO算法

强化学习是机器学习的一种范例,基于马尔可夫决策过程来学习决策智能体与环境之间的状态、动作和奖励的相互关系[9]。图2说明了强化学习之间决策主体与环境之间的交互过程。

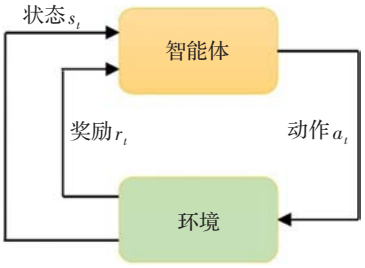

图2 强化学习中决策主体和环境的交互过程

PPO[13]是一种免模型的基于Actor-Critic架构的策略梯度算法,在TRPO算法[14]的基础上进行了优化。算法的目标在于获得一个能够使得以上交互过程中累计回报最大化的决策策略$\pi_\theta(s)$。其中s为模型的输入,即描述当前环境的一个向量,在网络控制中可以是整个网络的流量矩阵以及拓扑邻接矩阵等。决策函数输出动作,该动作在路由策略中可以是描述网络链路的权重,通过该权重可以唯一确定最优一种路由方案。决策函数可以用神经网络来进行拟合,θ即为网络参数。

策略梯度算法的工作原理是估计策略梯度,并利用梯度上升法来更新策略参数。通过在环境中运行策略以获取样本从而估算策略损失$J(\theta)$及其梯度[9]:

$$J(\theta)=E_{\tau\sim\pi_{\theta(\tau)}}[\sum_t R(s_t,a_t)]=E_{\tau\sim\pi_{\theta(\tau)}}[R(\tau)] \tag{1}$$

$$\nabla_\theta J(\theta)=E_{\tau\sim\pi_{\theta(\tau)}}[(\sum_{t=1}^{T}\nabla_\theta\ln\pi_\theta(a_t|s_t))R(\tau)] \tag{2}$$

策略梯度方法的主要挑战在于减小梯度估计的方差,从而可以朝着更好的策略进行优化。在实际中,通常使用Actor-Critic架构来进行优化,最终结果为:

$$Q^\pi(s,a)=\sum_t E[R(s_t,a_t)|s,a] \tag{3}$$

$$V^\pi(s)=\sum_t E_{\pi_\theta}[R(s_t,a_t)|s] \tag{4}$$

$$A^\pi(s,a)=Q^\pi(s,a)-V^\pi(s) \tag{5}$$

上式优势函数$A^\pi(s,a)$衡量了在状态下某项决策的好坏,因此算法最终的目标在于优化π以不断增大该优势函数。在Actor-Critic架构下,Actor即为最终决策的策略网络,Critic网络的作用即是对于当前策略进行优势函数的评估。PPO算法[13]在此基础上将优化目标变为下式代理目标函数,从而使得模型在训练时更稳定和高效。

$$L(\theta)=E[\min(r_t(\theta)A_t,clip(r_t(\theta),1-\epsilon,1+\epsilon)A_t)] \tag{6}$$

其中,A_t即为优势函数,ϵ为超参数,$r_t(\theta)$为重要性采样相关的修正参数,表达式如下:

$$r_t(\theta)=\frac{\pi_\theta(a_t,s_t)}{\pi_{\theta_{\text{old}}}(a_t,s_t)} \tag{7}$$

最终,算法在工作时,需要针对Actor和Critic设置两套网络,Actor网络为智能体进行决策获得交互历史,Critic根据交互累积回报与预测结果之间的MSE建立损失函数进行参数更新,而Actor则依据上式对代理目标函数进行更新即可。

2 模型与算法

本章将介绍基于强化学习的SDN路由控制算法。通过在SDN中运行该算法,可以自动优化网络延迟、抖动、吞吐率等性能参数,实现对网络的实时控制,从而有效减轻网络运维压力。

2.1 任务建模

使用强化学习来优化SDN路由,其中最重要的是确定决策智能体所获取到的环境状态(state),奖励(reward)以及决策动作(action)空间。

强化学习作为一种动态优化算法,智能体在进行决策时并不需要感知到整个过程中不会发生改变的量,例如在路由优化时,具体的物理链接以及相应的网络拓扑已经给定,SDN控制器只需要根据当前网络流量信息来不断规划路由方案。因此,对于特定的SDN网络,智能体输入的状态s可以用当前网络负载的流量矩阵变换后的向量来进行表示,而网络本身的物理属性,在智能体的训练过程中就已经得到了表示学习。

