



基于深度学习的流量工程算法研究与应用

胡道允¹, 齐进¹, 陆钱春¹, 李锋¹, 房红强²

(1. 移动网络和移动多媒体技术国家重点实验室, 广东 深圳 518057;

2. 中国科学技术大学, 安徽 合肥 230026)

摘要: 随着 5G 网络的发展和应用, 网络中的业务数量呈现出爆发式增长, 网络中的带宽资源日趋紧张。为了提高网络资源利用率, 并满足用户日益提高的业务服务质量要求, 基于软件定义网络 (SDN) 提出了一种基于深度学习的流量工程算法 (DL-TEA)。通过仿真证明该算法不仅能够实时地为业务计算一条高效的路径, 同时还能够提升业务的 QoS、网络资源利用率, 降低网络阻塞率。

关键词: 软件定义网络; 流量工程; 深度学习; 服务质量

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2021027

Research and application of traffic engineering algorithm based on deep learning

HU Daoyun¹, QI Jin¹, LU Qianchun¹, LI Feng¹, FANG Hongqiang²

1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518057, China

2. University of Science and Technology of China, Hefei 230026, China

Abstract: With the development and application of 5G network, the amount of traffic in network increased rapidly, which caused the lack of bandwidth resource. In order to improve the utilization of network resource and satisfy the critical user requirement for QoS (quality of service), a novel traffic engineering algorithm based on deep learning in SDN was proposed. At last, simulation results show that the proposed algorithm not only can calculate an efficient path for service in real time, but also can improve the QoS and the utilization of network resource, as well as reduce network congestion.

Key words: SDN, traffic engineering, deep learning, QoS

1 引言

随着互联网技术的发展, 尤其随着 5G 网络的部署和应用, 网络中需要传输的数据量呈现出爆发式的增长, 网络资源日趋紧张。而用户对业务

QoS (quality of service, 服务质量) 的要求却在不断提高, 这给传统的网络管理系统带来了巨大的挑战。近年来, SDN (software defined networking, 软件定义网络) 的概念被广泛关注, 它通过解耦传统路由器中的控制和转发功能, 形成一个统一



的集中式控制平面^[1]。控制平面可以实时地获得网络的全局视野和流量信息,实现网络业务流量的精细化管理,从而提升网络资源利用率^[2-3]。然而,网络中业务流量的突发性和动态性特点越来越明显,传统的最短路由算法极有可能使网络发生网络拥塞或者带宽闲置,这会造成业务 QoS 下降或者网络资源浪费^[4]。传统的基于全局业务的网络资源优化算法,虽然能够提升网络资源利用率,但是无法满足网络中业务请求的动态性和实时性要求^[5-6]。

为解决上述问题,本文基于 SDN 提出了一种基于 DL (deep learning, 深度学习) 的流量工程算法 (DL-TEA)。该算法主要分为离线训练和在线决策两个阶段:在离线训练阶段,通过深度学习模型学习启发式算法的路径信息;在线决策阶段利用训练好的深度学习模型对网络拓扑进行剪枝,然后利用贪婪算法计算最优路径。因此,DL-TEA 的在线运行耗时和贪婪算法相近,满足了动态业务的实时路由需求。另外,通过深度学习模型保留了启发式算法的特征信息,使得 DL-TEA 的性能接近启发式算法,与贪婪算法相比,提升了请求 QoS、网络资源利用率,降低了网络阻塞率。

2 研究现状

针对动态业务的实时路由问题,参考文献[7]提出利用遗传算法更新网络中链路的权重,然后再利用最短路径算法计算业务路由,该方法虽然能够保证当前业务路径最优,但是无法保证网络全局资源的合理化分配。而针对批量业务的全局最优问题,已经被证明为多商品流问题^[5]。由于多商品流问题为 NP 难问题,所以参考文献[6]提出了一种 SAGA 算法,该算法结合遗传算法和模拟退火算法,通过局部搜索达到负载均衡的目的,但是该算法时间复杂度较高,并且是针对全局静态请求的优化,无法满足动态业务的实时性需求。

