孙中祥, 高刃. 基于 SDN 的强化学习路由算法研究[J]. 智能计算机与应用,2025,15(6):146-150. DOI:10.20169/j.issn.2095-2163.250622

基于 SDN 的强化学习路由算法研究

孙中祥, 高 刃
(湖北汽车工业学院 电气与信息工程学院, 湖北 十堰 442002)

摘 要: 传统的路由算法使用有限的信息来做出路由决策,不能满足用户的不同 QoS 需求,针对以上问题提出了一种基于 SDN 的强化学习路由算法。使用强化学习常用的 Q-Learning 算法,首先对其 Q 表进行初始化,初始化的内容与目的主机的链路权重有关,有效降低了端到端的时延。然后对奖励函数做了赋权重值改进,能够更好地满足不同的 QoS 需求。在 Mininet 平台上搭建 SDN 拓扑使用 Ryu 控制器进行试验分析。相较于最短路径、ECMP 和 NSGA2 算法在延迟抖动、平均时延和丢包率上拥有更高的 QoS 性能,能够更好地适应差异化网络。

关键词: SDN; 强化学习; 服务质量(QoS); 路由算法; 时延抖动
中图分类号: TP393 **文献标志码**: A **文章编号**: 2095-2163(2025)06-0146-05

Research on reinforcement learning routing algorithm based on SDN

SUN Zhongxiang, GAO Ren
(School of Electrical and Information Engineering, Hubei University of Automotive Technology, Shiyan 442002, Hubei, China)

Abstract: Traditional routing algorithms use limited information to make routing decisions, which cannot meet the different QoS needs of users. A reinforcement learning routing algorithm based on SDN is proposed to address the above issues. Using the commonly used Q-Learning algorithm for reinforcement learning, the Q-table is firstly initialized, and the initialization content is related to the link weight of the destination host, effectively reducing end-to-end latency. Then, the reward function is improved by assigning weights and values, which can better meet different QoS needs. Finally, an SDN topology on the Mininet platform is built and experimental analysis is conducted using Ryu controllers. Compared to mainstream algorithms such as shortest path, ECMP and NSGA2, the proposed algorithm has higher QoS performance in latency jitter, average latency, and packet loss rate, and can better adapt to differentiated networks.

Key words: SDN; reinforcement learning; Quality of Service(QoS); routing algorithm; delay jitter

0 引 言

软件定义网络(SDN)是一种新型的网络架构,通过软件编程的形式定义和控制网络,从而极大地简化了网络的管理,促进了网络创新和发展[1]。近年来,随着人们对服务质量要求的提高,强化学习(RL)[2]被应用到 SDN 中用来优化路由,为智能路由算法的部署提供了一种新思路[3]。

在传统的路由方法中,每个路由器都会根据自己的数据包制定转发决策,而不会考虑其他路由器的决策。尽管这种分布式路由的方法提供了可扩展性,但面对当前视频音频实时传输丢包率大等问题,难以实现路由优化。为了解决这些问题,学界提出了软件定义网络(SDN)作为管理整个网络的有效手段,通过将网络中的控制平面和数据平面分开,将数据传输与控制操作区分开来[4-5]。SDN 提供整个网络的全局视图,并通过逻辑解耦网络中的控制平面和数据平面来提高网络可编程性[6]。

Q-Learning 算法作为常用的强化学习算法,有着不需要环境模型,可以直接从经验中学习、广泛适用于离散和连续的状态空间,以及离散和连续的动作空间等优点。近年来被广泛应用于路径优化:在文献[7]中引入了探索因子加快了前期的收敛速度,提升了路由的效率避免了路径拥塞。在文献

基金项目: 湖北省教育厅重点科研项目(D20211802); 湖北省科技厅重点研发计划项目(2022BEC008)。
作者简介: 孙中祥(1999—),男,硕士研究生,主要研究方向:网络通信,软件定义网络。
通信作者: 高 刃(1979—),男,博士,教授,主要研究方向:智慧供应链、软件定义网络。Email:gaoren@huat.edu.cn。
收稿日期: 2023-10-17

