



计算机科学

COMPUTER SCIENCE

知识定义算力网络下的重击流智能流量调度机制

粘英璞, 易波, 李沛辰, 王兴伟, 黄敏

引用本文

粘英璞, 易波, 李沛辰, 王兴伟, 黄敏. 知识定义算力网络下的重击流智能流量调度机制[J]. 计算机科学, 2024, 51(12): 20-29.

NIAN Yingpu, Yi Bo, LI Peichen, WANG Xingwei, HUANG Min. Knowledge-defined Intelligent Traffic Scheduling Mechanism in Computing Network [J]. Computer Science, 2024, 51(12): 20-29.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于大数据的进化网络影响力分析研究综述](#)

Survey of Influence Analysis of Evolutionary Network Based on Big Data

计算机科学, 2022, 49(8): 1-11. <https://doi.org/10.11896/jsjcx.210700240>

[企业内部工业互联网现状与发展对策研究](#)

Study on Development Status and Countermeasures of Industrial Intranet in Enterprises

计算机科学, 2022, 49(7): 1-9. <https://doi.org/10.11896/jsjcx.210900029>

[基于GR-AD-KNN算法的IPv6网络DoS入侵检测技术研究](#)

Research on DoS Intrusion Detection Technology of IPv6 Network Based on GR-AD-KNN Algorithm

计算机科学, 2021, 48(6A): 524-528. <https://doi.org/10.11896/jsjcx.200500001>

[可重构数据中心网络研究综述](#)

State-of-the-art Survey on Reconfigurable Data Center Networks

计算机科学, 2021, 48(3): 246-258. <https://doi.org/10.11896/jsjcx.201100038>

[基于自适应虚拟机迁移的云资源调度机制](#)

Cloud Resource Scheduling Mechanism Based on Adaptive Virtual Machine Migration

计算机科学, 2020, 47(9): 238-245. <https://doi.org/10.11896/jsjcx.190900189>

知识定义算力网络下的重击流智能流量调度机制

粘英璞¹ 易波¹ 李沛辰¹ 王兴伟¹ 黄敏²

1 东北大学计算机科学与工程学院 沈阳 110169

2 东北大学信息科学与工程学院 沈阳 110819

(nianyingpu0124@163.com)

摘要 当前,知识定义网络赋能 AI 技术发展,算力网络提供 AI 所需算力资源,二者逐渐趋于融合,形成了知识定义算力网络(Knowledge Defined Computing Networking,KDCN)。KDCN 赋能发展了诸多新型网络应用,如元宇宙、AR/VR、东数西算等,这些新型应用对算力资源和网络资源有极大的需求,被称为重击流(Heavy Hitter,HH)。HH 流的存在严重加剧了 KDCN 网络的拥塞情况。针对这一挑战,提出了一种智能流量调度机制,旨在通过深度 Q 神经网络来解决 KDCN 中的拥塞问题。相较于离线训练过程,通过流量数据检测与采集、在模型训练和拥塞流调决策之间建立实时闭环,来实现深度 Q 神经网络模型的在线训练。基于该闭环控制,智能流调模型通过不断学习可以实现持续演化,并用于提供实时决策。实验结果表明,该算法在资源利用率、吞吐量、平均丢包率等方面优于现有方法。

关键词:知识定义算力网络;深度 Q 神经网络;智能拥塞流调;重击流;闭环控制

中图分类号 TP391

Knowledge-defined Intelligent Traffic Scheduling Mechanism in Computing Network

NIAN Yingpu¹, YI Bo¹, LI Peichen¹, WANG Xingwei¹ and HUANG Min²

1 School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China

2 School of Information Science and Engineering, Northeastern University, Shenyang 110819, China

Abstract At present, the knowledge-defined network empowers the development of AI technology, and the computing power network provides the computing power resources required by AI. The two gradually tend to integrate to form the knowledge defined computing networking (KDCN). KDCN has empowered the development of many new network applications, such as the meta-universe, AR/VR, east-west computing. These new applications have a great demand for computing power resources and network resources, and are called heavy hitter (HH). The existence of HH flow seriously aggravates the congestion of KDCN network. In response to this challenge, this paper proposes an intelligent traffic scheduling mechanism, which aims to solve the congestion problem in KDCN through deep Q neural networks. Compared with the offline training process, a real-time closed loop is established between traffic data detection and acquisition, model training, and congestion flow modulation decision-making to realize the online training of the deep Q neural network model. Based on this closed-loop control, the intelligent flow modulation model can realize continuous evolution through continuous learning, and is used to provide real-time decision-making. Experimental results show that the proposed algorithm is superior to the existing methods in resource utilization, throughput, average packet loss rate and so on.

Keywords Knowledge definition computing networking, Deep Q neural networks, Intelligent congestion flow modulation, Heavy hitter flow, Closed loop control

1 引言

随着 5G/6G、边缘计算^[1]等互联网技术的快速发展,海量终端用户和设备进入互联网,网络中的计算能力在云计算

及边缘计算的基础上将进一步下沉,计算形态将向着“云-边缘”泛在分布的趋势发展,计算与网络的融合将更加紧密,并逐步趋于算网融合形态。网络流量呈指数级增长,随之而来的是高丢包率、高端到端时延、低吞吐量、单一形态和单一

到稿日期:2024-03-11 返修日期:2024-08-10

基金项目:国家重点研发计划(2022YFB2901303);国家自然科学基金(62032013, U22A2004, 92267206);中央高校基本科研业务费项目(N2316006);辽宁省科技联合计划项目(2023-MSBA-079)

This work was supported by the National Key Research and Development Program of China(2022YFB2901303), National Natural Science Foundation of China(62032013, U22A2004, 92267206), Fundamental Research Funds for the Central Universities of China(N2316006) and Liaoning Provincial Science and Technology Joint Program(2023-MSBA-079).

通信作者:易波(yibo@cse.neu.edu.cn)

提供主体的发展都进入了瓶颈期等问题。针对这些挑战,许多新颖的工作被提出,因为大型的算力业务往往需要通过计算联网来实现,所以业界提出了算力网络(Computing Networking, CN)的思想。其中云网融合是近年来具有代表性的理念,SDN/NFV是云网融合最重要的技术支撑。软件定义网络SDN^[2]打破了传统网络的垂直集成架构,将网络转发与控制分离,形成一个集中的控制平面,并通过南接口OpenFlow^[3]引入网络的编程能力。基于上述定义,SDN控制器实现了全局控制能力,可以灵活地改变流量路由,从而缓解甚至避免底层网络可能出现的拥塞情况。

但是SDN本身并不支持智能决策,因此提出了一种新的网络范式—知识定义网络(Knowledge-Defined Networking, KDN)^[4]。实际上,KDN继承了SDN的控制和转发分离的特点。此外,还介绍了一个新的平面,即知识平面和AI驱动的^[5]。知识平面利用各种机器学习技术为SDN控制平面提供决策指导,实现网络的智能管控。尽管如此,在5G/6G技术的帮助下,大量新型网络应用(如元宇宙、AR/VR)不断出现,这些新型应用对网络算力和带宽资源均有着极大的需求。这促使知识定义网络和算力网络逐渐融合,形成了知识定义算力网络(KDCN)。二者的结合,旨在实现更智能、灵活和高效的计算资源管理和任务调度。它基于知识图谱的思想,将计算任务、资源节点以及它们之间的关系映射到图形化结构中,从而实现对计算任务和资源的智能化管理。知识图谱可以描述计算任务的属性、依赖关系、优先级等信息,以及计算节点的能力、可用性、位置等属性。基于这些信息来动态分配和调度计算任务,根据任务的需求和资源的可用性进行智能的决策和优化。