智能体在感知到流量情况后,需要给出动作a,该动作能够为当前网络环境确定唯一的最优方案。在这里将动作设定一个表征所有链路权重的向量通过该链路权重向量,使用Floyd算法能够为网络唯一确定一套最优路由方案。

智能体获得的奖励与网络整体的性能指标相关,例如网络延迟、吞吐量,或者是考虑各个参数的综合奖励,例如式(8)即为综合考虑了多个性能指标的奖励函数。

$$R_{i\to j}=R(i\to j|s_t,a_t)=-h(a_t)-\alpha delay_{ij}+\beta BW_{ij}+\gamma loss_{ij} \tag{8}$$

其中,s_t表示网络当前状态,a_t为控制智能体产生的动作,通过该动作调整网络链路权重,并根据该链路权重重新计算网络中点对点的最优路径以获得唯一最优路由策略。假设链路权重调整后计算出的某条最优路径为p_{ij}。在等式中,函数h表示调整该路径的成本,例如对开关操作的动作影响。$\alpha,\beta,\gamma\in[0,1)$是可调权重,$delay$表示该条路径下的延迟时间,$BW$为带宽,$loss$为丢包率,奖励函数具体可由运行维护策略进行灵活确定。

2.2 算法框架

如图3所示,为本文提出的算法框架。PPO智能体通过以下三种变量与环境进行交互:状态、动作和奖励。

其中,状态s是当前网络负载的流量矩阵(TM),智能体对环境采取的动作为更改网络中链路的权重向量,通过该权重向量可以唯一确定一种网络中的路由方案,整体模型的训练过程伪代码如下算法所示:

1. 初始化PPO决策函数$\pi_\theta(s)$和价值函数$V_\omega(s)$
2. While Not Done:
3. $k=0$
4. $\theta_{old} \leftarrow \theta$
5. 随机初始化环境s_k以及缓存列表$buffer$
6. While $k<K$ do:
7. 获取动作$a\sim\pi_\theta(s_k)$
8. $s_{k+1},r_k = Env(s_k,a_k)$
9. 存储$[s_k,a_k,r_k]$于$buffer$
10. $k \leftarrow k+1$
11. End While
12. 根据$buffer$中的数据计算A_k与$Q_k(s_k,a_k)$
13. 依据式(6)计算$L(\theta)$
14. $J(\omega)=\frac{1}{K}(Q_k - V_w(s_k))^2$
15. $\theta \leftarrow \theta + \alpha_\theta \nabla L(\theta)$
16. $\omega \leftarrow \omega - \alpha_\omega \nabla J(\omega)$
17. End While

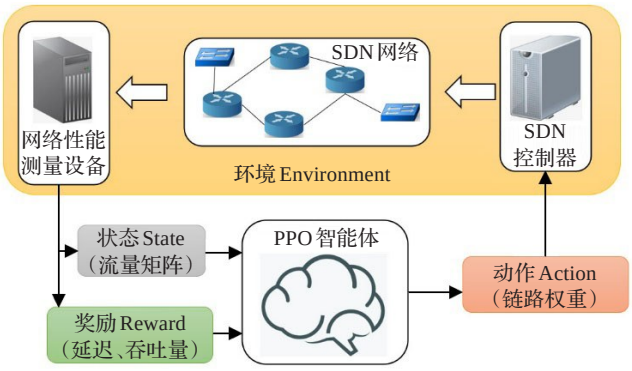

图3　整体模型架构

以上算法中,第6行至第11行为采样过程,第12行至第16行为模型参数的训练过程。其中第8行的Env将除PPO决策主体之外的环境进行了封装。另外决策函数$\pi_\theta(s)$通常表示的是一个高维正态分布,而动作a从该分布中进行采样得到。

PPO智能体的训练目的是根据输入状态s找到最佳动作a,以最大化累计回报。整体的工作过程可总结如下:通过SDN控制器的网络分析和测量,PPO智能体可以获取准确的网络状态s并确定最佳操作,即给出一组链路权重$[w_1,w_2,\cdots,w_n]$。根据一组更新的权重重新计算新的流路径,具体路径生成可以根据相应权重按Floyd最短路径算法来进行求解,从而SDN控制器生成新规则以建立新路径。在路径更新之后,通过下一次网络分析和测量获得奖励r和新的网络状态,从而迭代的优化网络的性能。