[8]中改进了多目标路由的方法，有效地提高了无线通信的时延、带宽和丢包率。在文献[9]中引进了拓扑收敛的技术来规划备用路径，确保在周期窗口内备用路径的性能。子拓扑网络是指一个网络的一部分，可以独立于整个网络进行规划和管理。这种方法可以提高备用路径的性能，并确保网络在故障时能够快速恢复。在文献[10]中提出了一种负载均衡的方法，能够提高网络性能和管理效率，可以根据网络环境自动做出决策，避免网络拥塞，实现网络资源的合理分配。

根据以上文献对 Q-Learning 算法的改进，缺少对 Q 表初始化和对奖励函数的改进方法。针对以上问题，本文提出了一种基于 SDN 的强化学习路由算法(QI-RL)，主要对初始化 Q 表的方法做了改进，使算法能够快速地收敛，减少了前期不必要的盲目探索，节省了一定的时延。其次，对奖励的函数做了一定的改进，对于不同的 QoS 需求可以设置不同的权重。

1　理论基础

1.1　SDN 框架

SDN 采用了 2 个相互分离的平面：集中式的控制平面和分布式的转发平面。其中，控制平面利用控制-转发通信接口对转发平面上的网络设备进行集中控制，并提供灵活的可编程能力。SDN 的基础框架如图 1 所示。

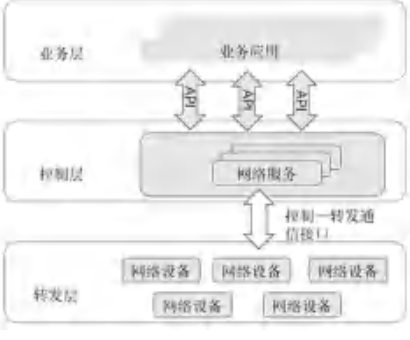


图 1　SDN 框架图

Fig. 1　Diagram of SDN framework

在 SDN 的架构中，控制平面通过控制-转发通信接口对网络设备进行集中控制，这部分控制信令的流量发生在控制器与网络设备之间，独立于终端间通信产生的数据流量，网络设备通过接收控制信令生成转发表，并据此决定数据流量的处理，不再需要使用复杂的分布式网络协议来进行数据转发。处理流程表如图 2 所示。

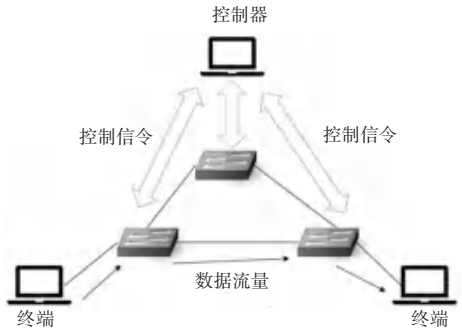


图 2　处理流程表

Fig. 2　Flow chart of the process

1.2　传统的 Q-Learning 算法

强化学习是一种不含标签的机器学习算法，通常包含智能体(Agent)、环境(Environment)、状态(State)、动作(Action)以及奖励(Reward)等基本元素，核心思想是智能体在环境中不断地尝试，通过环境反馈的奖励来对自身策略进行持续优化以实现最佳决策[11]。

Q-Learning 是强化学习中一种具有代表性的算法[12]。Q-Learning 算法的核心是贝尔曼等式(Bellman equation)，描述了当前状态和行动的 Q 值与下一状态的 Q 值之间的关系。贝尔曼等式具体如下：

$$Q(s,a) = r + \gamma \max\_a' \times Q(s',\ a') \qquad (1)$$

其中，s 表示当前状态；a 表示当前的行动；r 表示状态 s 采取行动 a 获得的即时奖励；γ 表示折扣因子，取值范围为[0,1]，用于平衡即时奖励和未来奖励；s' 表示采取行动后 a 将进入下一个状态；$\max\_a'$ 表示在下一个状态 s' 下，对所有可能行动 a' 的 Q 值取最大值。