在KDCN中,对算力和网络资源均有着极大需求的流量被称为重击(HH)流^[6],其可能占据大量的网络带宽,导致其他流受到影响,出现延迟或者丢包的情况。并且,如果重击流对应的任务需要大量的计算资源和执行时间,可能会影响其他任务的调度和执行。

因此,通过设计和实现及时拥塞检测与智能拥塞流调,来提高KDCN业务质量。首先,引入集中式控制架构,为快速拥塞流调提供全局网络状态信息;其次,利用知识平面智能地分析网络状态的变化,从而及时、智能地检测网络拥塞;再次,通过在知识平面中应用深度Q神经网络,为HH流量寻找最优路由路径提供智能决策,并缓解甚至避免网络拥塞;最后,通过南接口将智能决策方案部署在转发设备中,指导转发过程,最大限度地减少网络拥塞、提高网络性能。

本文的主要贡献包括3个方面:

1)设计了一种面向KDCN的流量调度框架,该框架主要由网络监控、拥塞分析、拥塞决策以及决策执行4个部分组成,各部分之间相互协调,形成KDCN拥塞流调闭环系统。另外,该框架支持通过计算成网的方式,综合泛在闲散算力来缓解算力需求的潮汐效应,提高全网算力资源利用率。

2)分别通过计算链路利用率和数据流传输速率来检测拥塞链路与HH流,并提出了基于阈值的HH流分类方法与基于知识库的HH流分类方法。

3)设计了一种基于深度Q网络(Deep Q Network, DQN)

的拥塞决策机制,以实现通过网络拥塞的智能化流调。通过拥塞信息预处理对HH流进行优先级排序以及知识抽取,并通过D3QN(Dueling Double DQN)拥塞决策算法依次为拥塞链路上经过的HH流计算最优的路径策略,从而降低拥塞链路的带宽利用率,缓解网络拥塞,提高网络性能。

2 相关工作

2.1 研究现状

网络拥塞问题长期存在,无法消除。许多学者从多个角度对网络拥塞进行了研究,并提出了多种改进方案^[7-10]。在利用机器学习技术解决网络拥塞方面,文献[8]提出了一种新的拥塞架构DeepCC,文献[9]提出了一种基于DQN算法的TCP拥塞流调算法,文献[10]提出了一种利用主动队列管理(AQM)配合传输控制协议的拥塞控制机制。文献[11]介绍了一种名为CONGA的网络分布式拥塞感知负载均衡机制,CONGA将TCP流分为流波,并根据远程交换机的反馈将流波分配到路径上,从而实现了路径间的负载均衡。

KDCN是一种特殊的网络架构,它通过引入领域知识,对计算任务的分配和处理过程进行定义。这些领域知识可以是关于任务特性、网络拓扑等方面的先验知识,而其他基于DRL的闭环算法通常是通过数据驱动的方式学习网络拥塞流调策略,对网络环境和任务特性没有显式的先验知识。除此之外,KDCN可以根据任务的特性和需求,结合领域知识,对任务进行智能的调度和资源分配,而常规DRL的闭环算法通常是以一种固化的模式来解决拥塞控制。

文献[12]介绍了一个名为Hedera的可扩展的、动态的流调度系统,该系统通过自适应地调度多级交换结构来高效利用数据中心的网络资源。Hedera系统是自适应流调度系统,不同于固定规则的静态负载平衡机制,它能够动态地通过流量监控和调度来实现网络资源的优化利用。

在算网融合背景下,资源的调度是需要处理的重要问题之一。文献[13]提出了基于可用带宽调度数据流的分布式框架来调度rat流,本文提出了一种称为Sieve的轻量级采样机制来调度大象流量,文献[14]设计了一个分布式任务调度算法(Cost Aware Task Scheduling, CATS),在系统平均代价方面提供近似最优的性能。

在路由的选择以及路径优化方面,文献[15]提出了一种基于流分类的深度神经网络路径优化模型,可以根据QoS要求选择路由;文献[16]提出了一种协作式多域路由框架,该框架利用多个网络域操作员之间的协作,使整体网络利用率最大化;文献[17]提出了一种用于部署弹性路由的框架,其在SDN/NFV控制和编排架构中展示NFV的全闭环控制环路,易于管理,增强了数据包处理能力。

路由资源和计算资源的调配和利用是现在算网融合下控制网络拥塞的主要途径。文献[18]提出了一种移动算力网络,将通信资源和计算资源联合优化,按需动态调整业务调度和路由策略,实现资源高利用率和用户体验的优化。

目前,KDN在负载均衡、路由优化和数据流识别等方面已经取得了很大的进展。在基于KDN的负载均衡方面,文献[19]提出了一种基于人工神经网络(Artificial Neural Net-

work, ANN)的负载均衡方法,通过创建 ANN 流量行为模型来预测网络性能,并选择负载最小的路径进行流量传输;在基于 KDN 的路由优化方面,文献[20]提出了一种基于 DRL 的路由机制,通过卷积神经网络学习网络中数据流之间的相关性,从而提供更好的路由配置,提高 QoS 感知路由的性能;在基于 KDN 的数据流识别方面,文献[21]提出了一种在 KDN 知识平面上利用机器学习技术识别 HH 流(即消耗大量网络资源的流量)的方法,并实现了一种能够根据网络行为有效动态识别 HH 流的智能系统。

本文分别从实时拥塞检测和智能拥塞决策两个方面设计并实现了 KDCN 环境下的智能拥塞流调机制,相较于现有工作,其能够同时保障网络拥塞检测的时效性和拥塞决策的智能性。

2.2 问题与挑战

随着网络技术的飞速进步,网络流量呈现爆炸性增长,这对网络拥塞控制机制的即时性和智能化提出了新的要求。软件定义网络作为下一代网络的关键技术,在网络流量调度、路径优化和拥塞控制等领域已经得到广泛应用。知识定义算力网络则是在 SDN 基础上演进而来的新型架构,其知识算力平面利用机器学习技术,能够智能化

解决网络中的各种问题。将这两者的优势有效结合,是解决网络拥塞问题的重要思路。

然而,在 KDCN 中,特别是对高带宽高算力 HH 流的拥塞管理面临诸多挑战。1)HH 流的需求超出传统基础设施的能力范围,可能导致业务分布不均衡。2)高算力与大带宽需求加剧了网络拥塞,增加了运维成本。因此,解决 KDCN 中 HH 流引发的拥塞问题显得尤为重要。3)通常,流量调度是从端到端的角度来实现,例如广泛使用的 TCP 流量调度算法^[22-26]就试图优化任意网络两终端之间的拥塞。但是, TCP 拥塞流调算法缺乏全局网络视图,无法快速发现拥塞,这就导致实际发生拥塞的时间点与检测到拥塞的时间点之间的延迟变长,从而增加了发送的数据包数量,最终加剧了网络拥塞。在大规模网络环境下,如何保证拥塞控制机制的适应性和有效性也是一个需要解决的问题。