最近,人工智能技术受到大家的广泛关注,并且已有相关研究在讨论将人工智能技术引入网络管

理中^[8]。参考文献[9]探讨了将机器学习应用到网络的路由场景中。参考文献[10]提出为业务计算多条路径,并通过机器学习模型实现快速路由决策。参考文献[11]通过统计历史路径信息,然后利用深度强化学习算法从历史路径集合中选择合适的传输路径。虽然上述两种算法均能够解决动态业务的实时路由问题,但是业务的路径只能从固定的路径集合中选择。而本文提出的 DL-TEA 则是利用深度学习模型统计各链路在不同网络状态下被不同请求占用的概率信息,然后根据概率信息进行拓扑剪枝,因此业务传输路径不会受限于固定的路径集合。参考文献[12]提出先将网络流进行分类,然后基于深度强化学习实现 QoS 的路由算法。参考文献[13]指出深度强化学习在网络流问题中存在收敛慢、鲁棒性低的缺点,并提出通过将网络节点划分为驱动节点和从节点的方式压缩问题的维度,从而提高了模型的收敛时间,但是使用的网络规模仍控制在 100 个节点以内,而本文提出的 DL-TEA 在仿真实验中能够支持 1 000 个节点以上规模的网络拓扑运算。

3 算法设计

本文提出的 DL-TEA 算法主要包含两个部分——离线训练过程和在线决策过程,如图 1 所示。

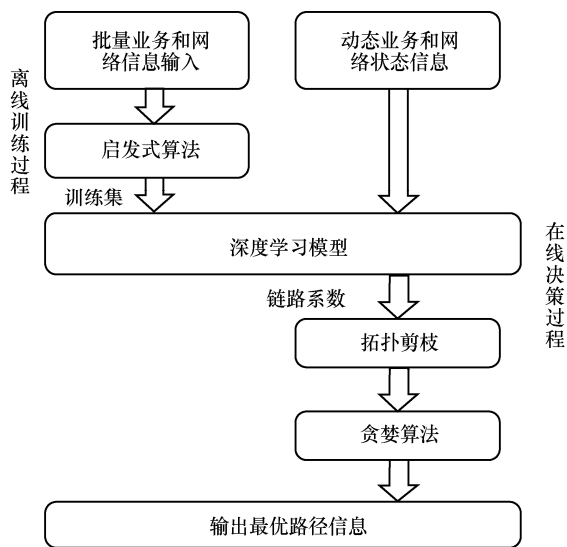


图 1 基于深度学习的动态路由决策过程

离线训练过程：首先统计或模拟网络中的历史请求信息，然后将批量业务请求和网络信息作为启发式算法的输入，再通过第 3.2 节中的启发式算法计算业务路由信息并生成训练集，最后对深度学习模型进行离线训练。

在线决策过程：当 SDN 控制器收到新的业务请求时，将请求信息和当前网络状态信息作为输入传递给训练好的深度学习模型，经过模型计算得到链路系数。然后根据链路系数大小，通过设置阈值对网络拓扑进行剪枝，保留系数较大的链路。最后根据剪枝后的拓扑，利用第 3.1 节中的贪婪算法进行路由计算并输出业务路由信息。另外，由于有限的训练集无法涵盖所有的网络状态和请求信息，所以模型存在一定概率导致拓扑剪枝过度，造成业务路由计算失败。为避免上述情况，当业务路径计算失败时，DL-TEA 会取全量的网络信息重新进行路由计算。