Q-Learning 算法的步骤为：首先初始化 Q 值表，根据当前的状态 s，选择一个动作 a。通常通过 ε - 贪心策略实现，ε 是用在决策上的一种策略，比如 $\varepsilon = 0.8$ 时，就说明有 80%的情况会按照 Q 表的最优值选择行为，20%的时间使用随机选择行为。接下来，执行行动 a，观察即时奖励和下一个状态 s'，使用贝尔曼等式更新 Q 值，更新的等式如下所示：

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max\_a' \times Q(s',a') - Q(s,a)] \qquad (2)$$

其中，α 表示学习率，当 Q 值更新后更新为下一状态 $s = s'$。通过反复执行这些步骤，智能体学习到一个能够知道其在各种状态下选择最优行动的 Q 值表。Q-Learning 算法流程如图 3 所示。

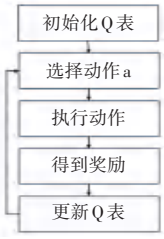


图 3 Q-Learning 算法流程图

Fig. 3 Flowchart of Q-Learning algorithm

1.3 QoS 指标

本文基于优化路由算法,主要对 QoS 的时延抖动、平均时延和丢包率三项指标进行对比。

(1)传送时延。是衡量数据包穿越网络所用的时间指标,指从网络的一端发送到另一端所需要的延迟时间。随着网络不断地更新迭代,对时延的要求也越来越高。

(2)抖动。是衡量一个网络延迟稳定性的指标,在数值上等于延迟变化量的绝对值。抖动产生的原因是延迟随机性,在网络环境中由于分组转发的原因,同一数据流中的 2 个包通过不同的路径到达对端,时延相差较大。在相同的路径中,网络设备和链路资源的情况也是不断变化的,就会造成到达对端的时延的不同,从而产生不同的时延抖动。通常情况下,延迟越小就意味着抖动的幅度和范围越小。

(3)丢包率。本质上是指在网络传输的过程中丢失报文的数量占传输报文总数的百分比。少量的丢包对业务的影响并不大,例如,在语音的传输中丢失一个比特或者分组,通话的双方往往注意不到。但是在视频的传输中,丢失一个比特或一个分组却可能对视频的波形造成很大的影响。

实验发送的是 UDP 数据包,不提供数据的可靠传输。没有重传机制,当网络拥塞时会导致丢包增加。究其原因,主要来自于网络拥塞和流量限制。

2 算法设计

2.1 改进 Q-Learning 的方法

本文主要对 Q-Learning 算法主要做了 2 个改进,使算法以更少的训练次数得到最短路径,从而减少了路由的时延。对 Q-Learning 算法的改进主要从 2 个角度考虑:首先是对 Q 表格进行初始化并引入了修改 Q 表初值的方法。然后对奖励函数做了一定的改进,使路由算法更好地适用于各种 QoS 需求。

2.1.1 Q 表初始化

对于传统的 Q-Learning 算法,核心思想是通过估计旧的 Q 值和新的 Q 值进行权重平均值的迭代更新,以逐步优化智能体在给定状态下采取给定行动的预期效用。相比于初始 Q 表是空表的状态,将当前主机的位置到目标主机的最近下一跳交换机距离作为 Q 表的初始值,在前期的训练过程中就更容易找到最短路径。对每 2 台主机之间的链路赋予一个权重值,用于计算链路间的距离。相比于传统算法,这种做法可以避免在训练的初期出现过多的随机性,从而加速收敛。如果 Q 表中已有赋值,那么智能体在训练过程中会尝试所有可能的动作,从而避免陷入局部最优解。因此,通过赋初值的方法有效提高了算法的收敛速度。

2.1.2 奖励函数的改进

本文利用 QoS 指标来评估网络的状态。评价算法的指标为常见并重要的时延 d_{ij}、带宽 b_{ij} 和丢包率 l_{ij}。奖励函数可定义为:

$$R_t = \beta d_{ij} + \theta b_{ij} + \varphi l_{ij} \tag{3}$$

其中,β、θ、φ 表示 QoS 的指标的权重值。随着当前人们生产生活水平的不断提高,在不同的时间对 QoS 有着差异的需求,不同的 QoS 需求可以通过设置不同的权重来实现。