3 整体框架

3.1 整体框架

本文提出的 KDCN 整体框架主要由网络状态监控、实时拥塞链路和 HH 流量检测、拥塞决策等部分构成。整体框架如图 1 所示。

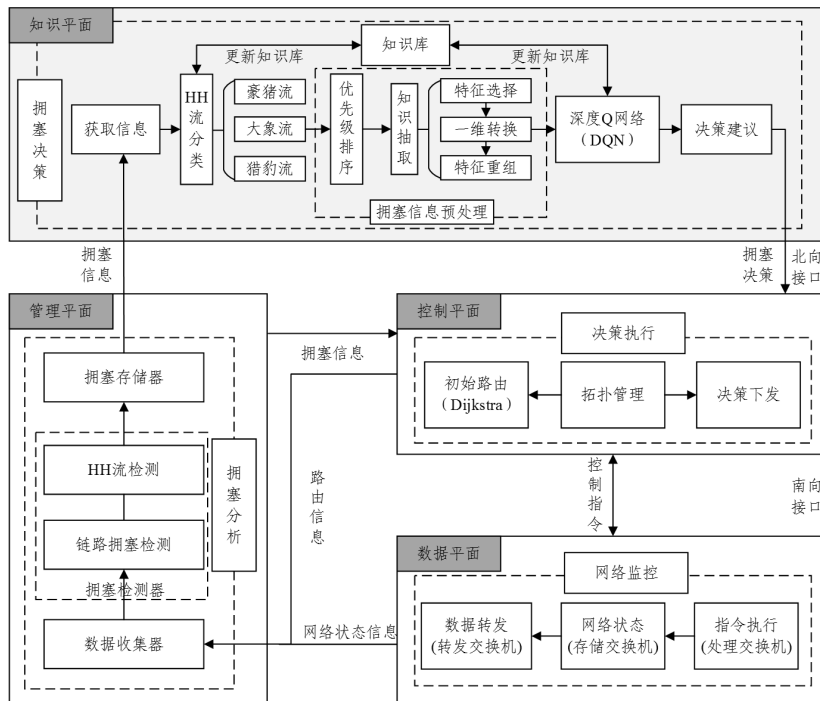


图 1 KDCN 总体框架

Fig. 1 General framework of KDCN

首先,数据平面的网络监控实时监控网络并将拓扑信息传递给管理平面的拥塞分析模块。其次,拥塞分析模块定期采集网络状态信息,进行链路拥塞和 HH 流量检测,获取拥塞信息,并将其发送至拥塞决策和知识控制平面。随后,拥塞决策模块根据 HH 流对拥塞信息进行分类、预处理,并形成拥塞知识。它利用基于深度 Q 网络的 D3QN 算法处理拥塞情况,并将决策建议发送至控制平面。最后,决策执行模块将决策建议转化为具体的执行策略,并发送至数据平面,形成一个闭环的可演进体系。通过持续的循环演进,该体系能够

支持不同场景的需求。

本文设计了基于四面一体的 KDCN 架构,包括数据平面、控制平面、管理平面和知识平面。数据平面负责执行来自控制平面和管理平面的指令,同时向知识平面提供用于凝练知识的元数据;控制平面负责向数据平面提供拥塞和流量调度的决策方案,同时从知识平面获取用于训练模型的知识数据;管理平面定期从数据平面和控制平面获取全局网络状态信息,并实时监控网络拥塞情况;知识平面主要从数据平面获取元数据,凝练知识,并提供给控制平面和管理平面进行

决策。整体 KDCN 架构的 4 个平面相互支持和交互,共同实现优化网络拥塞的目标。

3.2 工作流程

本文提出的 KDCN 环境下的拥塞流调机制的总体流程如图 2 所示。

1)决策执行模块的拓扑管理首先对网络拓扑进行发现,得到网络拓扑。

2)位于决策执行模块的初始路由通过执行 Dijkstra 算法获得数据流的初始路由由状态。

3)存储交换机存储网络状态并更新网络状态信息,包括链路信息表和数据流信息表。该模块转发交换机转发信息,并将更新后的网络状态信息转发给网络分析模块。

4)拥塞分析模块的数据采集器采集数据平面和控制平面的数据,并传输给拥塞检测器。拥塞检测器通过链路拥塞检测确定拥塞链路,然后通过 HH 流检测获取拥塞连接上的 HH 流集,并将其发送到拥塞内存中存储。拥塞内存存储拥塞信息,并将其发送给拓扑管理部分的拥塞决策模块和决策执行模块。

5)拥塞决策模块对拥塞信息中的 HH 流进行分类,并对拥塞信息进行预处理,形成拥塞知识。拥塞分析模块的 D3QN 决策以拥塞知识为输入,为排序后的 HH 流选择最优路径,更新知识库,并向决策执行模块提交决策建议。

6)最后,决策执行模块的决策传递部分接收决策建议,并将其转换为具体的执行策略发送到数据平面。网络监控模块的处理交换机依次执行决策执行模块下发的执行策略。决策执行模块的拓扑管理部分根据拥塞信息更新网络拓扑状态。

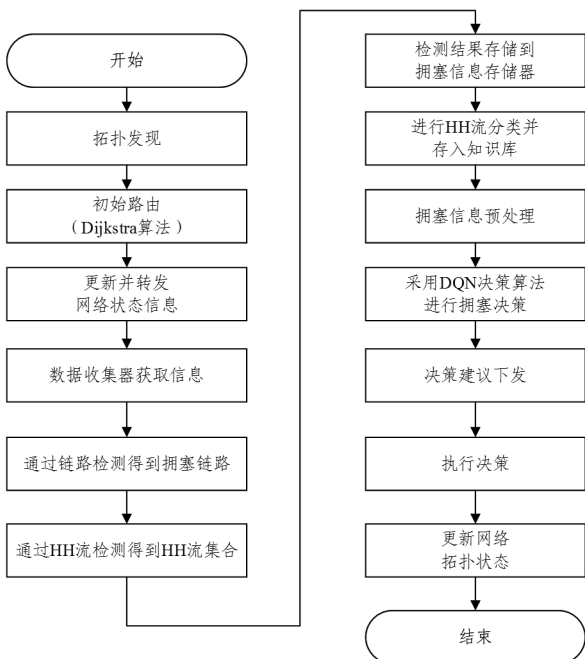


图 2 KDCN 环境下的拥塞流调机制流程

Fig. 2 Flow of congestion traffic scheduling mechanism in KDCN environment

4 拥塞检测和 HH 流分类

面向 KDCN 环境的拥塞检测和 HH 流分类机制主要

功能包括实时网络状态分析、网络拥塞链路发现和拥塞 HH 流发现,以及对 HH 流进行分类。主要过程则由 3 部分组成:链路拥塞检测、HH 流检测和 HH 流分类。

4.1 链路拥塞检测

链路拥塞检测基于链路的带宽利用情况来判断,当网络拓扑中某条链路的带宽利用率过高时,若有新的数据流或者现有数据流的传输速率突然提高,可能导致该链路负载超出其能承受的最大带宽容量,从而引发网络拥塞。链路拥塞检测逐一评估网络拓扑中的各条链路,一旦某条链路的带宽利用率超过设定的拥塞阈值,其就会被标识为拥塞链路。