本文中，使用符号 $G(V, E, B, D)$ 表示网络状态信息，其中， $v \in V$ 表示网络节点， $e=(u, v) \in E$ 表示网络链路； b_e 表示链路 e 上可用的带宽。 D_e 表示链路 e 上的时延信息。另外，定义链路 $e \in E$ 上权重设置为 m_e 。定义 $q(s_q, d_q, b_q) \in Q$ 表示业务请求，其中， s 表示请求的源地址， d 表示请求的目的地址， b 表示请求的带宽需求。最终，需要为每个业务都计算一条满足带宽需求的路径 $P_i=(e_{i1}, e_{i2}, \dots, e_{ik})$ 。

3.1 贪婪算法

由于在实际网络中，用户会根据不同的业务场景设置不同的优化目标。因此，贪婪算法需要根据用户的需求设置不同的链路权重，然后利用 Dijkstra 算法计算权重最优路径。本文将针对占用带宽之和最小、请求时延之和最小和网络拥塞最小 3 个场景进行讨论。针对业务占用带宽之和最小的优化目标，如式 (1) 所示。

$$\text{Minimize } \sum_{q \in Q} b_q \cdot |P_q| \quad (1)$$

此场景中，只需保证每个业务的路径最短，

即可得到业务的最优路径。因此，定义链路权重如式 (2) 所示，针对请求 q ，如果请求带宽 b_q 小于链路剩余带宽 b_e ，链路权重 $m_e=1$ ，否则为无穷大。

$$m_e = \begin{cases} 1, b_e \geq b_q \\ \infty, b_e < b_q \end{cases}, \forall e \in E, \forall q \in Q \quad (2)$$

同理，针对请求时延之和最小的优化目标，将链路的时延设置为链路的权重，然后利用 Dijkstra 算法计算一条满足带宽需求的最短时延路径。

而针对网络拥塞最小的优化目标，链路 $e \in E$ 上权重设置为 m_e ，如式 (3) 所示，如果链路剩余带宽大于请求带宽，则链路权重等于链路总带宽除以链路剩余带宽，然后乘以负载均衡系数，否则设置为无穷大。因此，当链路上剩余的带宽资源较多时，则链路的权重 m_e 较小，在使用 Dijkstra 算法计算路由时，将更倾向经过该链路，从而保证网络的资源部署相对均衡，减少网络拥塞。

$$m_e = \begin{cases} \chi \cdot b_e^{\text{total}} / b_e, b_e \geq b_q \\ \infty, b_e < b_q \end{cases}, \forall e \in E, \forall q \in Q \quad (3)$$

3.2 启发式算法

针对批量业务全局优化的流量工程问题，业界已经提出了很多优秀的启发式算法^[5,7]。本文基于模拟退火算法设计了启发式算法，该算法通过调整业务的路由信息，不断搜索较优的路由结果，从而实现网络资源优化。

步骤 (1)、(2) 利用贪婪算法依次为请求求解并扣除带宽资源，产生初始解。步骤 (3)~(11) 为模拟退火过程，每次迭代会随机选择一批请求 R' ，在释放带宽资源后，随机调整请求顺序，然后利用贪婪算法重新计算和部署业务，产生新的结果 P' 。最后计算评价函数 $E(P')$ ，如果新解的结果优于之前的解，则接受该结果，否则按概率 γ 接受该组结果。值得注意的是，不同优化目标场景下的评价函数 $E(P')$ 是不同的。

算法 1 模拟退火算法

输入 网络拓扑 G ，请求 R (包含 N 个请求)



输出 请求的传输路径

- (1) 根据不同目标设置链路权重值
- (2) 利用贪婪算法依次计算出每个请求的最短路径, 并扣除带宽资源
- (3) while 评价函数没有收敛:
- (4) 按概率 λ 随机抽取部分业务 R'
- (5) 释放业务的路由和带宽信息
- (6) 乱序后重新利用贪婪算法计算业务路由集合 P'
- (7) if $E(P') < E(P)$;
- (8) 接受该解
- (9) else
- (10) 按概率 γ 接受新解
- (11) end if