2.2 算法流程图

基于前文论述的 Q-Learning 算法,改进后得到的 QI-RL 路由算法的伪代码描述如下所示。

输入 源地址 s, 目的地址 d

输出 路由路径

1. 初始化:初始状态为 s,训练次数为 0,获取网络拓扑信息和带宽、时延、丢包率等网络 QoS 信息,初始化 Q 表

2. while 训练次数小于等于训练总次数

3. while 当前状态下 $s_i \neq d$

4. 形成由当前状态 s_i 下所有可选动作组成的动作集 $A(s_i)$

5. 基于 ε -贪心算法选择动作 a_i 并执行,当前状态 s_i 转移到新状态 s_i'

6. 根据式(1)更新 $Q(s_i, a_i)$

7. end while

8. 训练次数加 1

9. end while

10. 根据训练后的 Q 表获取最终的路径

3 仿真和结果分析

3.1 环境配置

本实验所用的处理器为 INTER CoreI3 -10105F;机带内存 8 GB;实验的平台搭建在 Ununtu20.04 操作

系统下；所使用的仿真平台为 Mininet，用于搭建网络平台的不同拓扑；控制器使用的是轻量级、高效率的 Ryu 控制器，主要用于控制流表的下发等一系列的操作。实验所使用的拓扑为经典的 NSFNet 拓扑[13]，每条链路上都自定义了链路效率，在路由决策的过程中起到了关键的作用。实验拓扑如图 4 所示。算法的参数设置见表 1。

表 1　Q 算法中的参数

Table 1　Parameters in Q algorithm

参数	数值
学习率 α	0.6
折扣因子 γ	0.9
训练次数	100
探索率 ε	0.1
β	0.7
θ	0.2
φ	0.1

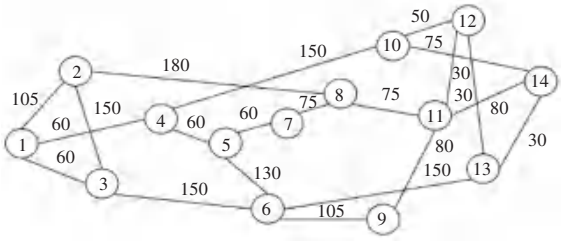


图 4　实验拓扑图

Fig. 4　Diagram of experimental topology

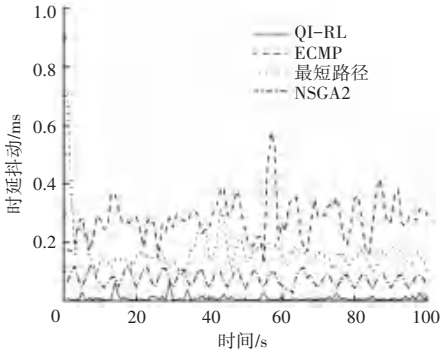
3.2　仿真结果分析

为了验证 QI-RL 算法的有效性，分别以 50 M、100 M、200 M 的带宽，在链路上持续发送数据。主要与最短路径算法[14]、等价多路径算法(ECMP)[15]、NSGA2 算法的时延抖动、平均时延和丢包率进行对比。

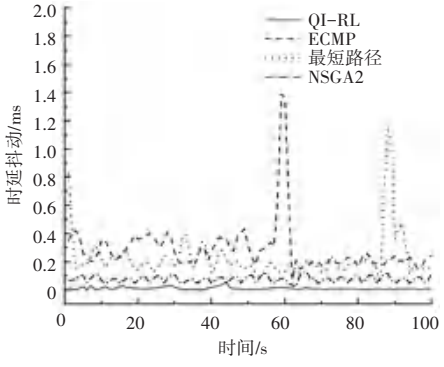
50 M 带宽的时延抖动如图 5(a)所示。分析可知，最短路径算法需要存储大量的中间结果，当处理较复杂的路由时导致内存消耗较大。在带宽比较小的时候时延的抖动非常大，无法满足不同的 QoS 需求。ECMP 路由算法由于没有拥塞控制的机制，只是简单地将流量分散到不同的路径上转发。对于已经产生拥塞的路径来说，很可能会加剧路径的拥塞，因此算法的时延抖动比较大。NSGA2 算法由于引进了精英策略，提高了算法的运算速度和鲁棒性，时延在比较固定且小的范围内抖动，但是相比于 QI-RL 算法在 QoS 上略差。