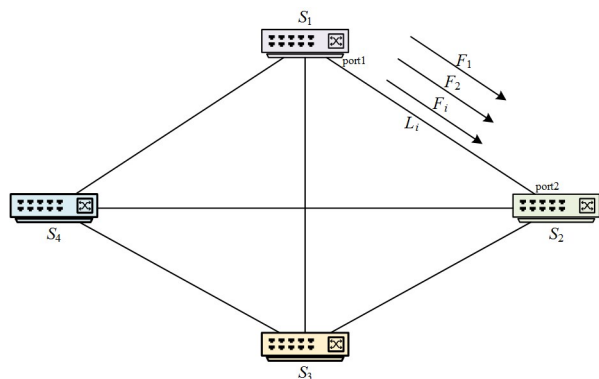


图 3 拓扑结构示例

Fig. 3 Topology illustration

以图 3 所示的拓扑结构为例,链路 L_i 的源交换机为 S_1 、源端口为 S_2 、目的端口为 $Port_1$,拥塞检测的周期性执行时间为 t ,从链路信息表中获得链路 L_i 当前周期中传输字节数为 $Byte_{L_i}^{cur}$,链路 L_i 上一周期总传输字节数为 $Byte_{L_i}^{pre}$,则可计算出链路 L_i 的负载带宽 $load_{L_i}$,计算式如式(1)所示:

$$load_{L_i} = \frac{Byte_{L_i}^{cur} - Byte_{L_i}^{pre}}{t} \quad (1)$$

然后,计算每条链路的带宽利用率。根据链路 L_i 负载带宽和链路 L_i 的最大容量带宽 Ca_{L_i} ,可以计算出链路 L_i 的带宽利用率 $useRate_{L_i}$,计算式如式(2)所示:

$$useRate_{L_i} = \frac{load_{L_i}}{Ca_{L_i}} \times 100\% \quad (2)$$

若链路 L_i 的 $useRate_{L_i}$ 大于链路拥塞阈值 T_{link} ,则将 L_i 标记为拥塞链路。依次计算网络拓扑中每条链路的带宽利用率并与 T_{link} 进行比较,返回拥塞链路信息。若网络拓扑中不存在拥塞链路,则进行下一周期的链路拥塞检测。

4.2 HH 流检测

HH 流检测根据数据流的传输速率来判断数据流是否为 HH 流。定期由数据采集器从数据平面获取链路拥塞的源交换机和目的交换机上的数据流信息表。通过将每个数据流的源 IP 地址和目的 IP 地址与拥塞链路的源 IP 地址和目的 IP 地址进行匹配,确定通过拥塞链路的数据流集。每个数据流的传输速率是根据当前数据流的传输字节数与前一个数据流的传输字节数之差计算得出的。

如图 3 所示,假设链路 L_i 是拥塞链路,则获取资源交换机 S_1 和目的交换机 S_2 的数据流信息表,并与拥塞链路 L_i 的源端口 $Port_1$ 和目的端口 $Port_2$ 进行匹配,得到拥塞链路 L_i 上的数据流集合为 $\{F_1, F_2, \dots, F_i\}$,然后依次计算每条数据

流的传输速率。拥塞检测的周期执行时间同样为 t , 数据流 F_i 的当前周期总传输字节数为 $byte_{F_i}^{cur}$, 数据流 F_i 的上一周期总传输字节数为 $byte_{F_i}^{pre}$, 因此可以计算出数据流 F_i 的传输速率 $rate_{F_i}$, 计算式如式(3)所示:

$$rate_{F_i} = \frac{byte_{F_i}^{cur} - byte_{F_i}^{pre}}{t} \quad (3)$$

数据流的传输速率是判断数据流是否为 HH 流的重要标志。若数据流 F_i 的传输速率 $rate_{F_i}$ 超过 HH 流阈值 T_{HH} , 则将数据流 F_i 判定为 HH 流。依次遍历拥塞链路中的所有数据流, 计算每条数据流的传输速率与其 HH 流阈值进行比较, 并返回拥塞链路上的 HH 流集合。

4.3 HH 流分类

HH 流根据属性的不同, 可分为大象流、豪猪流和猎豹流 3 种类型。大象流指在网络上传输的高占用、持久的数据流; 豪猪流指在网络上传输高占用、快速、突发的数据流; 猎豹流指数据流在网络上快速、突然的传输。

传统的网络拥塞检测方法大多是在终端设备进行检测, 然而这种检测方法一般只能在拥塞持续一段时间后才能检测到, 非常迟钝且具有严重的滞后性, 往往会出现丢包现象, 严重时甚至会造系统崩溃的后果。因此本文提出了基于阈值的 HH 流量分类和基于知识库的 HH 流量分类两种分类方法。

1) 基于阈值的 HH 流分类

在基于阈值的 HH 流分类中设置了两种 HH 流分类阈值, 分别是 HH 流分类的速率阈值 T_{rate}^{HH} 以及 HH 流分类的生存周期阈值 $T_{duration}^{HH}$ 。

在基于阈值的 HH 流分类中, 当 HH 流的持续时间大于 HH 流分类的生存周期阈值 $T_{duration}^{HH}$ 时, HH 流被判定为大象流; 当 HH 流的传输速率大于 HH 流分类的速率阈值 T_{rate}^{HH} 且 HH 流的持续时间小于 HH 流分类的生存周期阈值 $T_{duration}^{HH}$ 时, HH 流被判定为豪猪流; 当 HH 流的传输速率介于 HH 流阈值 T_{HH} 和 HH 流分类的速率阈值 T_{rate}^{HH} 之间且 HH 流的持续时间小于 HH 流分类的生存周期阈值 $T_{duration}^{HH}$ 时, HH 流被判定为猎豹流。

2) 基于知识库的 HH 流分类

在 HH 流分类中, 由于拥塞检测是周期性的, 因此有可能在传输结束之前进行 HH 流分类。基于阈值的 HH 流分类是根据生命周期分类指标设置的, 如果简单使用基于阈值的 HH 流分类方法, 容易对涉及生命周期属性的 HH 流进行错分类, 从而降低 HH 流分类的准确性。例如: 原本为大象流的 HH 流如果在传输过程进行 HH 流分类, 则可能被识别为豪猪流或猎豹流。

针对这个问题, 本文设计了一种基于知识库的 HH 流分类, 在每次 HH 流分类结束后, 将 HH 流的分类结果以四元组的形式存入知识库中, 具体存储形式如式(4)所示:

$$HH_{cls} = \{flag_{HH}, rate_{HH}, duration_{HH}, type_{HH}\} \quad (4)$$

其中, $flag_{HH}$ 为 HH 流的标识, $rate_{HH}$ 为 HH 流的传输速率, $duration_{HH}$ 为 HH 流的生存周期, $type_{HH}$ 为 HH 流的分类结果。

在进行下一次 HH 流分类时, 将以知识库中 HH 流的

历史分类记录为依据, 首先通过五元组进行匹配确定唯一一条 HH 流, 再从知识库历史分类记录中获取该条 HH 流可能属于的 HH 流类型。同时, 统计 HH 流类型出现的概率, 并从中选择概率最大的 HH 流类型作为该 HH 流的最终类别, 具体的计算方法如式(5)所示:

$$type_{HH} \leftarrow Type_{HH}(\max\{P_{ele}, P_{por}, P_{che}\}) \quad (5)$$

其中, $type_{HH}$ 为 HH 流的类型; P_{ele} , P_{por} , P_{che} 分别指大象流、豪猪流、猎豹流在该条 HH 流的历史分类记录中出现的概率。