3.3 深度学习模型

将请求信息和网络状态作为神经网络的输入, 以当前网络状态下业务经过链路的概率为输出, 如式 (4) 所示, 其中, s 表示请求的源站点信息, d 表示请求目的站点信息, b 表示请求带宽信息, B_i 表示链路 i 的带宽信息, D_i 表示链路 i 的时延信息, h_i 表示链路 i 被该请求占用的概率信息。

$$(h_1, h_2, \dots, h_{|E|}) = F(s, d, b, B_1, B_2, \dots, B_{|E|}, D_1, D_2, \dots, D_{|E|}) \quad (4)$$

本文中, 采用一个深度神经网络 (DNN), 如图 2 所示。另外, 将请求的源目的节点进行 one-hot 编码, 因此神经网络的输入维度扩展为 $|V|+1+2|E|$, 输出维度为 $|E|$ 。针对 DNN 的隐藏层, 采用 ReLU 函数作为激活函数, 并利用残差网络对其进行优化, 通过残差块能够有效地避免神经网络的退化问题。针对输出层采用了 sigmoid 函数作为激活函数, 并在输出层前添加 dropout 防止过拟合。输入的带宽以及时延不是 0 到 1 之间的数值, 在中间层加了几层批量正则化 (batch-normalization) 层使得经过该层的结果均值为 0、方差为 1, 从而使得模型训练得更快以及防止梯度爆炸和梯度消失的问题出现。

深度学习模型需要足够多的训练集才能保证模型的收敛。本文按泊松分布模型模拟生成动态业务请求, 定义请求平均到达率为 $\bar{\nu}$, 请求的平均生存时间为 $\bar{h}$, 则网络的负载量如式 (5) 所示。请求的源宿随机生成, 请求带宽处于 [1,3]。

$$\text{Erlangs} = \bar{\nu} \cdot \bar{h} \quad (5)$$

本文中模拟了不同负载量下不同时刻的请求

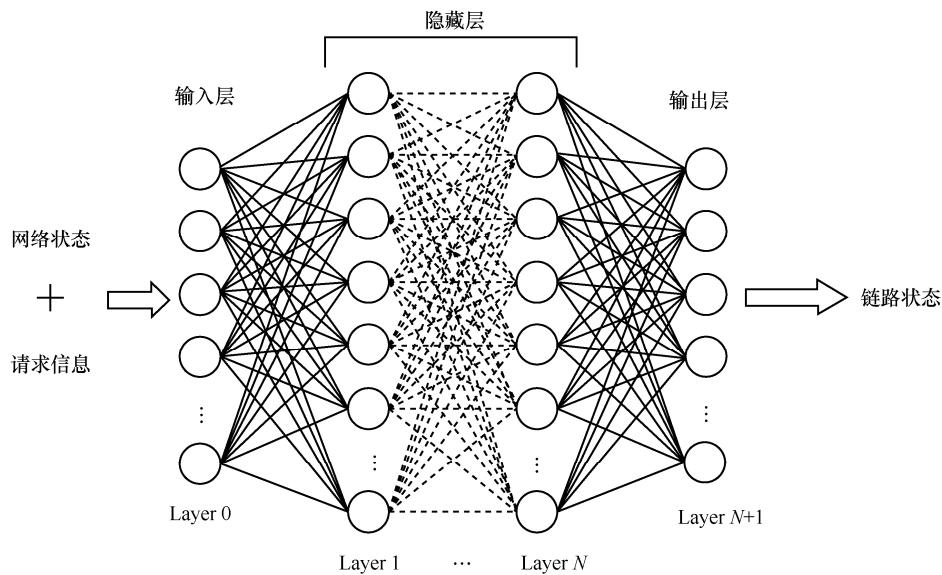


图2 深度神经网络模型

集合, 训练集生成流程如算法 2 所示, 针对某一时刻的请求集合 Q , 利用启发式算法进行网络优化, 得到请求对应的路径集合 P 。针对每组结果, 执行 K 次循环, 每次循环需要重置网络状态, 对请求进行随机排序, 然后依次部署请求。在部署过程中, 取当前网络状态以及当前请求信息和路径信息作为一个训练样本, 如步骤 (3) ~ (7) 所示。针对单个训练样本, 如果请求 q 的路径 p_q 经过链路 e , 则链路 e 对应的值为 1, 否则为 0。