传统基于机器学习的路由优化算法从特定配置的网络数据中学习路由方案,因此只能在相应配置下工作,当网络硬件设备发生变化时,路由方案便失效。而本文使用的PPO算法作为在线学习方法,能够在与网络环境的交互中,不断使用近期的经验片段对算法智能体进行梯度更新,从而具有较强的适应性。即当网络物理设备发生局部变动时,PPO智能体也能够对此进行合理决策,以获得相应优化目标下的最优路由方案。除此之外,PPO算法作为强化学习中的策略方法,相较于各类基于Q-Learning的值方法路由算法,能够直接输出动作,从而更精细地对网络路由进行控制。且这种控制体方法将输出的动作向量与路由性能建立直接的映射,也更加便于PPO智能体的神经网络参数进行高效训练。

3　实验

3.1　实验环境

实验采用的计算机硬件配置为NVIDIAGeforce1080Ti GPU,32 GB内存,CPU为i9-9900k,操作系统选用Ubuntu16.04。使用tensorflow作为搭建算法的代码框架,使用OMNeT++[15]作为网络仿真的软件。对比了本文提出的基于PPO的路由算法和传统的基于最短路径主流路由算法以及随机生成的路由策略之间的路由性能差异。

实验选用了Sprint结构网络[16],该网络包含25个节点和53条链路,每条链路的带宽设为相同的值。实验针对该网络结构设置了几种不同级别的流量负载(Traffic Load,TL)来模拟真实的网络场景,每种不同的TL级别都为特定网络总容量的百分比,针对同一种级别的流量负载,使用引力模型(gravity model[17])来生成多种不同的流量矩阵。

为了验证本算法的有效性,在不同级别的流量负载上都对PPO决策智能体进行了训练以及测试。

3.2　算法的收敛性和有效性

实验首先针对PPO算法在不同级别的流量负载上的训练进行了实验,使用了四种流量负载:分别占整个网络带宽的10%、40%、70%和100%。每种流量负载下随机生成250个流量矩阵,其中200个作为训练环境,50个作为测试环境。训练模型时,针对每个级别的流量负载,每训练1 000步便测试模型在测试环境中的平均性能。给定流量矩阵和路由方案,使用OMNeT++来获取到网络延迟等性能参数,最终PPO模型在训练时针对不同的流量负载下的网络延迟测试结果如图4所示。

从图4可以看出,随着训练的不断进行,模型给出的路由方案能使得网络延迟不断降低,最终收敛到最优值。

接下来为了验证该算法的有效性,对比了PPO算法以及随机生成的50 000种路由方案在上述四种不同级别的流量负载下的延迟性能。PPO模型在这几种流量负

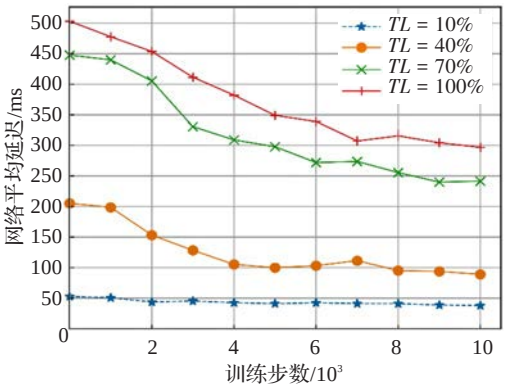

图4　PPO训练过程

载下的训练环境中训练至性能收敛后,在测试环境中进行多次测试。而这50 000种随机路由方案能够为PPO性能数据提供有代表性的对比数据。最终通过这些方案进行仿真获得网络延迟性能数据,画出箱形图如图5所示。

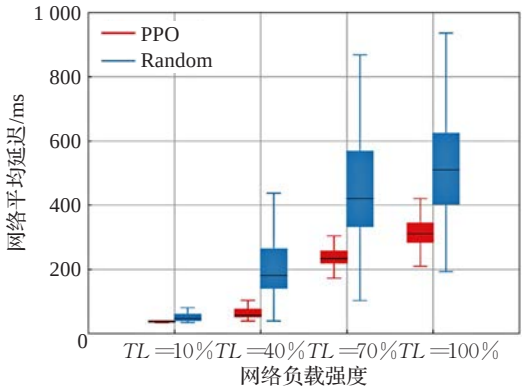

图5　随机路由策略与PPO优化算法的网络延迟

在图中,矩形的上部和底部分别表示网络延迟的上四分位数和下四分位数,矩形中间的线表示延迟的中位数。从矩形延伸的直线的上端和下端分别表示延迟的最大值和最小值。实验结果表明,本文提出的基于PPO的优化算法的最小延迟非常接近随机生成的路由配置的最佳结果,并且方差非常小,充分证明了PPO算法在SDN优化上的有效性。