5 基于深度 Q 网络的拥塞决策机制

在网络中, 数据包的传输路径与流量负载是不断变化的, 优化传输路径的问题并不是一个静态问题, 而是一个动态问题, 在实际情况中难以直接求解最优路径。然而, 在 KDCN 的场景下, 强化学习算法能够对动态和复杂的问题进行快速有效的求解。与传统优化算法相比, 强化学习算法具有更强的适应性和快速响应能力, 更符合实际应用的需求。

因此, 本文基于 D3QN 提出了一种面向 KDCN 的智能拥塞决策机制 (Congestion Decision Mechanism Based on Dueling Double Deep Q Network, CDMB-D3QN)。该机制的主要功能是为 HH 流做出拥塞决策, 降低拥塞链路的利用率, 解决网络拥塞问题。该机制包括两部分: 拥塞信息预处理和 D3QN 拥塞决策算法 (D3QN Congestion Decision Algorithm, D3QN-CDA)。

D3QN 首先针对一些问题进行了改进, 能够更加准确、有效地进行训练和预测; 其次, D3QN 算法可以对多个状态进行深度学习, 同时利用优先级经验回收, 有效避免了传统强化学习算法的缺陷; 最后, D3QN 算法在强化学习范式中已经被广泛应用, 具有良好的稳定性和可靠性。在处理智能拥塞流量调度机制的情况下, D3QN 算法可以学习适应于网络动态变化的最优策略, 以提高网络性能和可靠性。这也是在 KDCN 场景下选取 D3QN 作为智能拥塞流量调度机制训练模型的原因之一。

5.1 拥塞信息预处理

在实现 D3QN 拥塞决策算法之前, 需要对获得的拥塞信息 (包括拥塞链路、HH 流集和 HH 流分类结果) 进行一系列预处理, 形成结构化的拥塞知识, 作为后续 D3QN 拥塞决策模型的输入。拥塞信息预处理包括排序和知识提取两部分。主要处理流程如图 4 所示。

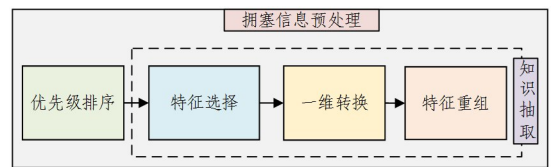


图 4 拥塞信息预处理流程

Fig. 4 Preprocessing process of congestion information

1) 优先级排序: 对分类后的 HH 流根据其其对网络状态的影响程度进行优先级排序, 以便在后续采用 D3QN 拥塞决策算法时, 优先处理对网络状态影响较大的 HH 流。本文采用了两级排序方法, 对分类后的 HH 流进行优先级排序, 具体

分为 HH 流分类间的优先级排序和 HH 流分类内部的优先级排序,其中 HH 流分类间的排序优先级高于 HH 流分类内部的排序。

HH 流分类间的优先级排序:根据 HH 流属性对网络状态的影响程度进行分类,其中,HH 流的大小和速率是对网络状态影响最大的属性,而 HH 流的突发性也对网络状态有较大影响,HH 流的生存周期对网络状态的影响最小。因此,在进行 HH 流分类间的优先级排序时,对高占用、高速率且高突发性的豪猪流赋予最高优先级;对高占用、长周期的大象流赋予次高优先级;对高速率、高突发性的猎豹流赋予最低优先级。综上所述,本文采用的 HH 流分类优先级为:豪猪流 > 大象流 > 猎豹流。

HH 流分类内部的优先级排序:在 HH 流分类内部,根据其特殊属性从高到低进行排序。其中,大象流根据数据流的大小(即带宽占用量)进行排序;豪猪流同样根据数据流的大小进行排序;猎豹流则根据数据流的传输速率进行排序。

2)知识抽取:对经过优先级排序后的 HH 流进行知识抽取以形成结构化的拥塞知识,知识抽取包括特征选择、一维转换以及特征重组 3 个步骤。

特征选择就是在原始特征集中选取有用的特征以减小数据维度。首先从经过优先级排序后的 HH 流表中提取每条 HH 流的传输速率。设 HH_{sort} 为排序后的 HH 流集合, $HH_{\text{sort}} = \{hh_1, hh_2, \dots, hh_n\}$, 从中提取 HH 流的速率特征,表示为:

$$\mathbf{RT} = [\text{rate}_{hh_1}, \text{rate}_{hh_2}, \dots, \text{rate}_{hh_n}]^T \quad (6)$$

其中, i 表示 HH 流信号的优先级序号, n 表示 HH 流的条数, \mathbf{RT} 为 $n \times 1$ 阶的矩阵。从链路信息表 $Link = \{link_1, link_2, \dots, link_m\}$ 中提取全局链路的最大容量带宽特征 CA_{link} 和链路负载特征 $load_{\text{link}}$, 分别如式(7)、式(8)所示:

$$\mathbf{CA} = [CA_{link_1}, CA_{link_2}, \dots, CA_{link_m}]^T \quad (7)$$

$$\mathbf{LD} = [load_{link_1}, load_{link_2}, \dots, load_{link_m}]^T \quad (8)$$

其中, m 为网络拓扑中链路的条数, \mathbf{A} 和 \mathbf{LD} 均为 $m \times 1$ 阶的矩阵。

转置处理是将经过特征选择后的 $n \times 1$ 阶 HH 流速率矩阵转置为 $1 \times n$ 阶的一维矩阵,将 $m \times 1$ 阶链路最大容量带宽矩阵和链路负载矩阵转置为 $1 \times m$ 阶的一维矩阵。经转置处理后的 HH 流速率矩阵 \mathbf{RT} 、链路最大容量带宽矩阵 \mathbf{CA} 以及链路负载矩阵 \mathbf{LD} 如下所示:

$$\mathbf{RT} = \mathbf{RT}^T = [\text{rate}_{hh_1}, \text{rate}_{hh_2}, \dots, \text{rate}_{hh_n}] \quad (9)$$

$$\mathbf{CA} = \mathbf{CA}^T = [CA_{link_1}, CA_{link_2}, \dots, CA_{link_m}] \quad (10)$$

$$\mathbf{RLD} = \mathbf{LD}^T = [load_{link_1}, load_{link_2}, \dots, load_{link_m}] \quad (11)$$

特征重组是将提取出的 HH 流速率特征、链路最大容量带宽矩阵特征以及链路负载特征进行重新组合。特征重组后的形式如式(12)所示:

$$\mathbf{P} = [\mathbf{RT}, \mathbf{CA}, \mathbf{LD}] \quad (12)$$

5.2 基于 D3QN 的智能拥塞决策算法

基于 D3QN 的智能拥塞决策算法采用改进的深度 Q 网络算法为拥塞链路上的每个 HH 流重新计算最优路径策略,从而降低拥塞链路的带宽利用率,解决网络拥塞问题。本节详细介绍了该算法的决策架构、状态空间、动作空间和奖励函数。

1)D3QN 拥塞决策架构: D3QN 拥塞决策算法根据输入的状态、动作集和奖励函数对模型进行训练,并将训练好的 D3QN 模型存储在知识库中。通过训练好的 Q 网络,可以获得当前状态下所有动作的 Q 值,并选择 Q 值最大的动作作为 HH 流的最优路径策略。D3QN 拥塞决策算法的架构如图 5 所示。