100 M 带宽的时延抖动如图 5(b)所示。随着带宽的增加，对最短路径算法和 ECMP 算法的影响不大，但是 NSGA2 算法的时延抖动得到了进一步压缩，而且 QI-RL 算法的时延抖动已经控制到很小。

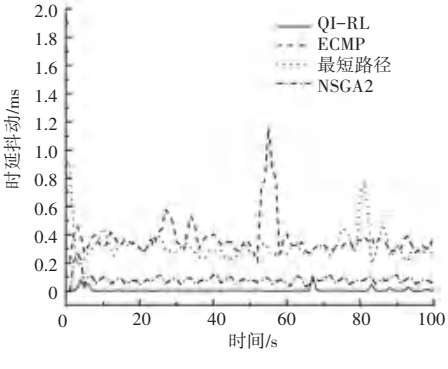
200 M 带宽的时延抖动如图 5(c)所示。随着带宽增加到接近真实环境，最短路径算法和 ECMP 算法有了一定的改善、但效果并不明显。NSGA2 算法的时延抖动基本不变，此时 QI-RL 算法的时延抖动可以较好地满足不同的网络服务。



(a) 50 M 带宽的时延抖动



(b) 100 M 带宽的时延抖动



(c) 200 M 带宽的时延抖动

图 5　不同带宽的时延抖动对比

Fig. 5　Comparison of delay jitter with different bandwidths

不同带宽平均时延对比如图 6 所示。这 4 种算法的平均时延有一定的区别。最短路径算法和 ECMP 算法的抖动比较明显，这说明网络中的流量并不稳定，相应的平均时延也比较高。NSGA2 算法的抖动控制的范围比较小，不同的带宽对时延的影响较小，相对稳定。QI-RL 算法的抖动最低，并且随着带宽的增加，平均时延能够继续得到改善，可以

更好地完成传输任务。

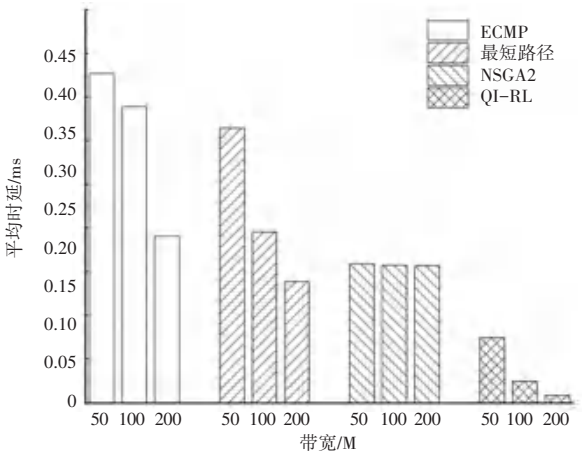


图 6 不同带宽平均时延对比图

Fig. 6 Comparison chart of average latency across different bandwidths

不同带宽丢包率如图 7 所示。随着带宽的增加,4 种算法的丢包率都有着不同程度的下降。最短路径算法和 ECMP 算法的丢包率较高,无法满足正常的 QoS 需求。NSGA2 算法在带宽较小的时候丢包率很大,但随着带宽增加到 200 M 丢包率急速下降,可以满足一些特定的需求。QI-RL 算法对奖励函数做出改进,当需要链路对丢包率要求较高时可以增加其权重,使丢包率一直处在较低的水平,可以满足用户的不同 QoS 需求。