3.3　模型性能对比

实验对比了PPO算法与文献[7]中提出的QAR路由算法在优化网络平均延迟和最大延迟上的性能差异,同时给出了传统基于最短路径的路由算法性能以作为参考。文献[7]中提出的QAR算法基于强化学习中的Q-learning算法建立模型,通过对网络状态建立状态-动作(state-action)值函数来对路由动作进行决策,是近年来真正将强化学习应用于SDN网络路由中的代表方法。

具体实验过程中,在不同负载强度级别上针对PPO模型与QAR模型进行收敛性训练后,使用了1 000个同级别的流量矩阵作为网络输入来测试这几种不同模型的网络平均延迟和网络端到端最大延迟,取平均值后的结果如图6和图7所示。

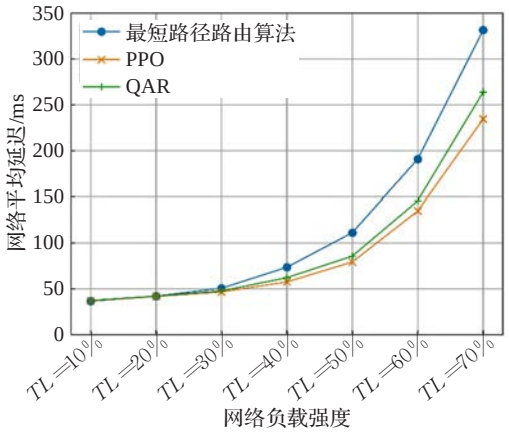

图6　网络平均延迟

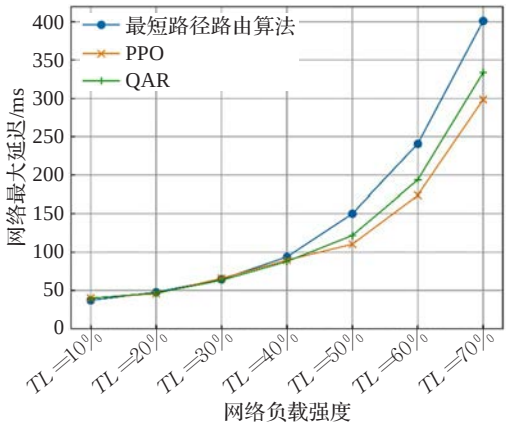

图7　端到端最大延迟

为了验证模型针对不同网络优化指标的广泛适用性,也针对网络吞吐率进行了实验。实验配置与上述基本一致,唯一差别仅在于奖励函数与吞吐率成正相关关系。实验结果如图8所示。

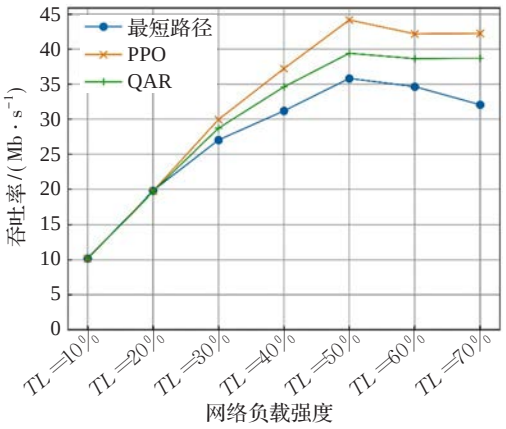

图8　网络吞吐率

实验结果表明,在负载强度较低时,这三种算法性能差异并不大。但当流量强度较高时,传统基于最短路径的路由算法会产生流量拥堵等问题,而QAR算法与基于PPO的路由优化算法可以有效地避免此类问题,从实验数据也可以发现,随着流量强度不断增大,这两类基于强化学习的模型给出的路由方案相比对传统算法性能提高的幅度越来越大。特别是在网络吞吐率的优

化问题上,当流量强度超过50%后,静态最短路径路由算法的吞吐率迅速降低,而基于强化学习的两种路由方法则在一定程度上规避了此问题。因此,通过在SDN路由管理中引入强化学习等机器学习方法确实能够提高路由效率。

对比QAR算法与基于PPO的路由算法,可以明显发现PPO的性能整体要优于QAR。分析原因如下,QAR算法在训练时,需要不断优化Q表,而复杂网络的状态空间十分庞大,仅仅依靠Q表难以处理。同时QAR在输出动作时,由于值方法的局限性,在具体控制路由时仅依靠离散动作对SDN控制器产生单一动作,无法进行精细控制。而基于策略方法的PPO算法一方面采用神经网络来对网络状态值进行拟合,同时直接输出与网络路由相关的动作向量以完成更加精细的控制,从而具有更好的优化性能。