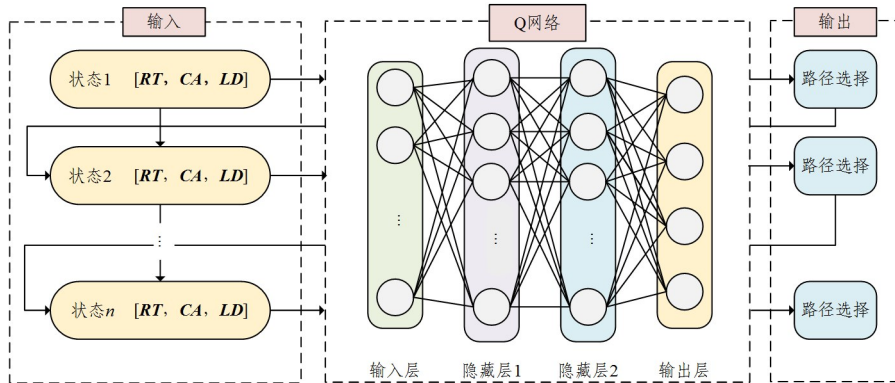


图 5 D3QN 拥塞决策架构

Fig. 5 D3QN congestion decision architecture

D3QN 拥塞决策算法采用了动态规划的思想:将每个回合拆分成多个步骤,每步执行时输出一条 HH 流的路径选择结果,整个回合结束时输出所有 HH 流的路径选择结果。该模型在 Q 网络中采用了两个卷积层和两个全连接层作为隐藏层。

然而,在本文的拥塞决策机制设计中,Q 网络的输入是经过拥塞信息预处理的一维矩阵,因此只使用了两层全连接层作为 Q 网络的隐藏层。因此,D3QN 拥塞决策算法的 Q 网络

由输入层、输出层以及两个隐藏层组成。

2)状态空间:将经过拥塞信息预处理获得的拥塞知识组成的一维矩阵 $[\mathbf{RT}, \mathbf{CA}, \mathbf{LD}]$ 作为 D3QN 拥塞决策模型的初始输入状态。在每一步执行中,输出当前 HH 流的路由决策,整个回合结束时获得对网络中所有 HH 流的路由决策结果。由于一条 HH 流的路径选择会导致网络状态发生变化,进而影响到其他 HH 流的路径选择,因此需要将当前状态下对 HH 流进行路由决策所产生的新链路负载更新到下一步的

状态中。这就需要更新状态中的全局链路可用带宽,即更新 $[RT, CA, LD]$ 中的 LD 矩阵。更新后的全局链路可用矩阵 LD' 表示,如式(13)所示:

$$LD' = [load_{link_1} + load_{link_1}^{new}, \dots, load_{link_m} + load_{link_m}^{new}]^T \quad (13)$$

其中, $load_{link_1}^{new}$ 指当前 HH 流进行路由决策所产生的新的链路负载。

3) 动作空间: 表示所有 HH 流可选择的路径策略的集合。在本文中采用 K 最短路径 (K -Shortest Pathes, KSP) 算法计算动作空间, 以链路负载为权值为每条 HH 流计算 k 条最短路径作为 HH 流可选择的路径策略。KSP 算法利用了递推中偏离路径的思想, 可以在非负权边的有向无环图中找到任意两个节点之间的 k 条最短路径。设 Action 为 n 条 HH 流的动作空间, 则动作空间表示为:

$$Action = \{action_{HH_1}, \dots, action_{HH_n}\} \quad (14)$$

其中, $action_{HH_n}$ 为第 n 条 HH 流的动作集合, 可表示为:

$$action_{HH_n} = path_1, path_2, \dots, path_k \quad (15)$$

其中, $path$ 代表了第 n 条 HH 流的 k 条路径策略。

4) 奖励函数: 奖励函数的设计旨在评估 D3QN 拥塞决策模型在处理 HH 流时的有效性, 以量化其决策质量的好坏。当 HH 流通过分配的路径策略流经某条链路时, 若该链路的带宽利用率超过拥塞阈值, 则会导致新的拥塞链路产生。因此应当保证 HH 流在经过转发路径时, 路径上的所有链路的利用率均小于拥塞阈值。换句话说, 当 HH 流经过路径上负载最大的链路时, 该链路的带宽利用率应当小于拥塞阈值, 如式(16)所示:

$$\frac{\text{Max } load_{link} + rate_{HH_1}}{Ca_{link}} \leq T_{link} \quad (16)$$

上述不等式可化简为:

$$rate_{HH_1} \leq T_{link} \times Ca_{link} - \text{Max } load_{link} \quad (17)$$

其中, $\text{Max } load_{link}$ 表示 HH 流转发路径上的链路最大负载, 如式(18)所示:

$$\text{Max } load_{link} = \max_{link_j \in path} load_{link_j} \quad (18)$$

其中, $path$ 为 HH 流的路径策略, $link$ 为路径策略所经过的链路。

因此, 当存在 HH 流决策路径上的最大负载链路的带宽利用率超过拥塞阈值时, 说明对当前 HH 流做出的决策导致了新的拥塞链路, 则对该决策进行惩罚, 将奖励函数置为 $-\infty$, 具体大小应根据实际网络环境设定。当任意 HH 流决策路径上的最大负载链路的带宽利用率均小于拥塞阈值时, 说明所有的 HH 流均被分配到了合理的路径策略, 网络拥塞得以缓解, 此时应给予奖励。将奖励函数设置为网络拓扑中所有链路的最小剩余可用带宽, 尽量使网络拓扑中链路的带宽分配地更加均衡。奖励函数表达式如式(19)所示:

$$\min_{link_j \in Link} (ca_{link_j} - load_{link_j}) \quad (19)$$

6 性能评估

本章对提出的拥塞流调机制进行了仿实现和性能评估, 包括实验环境、仿实现和性能评估。首先, 介绍了实验环境的参数和拓扑结构; 其次, 对这两种机制进行了仿真和

实现; 最后, 介绍了实验中采用的性能评价指标和基准算法, 并对本文设计的拥塞流调机制进行了实验比较。

6.1 实验参数

1) 拓扑: 针对本文提出的拥塞流调机制, 设计如图 6 所示的网络拓扑结构进行仿真测试。

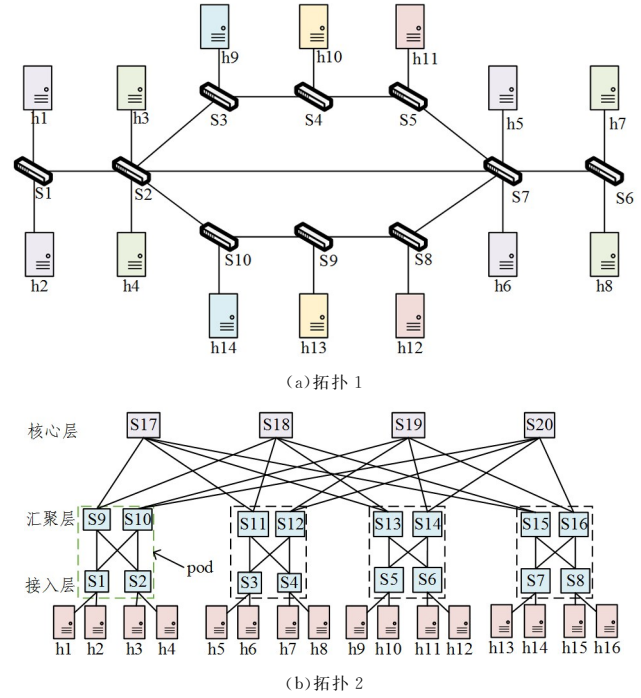


图 6 实验拓扑

Fig. 6 Experiment topology

2) 基于深度 Q 网络的拥塞决策机制实现: 该机制在基于 Mininet/Ryu 框架的 Ubuntu 16.04 系统中实现。此外, 本文还设计了一种基于 iperf 的流量生成工具 my_iperfmulti, 并将该工具编译到了 Mininet 仿真平台中。使用 my_iperfmulti 流量生成工具一次可以生成多条数据流, 并且可以根据需求设置每条流的发送端、接收端、传输速率、开始时间以及生存周期。