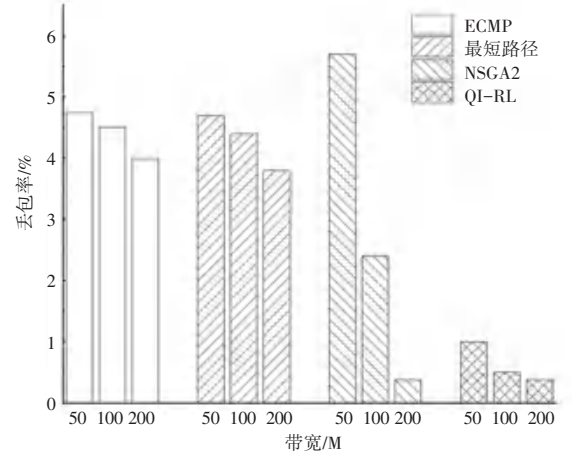


图 7 不同带宽丢包率图

Fig. 7 Graph of packet loss rates for different bandwidths

4 结束语

本文是在 SDN 网络架构的基础上,基于 Mininet 平台上搭建的拓扑,使用 Ryu 控制器易管理和发现网络拓扑并可以实施网络流量的控制等优点,提出了一种基于 Q-Learning 算法的路由改进算法,算法首先根据链路初始化 Q 表,能够更快速地找到路径,减少了前期盲目探索的时延,使延迟的抖动得到了有效的降低。对奖励机制做出的改进,可以更好地完成各种网络的 QoS 指标。

实验的结果表明,与传统的最短路径算法和 ECMP 算法相比,时延的抖动和丢包率问题在不同的带宽环境下都有着较大的优势。相比于 NSGA2 算法,在低带宽的情况下优势进一步扩大。结果显示,本文所提的算法与传统算法相比,能够更好地适应网络 QoS。

对于以后的工作,可以继续对动作选择策略进行优化,以及将深度强化学习运用到不同的子网络拓扑中,使路由算法更快地收敛,进一步提升算法的性能。

参考文献

[1] KREUTZ D, FERNANDO M V R, VERíSSIMO P E, et al. Software-defined networking: A comprehensive survey [J]. Proceedings of the IEEE, 2015,103(1):14-76.

[2] KAELBLING L P, LITTMAN M L, MOORE A W. Reinforcement learning: A survey [J]. Journal of Artificial Intelligence Research, 1996,4:237-285.

[3] 刘辰屹,徐明伟,耿男,等. 基于机器学习的智能路由算法综述[J]. 计算机研究与发展,2020,57(4):671-687.

[4] Cisco Systems, Inc. Software-defined networking [EB/OL]. [2025-04-01]. https: www. cisco. com/c/en-au/solutions/software-defined-networking/overview. html.

[5] ORTIZ S. Software-defined networking: On the verge of a breakthrough? [J]. Computer,2013,46(7):10-12.

[6] KIM G, KIM Y, LIM H. deep reinforcement learning-based routing on software-defined networks[J]. IEEE Access, 2022, 10:18121-18133.

[7] 宋丽君,周紫瑜,李云龙,等. 改进 Q-Learning 的路径规划算法研究[J]. 小型微型计算机系统,2024,45(4):823-829.

[8] 于佳禾,胡春燕,周园. 基于 Q 学习的无线通信网多目标智能路由策略[J]. 计算机仿真,2024,41(3):431-435.

[9] 李传煌,陈泱婷,唐晶晶,等. QL-STCT:一种 SDN 链路故障智能路由收敛方法[J]. 通信学报,2022,43(2):131-142.

[10] 王炜发,张大明,刘堃钤,等. 软件定义网络中基于 Q-学习的负载均衡算法[J]. 电讯技术,2021,61(9):1066-1072.

[11] 陈学松,杨宜民. 强化学习研究综述[J]. 计算机应用研究,2010,27(8):2834-2838.

[12] LI Runxia, FU Li, WANG Lingling, et al. Improved Q-learning based route planning method for UAVs in unknown environment [C]//Proceedings of 2019 IEEE 15th International Conference on Control and Automation (ICCA). Piscataway, NJ: IEEE, 2019: 118-123.

[13] STRAWN G, STRAWN G. Masterminds of the NSFNet: Jennings, Wolff, and Van Houweling [J]. IT Professional Magazine,2021,23(6):67-69.

[14] MOY J. OSPF version 2[EB/OL]. (1998-04-01). https://www. rfc-editor. org/info/rfc2328.

[15] HOPPS C. Analysis of an equal-cost multi-path algorithm[EB/OL]. (2000-11-01). https://www. rfc-editor. org/info/rfc2992.