算法 2 训练集生成过程

输入 网络拓扑 G , 批量请求集合 Q

输出 训练子集

- (1) 根据启发式算法进行求解, 得到路径集合 P
- (2) for $i=1$ to K ;
- (3) 重置网络状态, 对请求随机排序
- (4) for q in Q ;
- (5) 收集当前网络状态以及请求信息 q 和路径信息 p 作为训练样本
- (6) 按路径 p 部署该请求 q
- (7) end for
- (8) end for

4 仿真验证

在仿真验证阶段, 将对比贪婪算法、模拟退火算法和 DL-TEA 3 种算法在不同优化目标场景下的算法性能和算法耗时。为了保证仿真的准确性, 采用了国内某省份的回传网络作为仿真拓扑, 该拓扑共 1 161 个节点、2 760 条链路, 如图 3 所示。

另外, 本文按 Erlang=3 500 的负载规模, 模拟了 50 个时刻的请求分布情况, 每个时刻网络中平均 3 500 个请求。然后利用第 3 节中所述方法生成数据集。其中随机选择了 40 个时刻的数据作为训练集, 剩余 10 个时刻的数据作为测试集。

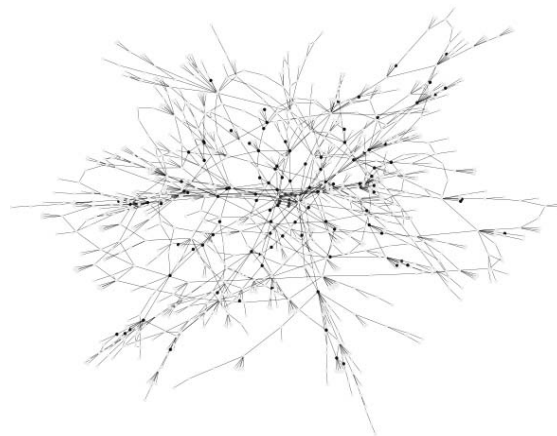


图3 仿真拓扑

用上述数据集对神经网络模型进行训练和测试。针对一个请求, 如果神经网络给出的相关链路输出值和训练集或测试集对应的输出一致, 则认为模型输出准确, 反之则认为输出错误。本文定义链路准确率等于正确的输出链路总数除以所有的链路输出总数。链路正确率统计结果如图 4 所示: 从结果中能够看出, 随着训练次数的增加, 模型的准确率也逐渐提升, 最终收敛在 99.5% 以上。

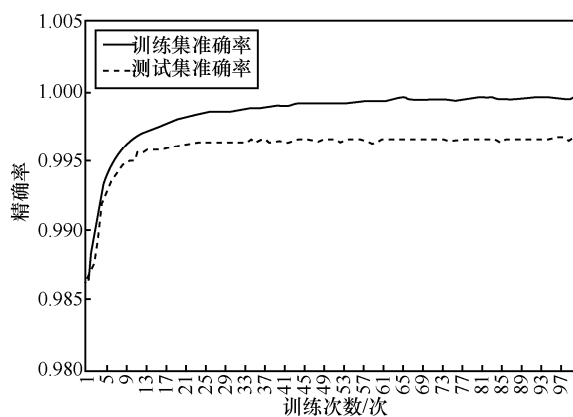


图4 深度神经网络模型链路准确率统计

另外, 利用上述训练好的模型, 使用 DL-TEA 对 10 组测试集中的请求进行求解, 最后取 10 组结果的平均值作为最终仿真结果。

DL-TEA 和贪婪算法、模拟退火算法的请求时延对比如图 5 所示。从图 5 中能够看出贪婪算法的请求平均时延最高, DL-TEA 算法的请求平均时延低于贪婪算法, 可以得出 DL-TEA 算法能够更好地