拥塞信息预处理包括两部分: 优先级排序和知识提取。首先, 根据不同类型 HH 流对网络拥塞的不同影响, 对分类后的 HH 流进行优先级排序, 然后进行一系列的知识提取, 包括特征提取、一维变换和特征重组。

D3QN 拥塞决策算法用于对网络中的拥塞进行决策, 即重置 HH 流的路径策略, 降低拥塞链路的利用率, 解决网络拥塞问题。该机制利用 Tensorflow 框架编写了 D3QN 拥塞决策算法, 并对 D3QN 模型进行了训练和测试。D3QN 拥塞决策算法训练参数如表 1 所列。

表 1 D3QN 拥塞决策模型训练参数

Table 1 Training parameters of D3QN congestion decision model

参数名	含义	数值
Learning_rate	学习率	0.005
Reward_decay	奖励折扣	0.9
E_greedy	折扣因子	0.9
Replace_target_iter	目标函数更新间隔	500
Memory_size	经验池容量	10000
Batch_size	批处理样本数量	32

6.2 基准算法和评价指标

1) 基准算法:为了评估本文提出的拥塞调度机制的性能,选取了 Q-Learning 算法和 Sarsa (state-action-reward-state-action) 算法两种经典的强化学习方法作为 D3QN 拥塞决策算法 (D3QN-CDA) 的对比算法。Q-Learning 算法是一种离线学习算法,也是一种采用贪婪策略的算法。Q-Learning 算法在每次决策时都是以最大化 Q 值为目标,即直接采用最大折扣未来奖励来更新 Q 表。Sarsa 算法是一种在线学习算法,也是一种采用保守策略的算法。Sarsa 算法是用下一步动作的现实奖励来更新 Q 表,即在更新 Q 表的同时也为未来规划好了动作。

另外,还选取了等价路由算法 (Equal-cost multi-path, ECMP)、随机路由算法 (Random Routing, RR) 以及最短路径优先 (Shortest Path First, SPF) 算法作为本文基于深度 Q 网络的拥塞决策机制 (CDMB-D3QN) 的对比算法。ECMP 算法是等价多路径路由算法,能够在多条路径中同时传输数据,以实现网络负载均衡。RR 算法是随机路由算法,能够按照概率模型随机为数据流选择转发路径,具有较强的鲁棒性,能够在一定程度上缓解网络故障。SPF 算法是最短路径算法,能够为网络拓扑中的任意两个节点选择代价最小的转发路径。

2) 评价指标:链路带宽利用率 (Link bandwidth utilization) 指当前链路的负载占链路最大容量带宽的比例。链路带宽利用率可用来衡量当前链路的负载情况,当链路负载过重时,则认为该链路产生了拥塞。链路带宽利用率的计算式如下所示:

$$LBU_i = \frac{\text{linkload}_i}{\text{linkCa}_i} \times 100\% \quad (20)$$

平均奖励函数 (Average Reward) 是用来衡量决策的好坏、表征模型性能的重要指标。通过计算训练过程中产生的平均奖励值,能够得出模型决策的趋势走向。平均奖励值越大,代表模型的决策效果越好。平均奖励函数的计算式如下所示:

$$\text{reward}_{\text{Average}} = \sum_{i=1}^n \text{reward}_i / n \quad (21)$$

拥塞次数 (Number of Congestion) 指每进行 n 次训练产生拥塞的次数,能够反映模型的拥塞程度,拥塞次数越少代表模型的拥塞程度越轻。其中, n 的取值为 10000。

吞吐量 (Throughput) 指在单位时间内成功传输的数据量,能够反映网络的传输能力。当网络发生拥塞时,吞吐量会比正常网络状态下低。因此,可以根据吞吐量的变化情况来看判断网络的拥塞情况。吞吐量的计算式如下所示:

$$\text{Throughput} = \sum_{i=1}^n \text{throughput}_i \quad (22)$$

丢包率 (Packet Loss Rate) 指在通信过程中未能成功传输的数据量占已传输的数据量的比例。丢包率可以用来衡量网络的性能,丢包率越低代表网络性能越好。当网络产生拥塞时,交换机负载过重,可能会导致部分数据包无法正常传送到目的主机,从而产生丢包现象。因此,可以根据丢包率来判断网络拥塞情况,丢包率越高代表网络拥塞越严重。丢包率的计算式如下所示:

$$PLR = \frac{\sum_{i=1}^n \text{input}_i - \sum_{i=1}^n \text{output}_i}{\sum_{i=1}^n \text{input}_i} \quad (23)$$

6.3 基于深度 Q 网络的拥塞决策机制性能分析

1) D3QN 拥塞决策算法性能分析:为了评价 D3QN 拥塞决策算法 (D3QN-CDA) 的性能,对所设计的 D3QN-CDA 模型进行了 4 万次训练,以训练过程中产生的平均奖励值和拥塞次数作为评价指标。采用 Q-learning 算法和 Sarsa 算法进行对比实验。

平均奖励值是评价强化学习模型性能的重要指标。在训练过程中,每 10000 次训练计算平均奖励值。统计整理后的平均奖励值对比结果如图 7 所示。

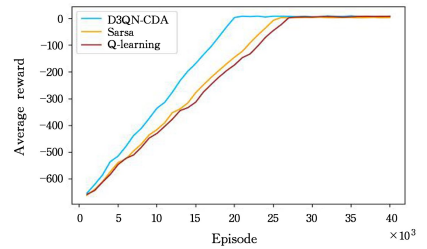


图 7 平均奖励值对比图

Fig. 7 Comparison of average reward value

拥塞时间是评价基于深度 Q 网络的拥塞决策模型有效性的重要指标。在训练过程中,每 10000 次训练中发生拥塞的次数,以及统计排序后拥塞次数的对比结果如图 8 所示。

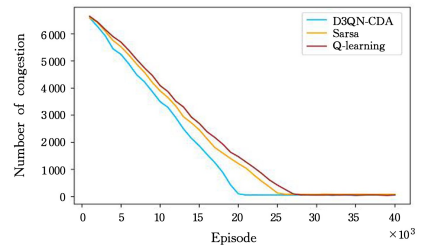


图 8 拥塞次数对比图

Fig. 8 Comparison of congestion times

实验比较了 D3QN-CDA 与 Q-learning 算法和 Sarsa 算法,并对平均奖励值和拥塞时间两个评价指标进行了分析。结果表明,所提出的 D3QN-CDA 模型不仅收敛速度快,而且具有良好的拥塞决策效果,可以实现网络拥塞的智能决策。在收敛速度方面,D3QN-CDA 比 Sarsa 算法快 20%,比 Q-learning 算法快 26%。在拥塞决策效果方面,D3QN-CDA 与 Q-learning 算法的拥塞决策效果相当,且均优于 Sarsa 算法。