总之,对比最短路径路由算法,基于PPO的路由算法在 $TL = 70\%$ 时,网络平均延迟降低了29.3%,端到端最大延迟降低了17.4%,吞吐率增加了31.77%。以上实验结果充分说明了本文提出的基于PPO的SDN路由控制算法不仅具有良好的收敛性,具有更好的性能和稳定性。

4 结论与展望

本文在知识平面网络的基础上提出了一种基于强化学习的SDN路由优化算法,该算法使用PPO深度强化学习机制来优化SDN网络的路由,从而实现了实时对SDN网络进行智能控制和管理。实验结果表明,本文提出的路由优化算法具有良好的收敛性和有效性,与现有的路由解决方案相比,该优化算法可以通过稳定、优质的路由服务来提高网络性能。

本文提出的算法能够对网络路由优化问题进行相对有效的处理。但在实践生产中,对于大规模复杂网络,获取大量训练样本也存在一定难度。PPO算法作为on-policy算法,尽管模型在训练时结合了重要性采样,但整体来说训练时的样本利用率还是较低。因此,如果能结合一些基于模型的强化学习算法,对于SDN路由优化的问题一定能有更高效的解决方案。

参考文献:

[1] BOUTABA R,SALAHUDDIN M A,LIMAM N,et al.A comprehensive survey on machine learning for networking:evolution,applications and research opportunities[J].Journal of Internet Services and Applications,2018,9(1):1-99.

[2] FADLULLAH Z M,TANG F,MAO B,et al.State-of-the-art deep learning:evolving machine intelligence toward tomorrow's intelligent network traffic control systems[J].IEEE Communications Surveys & Tutorials,2017,19(4):2432-2455.

[3] LI Wei,LI Guojun,YU Xiufen.A fast traffic classification method based on SDN network[M]//Electronics,communications and networks IV.London:CRC Press,2015.

[4] WANG Fu,LIU Bo,ZHANG Lijia,et al.Dynamic routing and spectrum assignment based on multilayer virtual topology and ant colony optimization in elastic software-defined optical networks[J].Optical Engineering,2017,56(7).

[5] PARSAEI M R,MOHAMMADI R,JAVIDAN R,et al.A new adaptive traffic engineering method for telesurgery using ACO algorithm over software defined networks[J].European Research in Telemedicine,2017,6(3/4):173-180.

[6] WANG J,DE LAAT C,ZHAO Z,et al.QoS-aware virtual SDN network planning[C]//IEEE Symposium on Integrated Network and Service Management,2017:644-647.

[7] LIN S,AKYILDIZ I F,WANG P,et al.QoS-aware adaptive routing in multi-layer hierarchical software defined networks:areinforcement learning approach[C]//IEEE International Conference on Services Computing,2016:25-33.

[8] JIANG J,HU L,HAO P,et al.Q-FDBA:improving QoE fairness for video streaming[J].Multimedia Tools and Applications,2018,77(9):10787-10806.

[9] SUTTON R S,BARTO A G.Reinforcement learning:an introduction[J].IEEE Transactions on Neural Networks,1998,9(5):1054.

[10] LIU C,JU W,ZHANG G,et al.A SDN-based active measurement method to traffic QoS sensing for smart network access[J].Wireless Networks,2020(2).

[11] XIONG F,LI A,WANG H,et al.An SDN-MQTT based communication system for battlefield UAV swarms[J].IEEE Communications Magazine,2019,57(8):41-47.

[12] MESTRES A,RODRIGUEZNATAL A,CARNER J,et al.Knowledge-defined networking[C]//ACM Special Interest Group on Data Communication,2017,47(3):2-10.

[13] SCHULMAN J,WOLSKI F,DHARIWAL P,et al.Proximal policy optimization algorithms.[J].arXiv:1707.06347,2017.

[14] SCHULMAN J,LEVINE S,ABBEEL P,et al.Trust region policy pptimization[C]//International Conference on Machine Learning,2015:1889-1897.

[15] ANDRÁS V,HORNIG R.An overview of the OMNeT++ simulation environment[C]//Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications,Networks and Systems & Workshops,SimuTools 2008,Marseille,France,March 3-7,2008.

[16] Sprint.Sprint IP network performance[EB/OL].[2011].https://www.sprint.net/tools/ip-network-performance.

[17] ROUGHAN M.Simplifying the synthesis of internet traffic matrices[J].ACM SIGCOMM Computer Communication Review,2005,35(3):93.