保证业务的 QoS。但是, 由于 DL-TEA 算法是基于模拟退火算法结果进行训练的, 所以请求平均时延略高于模拟退火算法。

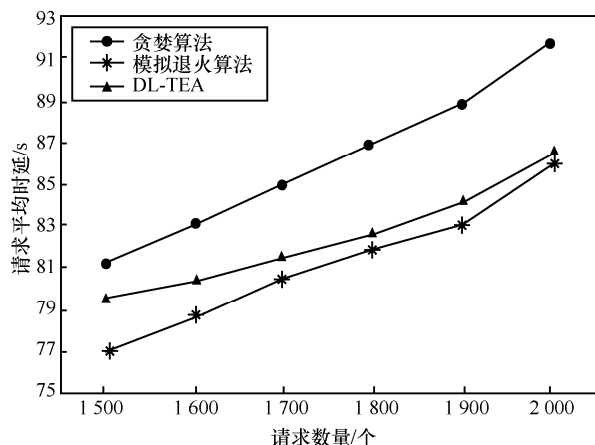


图5 请求平均时延

3 种算法的耗时对比如图 6 所示, 能够看出 DL-TEA 和贪婪算法的耗时基本一致, 甚至耗时更少, 这是因为 DL-TEA 在在线决策阶段, 仅需要根据剪枝后的拓扑进行算路, 拓扑规模会比原拓扑更小, 所以耗时更低。由于模拟退火算法需要不断迭代, 所以耗时最久。这也验证了 DL-TEA 能够满足业务的高动态实时路由需求。

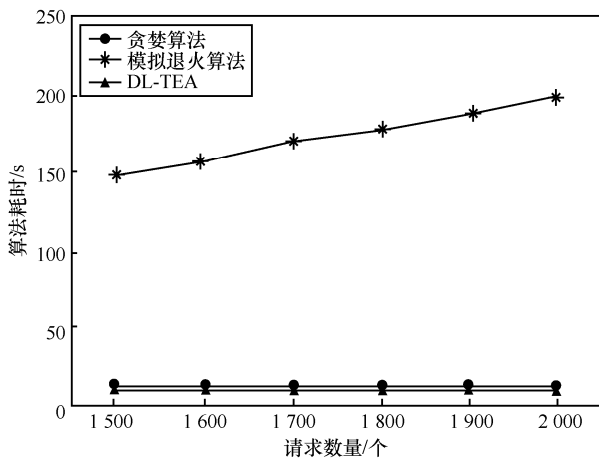


图6 算法耗时对比

带宽占用之和最小场景的仿真结果如图 7 所示。在该场景中, 统计了请求平均的带宽占用情况。从结果中能够看出, DL-TEA 比贪婪算法占

用了更少的带宽资源。换句话说, DL-TEA 能够提升网络的资源利用率。

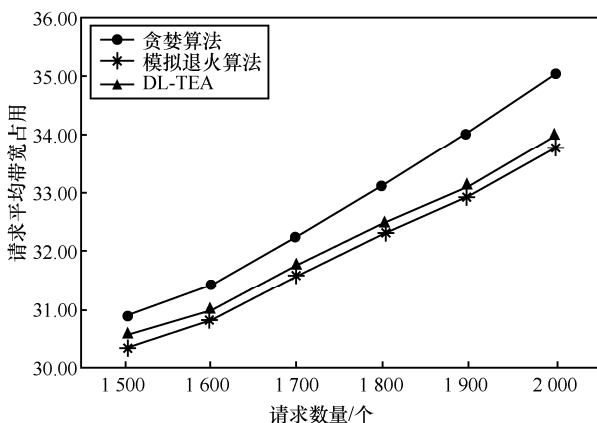


图7 请求平均占用带宽

另外, 从图 5 和图 7 的结果中能够发现, 随着网络负载量的提升, 请求的平均时延和平均带宽的占用均会提升。这主要是因为低负载量的情况下, 网络资源相对充足, 业务均可以选择最小时延或最小跳路径进行传输。当网络负载量增加时, 部分链路会出现资源不足的情况, 导致部分业务需要走其他路径进行传输, 因此请求的平均时延和平均带宽占用均略有提升。

3 种算法在负载均衡场景下的网络阻塞率仿真结果如图 8 所示。从结果中能够看出, DL-TEA 算法的阻塞率同样低于贪婪算法, 并且略高于启发式算法。综上所述, DL-TEA 算法与贪婪算法相比, 能够保证业务的 QoS, 提高网络的资源利用率, 降低网络阻塞率, 并且满足高动态业务的实时路由需求。

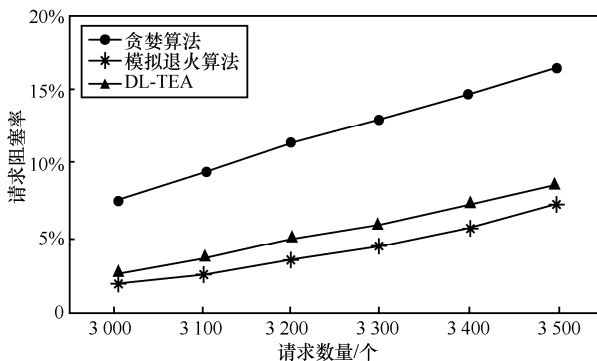


图8 网络阻塞率仿真结果

由于深度模型的输出结果是0到1之间的连续数值,所以还需要设置一个合理的阈值来对拓扑进行剪枝。因此本文统计了3 000个请求负载情况下网络阻塞率和链路保留率之间的关系见表1:当保留的链路太少时,由于网络连通性降低,网络的阻塞率会增加。当仅保留2%左右的链路时,网络阻塞率降至最低,但随着链路保留率的增加,DL-TEA的性能会逐渐退化成贪婪算法。因此,本文的仿真场景中,DL-TEA根据模型输出的链路系数对链路进行排序,然后每次仅保留前2%的链路集合进行路由。另外,还统计了训练集中同源宿请求在不同网络状态下的路径分布情况,发现在该仿真拓扑中,同源宿请求的路径分布相对比较集中,平均每个请求占用的链路数量比例也在2%左右。

表1 链路保留率和网络阻塞率关系

链路保留率	请求阻塞率	链路保留率	请求阻塞率
0	7.37	10	5.37
1	4.63	20	6.03
2	2.70	30	6.07
3	3.33	40	6.63
4	3.80	50	6.40
5	4.70	60	6.03
6	5.17	70	6.33
7	5.27	80	6.43
8	5.27	90	7.1
9	5.37	100	7.37

另外值得一提的是,在不同组网情况下,链路保留率和算法性能之间的关系也会有不同。在仿真场景中,还测试了200个节点的拓扑,发现链路保留率在30%时DL-TEA算法性能达到最优,因此,DL-TEA需要依据不同的拓扑组网进行调整。

5 结束语

本文提出了基于深度学习的流量工程算法,并通过仿真验证该算法既能够满足高动态业务的

实时路由需求,同时还能够保证业务的QoS、提升网络资源利用率、降低网络阻塞率。但是本文尚未考虑多约束场景下的业务路由问题,后续可以在该方向继续研究。

参考文献:

- [1] MCKEOWN N, ANDERSON T. OpenFlow: enabling innovation in campus networks[C]//Proceedings of ACM SIGCOMM 2008. New York: ACM Press, 2008.
- [2] 王素彬, 朱永庆. SDN与流量精细化运营[J]. 电信科学, 2014, 30(11): 145-153.
WANG S B, ZHU Y Q. SDN and Traffic Fine Operation[J]. Telecommunications Science, 2014, 30(11): 145-153.
- [3] 张奇. SDN在传送网中的关键技术及流量工程应用场景[J]. 电信科学, 2015, 31(Z1): 158-162.
ZHANG Q. Key technologies of SDN in transport network and application scenarios of traffic engineering[J]. Telecommunications Science, 2015, 31(Z1): 158-162.
- [4] 樊自甫, 伍春玲, 王金红. 基于SDN架构的数据中心网络路由算法需求分析[J]. 电信科学, 2015, 31(2): 36-45.
FAN Z F, WU C L, WANG J H. Requirements analysis of data center network routing algorithm based on SDN architecture[J]. Telecommunications Science, 2015, 31(2): 36-45.
- [5] YU M, REXFORD J, FREEDMAN M J, et al. Scalable flow based networking with DIFANE[C]//Proceedings of SIGCOMM 2010. [S.l.:s.n.], 2010: 351-362.
- [6] CURTIS A R, KIM W, YALAGANDULA P. Mahout: low-overhead datacenter traffic management using end-host-based elephant detection[C]//Proceedings of IEEE INFOCOM2011. Piscataway: IEEE Press, 2011: 1629-1637.
- [7] 黄婧洁. SDN网络的流量工程模型及算法研究[D]. 北京: 北京邮电大学, 2015.
HUANG J J. Research on traffic engineering model and algorithms of SDN[D]. Beijing: Beijing University of Posts and Telecommunication, 2015.
- [8] 张嗣宏, 左罗. 基于人工智能的网络智能化发展探讨[J]. 中兴通讯技术, 2019, 25(2): 57-62.
ZHANG S H, ZUO L. Network intelligence based on artificial intelligence[J]. ZTE Technology Journal, 2019, 25(2): 57-62.
- [9] VALADARSKY A, SCHAPIRA M, SHAHAF D, et al. Learning to route[C]//Proceedings of the 16th ACM Workshop. New York: ACM Press, 2017: 185-191.
- [10] 李彦君. 利用机器学习实现快速网络资源分配的研究[D]. 成都: 电子科技大学, 2016.
LI Y J. Research on fast network resource allocation with machine learning[D]. Chendu: University of Electronic Science and Technology of China, 2016.
- [11] XU Z. Experience-driven networking: a deep reinforcement learning based approach[C]//Proceedings of IEEE INFOCOM. Piscataway: IEEE Press, 2018: 1871-1879.
- [12] 李芑. SDN网络中基于机器学习的网络资源分配研究[D]. 南



京: 东南大学, 2018.

LI P. Research on network resource allocation based on machine learning in software defined network[D]. Nanjing: Southeast University, 2018.

- [13] SUN P H, LI J F, et al. SINET: enabling scalable network routing with deep reinforcement learning on partial nodes[C]//Proceedings of SIGCOMM 2019. [S.l.: s.n.], 2019: 88-89.

[作者简介]



胡道允 (1990-), 男, 移动网络和移动多媒体技术国家重点实验室工程师, 主要研究方向为网络管控系统、智能网络算法、智能运维等。



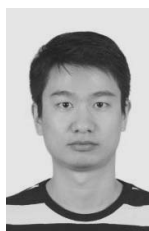
齐进 (1971-), 男, 博士, 移动网络和移动多媒体技术国家重点实验室工程师, 主要研究方向为网络控制与管理、大数据分析、智能运维等。



陆钱春 (1986-), 女, 移动网络和移动多媒体技术国家重点实验室工程师, 主要研究方向为智能网络优化。



李锋 (1978-), 男, 移动网络和移动多媒体技术国家重点实验室工程师, 主要研究方向为基于深度学习的控制器策略闭环、控制器自动化能力演进等。



房红强 (1995-), 男, 中国科学技术大学硕士生, 主要研究方向为光通信网络和机器学习等。