此外,在 D3QN 拥塞决策算法 (D3QN-CDA) 之前,本文根据不同类型的 HH 流对网络状态的不同影响,对其进行优先级排序,从而确定 HH 流的处理顺序。为了验证对 HH 流进行优先级排序的必要性,本文将 D3QN-CDA 模型与不进行优先级排序的 D3QN-CDA 模型进行比较。

图 9 展示了 4 组 D3QN-CDA 模型在训练过程中平均奖励值的变化情况。可以看出,在 D3QN-CDA 模型训练前,有必要对 HH 流进行优先级排序,这样可以有效减少训练时间,使模型尽快收敛。

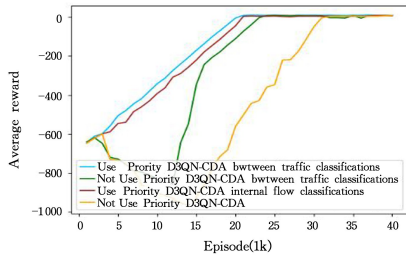


图9 二级优先级排序对比

Fig. 9 Second level priority sorting comparison

表2 数据流详细信息

Table 2 Detail of data flows

数据流	发送端	接收端	速率/Mbps	开始时间/s	生存周期/s
flow1	h3	h5	5	1	15
flow2	h3	h6	4	3	2
flow3	h3	h7	30	5	11
flow4	h3	h8	6	8	2
flow5	h2	h6	40	11	5
flow6	h2	h7	20	13	3

2) 整体性能分析: 为了评估本文提出的基于深度 Q 网络的拥塞决策机制(CDMB-D3QN)的有效性, 采用图 6(a) 拓扑 1 所示的拓扑进行性能测试, 每条链路的最大容量带宽均设置为 100 Mbps。设置了 3 条关键路径, 将链路 S2-S3-S4-S5-S7 之间的路径设定为 Path1, 将链路 S2-S7 之间的路径设定为 Path2, 将 S2-S10-S9-S8-S7 之间的路径设定为 Path3。统计关键路径上带宽利用率的变化情况。

图 10 展示了所有数据流依次注入网络拓扑时, 3 条关键路径的带宽利用率。

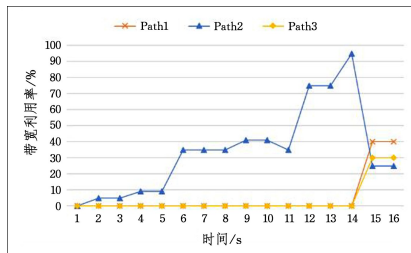


图10 关键路径的带宽利用率

Fig. 10 Bandwidth utilization of critical path

Path1 的带宽利用率提高到 40%, Path2 的带宽利用率降低到 25%, Path3 的带宽利用率提高到 30%。此时, 网络拓扑中所有关键路径的带宽利用率均小于拥塞阈值, 表明在当前网络状态下, 关键路径暂未出现网络拥塞状况。

为了评价所提出的基于深度 Q 网络的拥塞决策机制(CDMB-D3QN)的性能, 本文选择 ECMP 算法、RR 算法和 SPF 算法作为比较机制。在图 6 所示的拓扑结构下, 测试不同 HH 流个数的网络的吞吐量和丢包率(每个 HH 流的传输速率设置为 10 Mbps, 其生存周期为 10s)以及流完成时间。

图 11 展示了网络中有 1~4 条 HH 流时的吞吐量变化, 图 12 显示了丢包率的对比。

吞吐量可以反映网络的传输性能, 吞吐量越大, 表示网络的传输能力越强, 产生拥塞的概率也越小。如图 11 所示, CDMB-D3QN 的吞吐量最高且接近所发送的流量带宽。

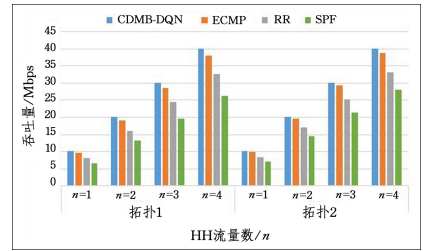


图11 吞吐量对比图

Fig. 11 Throughput comparison

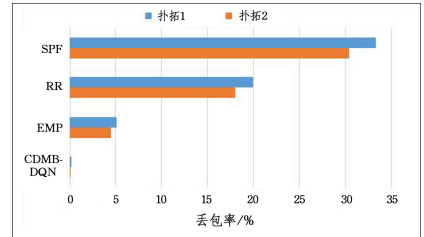


图12 平均丢包率对比图

Fig. 12 Comparison of average packet loss rate

由图 12 可以看出, CDMB-D3QN 的丢包率非常小, 仅为 0.19% 左右, 说明该网络具有良好的传输性能, 能够成功传输所有流量, 可以有效解决网络拥塞, 提高网络性能。

当出现拥塞时, 将当前状态下的流完成时间设置为 1 ms, 平均流完成时间如图 13 所示。可以看出, SPF 与 RR 算法出现的拥塞次数过多, 本文所设计的拥塞决策机制 CDMB-D3QN 平均流处理时间约为 0.584 ms, 与 ECMP 算法的流平均处理时间 0.5813 ms 差距较小。原因可能是在每次处理流时, RL 会根据观察选择动作, 之后需要获取下一次观察与奖励, 因此需要耗费一定的时间。整体来说, CDMB-D3QN 可以较为高效地处理网络拥塞, 提高网络的性能。

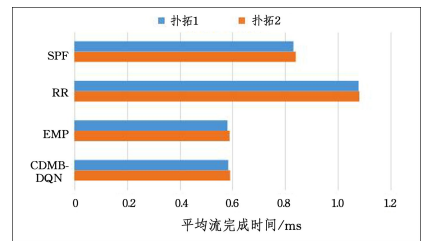


图13 平均流完成时间对比图

Fig. 13 Comparison of average flow completion time

结束语 网络拥塞是互联网长期面临且难以消除的问题。本文提出了一种 KDCN 环境下的拥塞流调机制, 该机制可以有效检测和解决网络中的拥塞, 通过全局网络状态信息准确识别网络中的拥塞, 根据阈值法和知识库两种方法有效识别造成拥塞的 HH 流。另外, 本文将 KDCN 知识定义算力网络与机器学习技术相结合, 解决网络拥塞问题, 从而保证网络的性能。与传统的拥塞流调机制相比, 它具有更强的智能性和前瞻性。

未来的工作包括两个方面: 1) 尝试实现分层分粒度的拥塞流量调度, 使拥塞流调机制能够智能地感知拥塞级别, 针对不同粒度的拥塞采用不同的处理方法; 2) 从丰富知识库的角度出发, 建立更加丰富且结构化的知识库, 进一步提高拥塞检测和决策的效率。

参 考 文 献

- [1] Cisco. White Paper. Cisco Annual Internet Report 2018—2023 [R/OL]. Mar 2020. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- [2] MORANDIN G M. Transmission Control Protocol—TCP[OL]. <https://www.zhangqiaokeyan.com/patent-detail/06130437989464.html>.
- [3] LATIF Z, SHARIF K, LI F, et al. A Comprehensive Survey of Interface Protocols for Software Defined Networks[J]. Journal of Network and Computer Applications, 2020, 156: 102563.
- [4] MOIN S, KARIM A, SAFDAR K, et al. GREEN SDN—An Enhanced Paradigm of SDN: Review, Taxonomy, and Future Directions [J]. Concurrency and Computation: Practice and Experience, 2020, 32(21): e5086.
- [5] ALBERT M, RODRIGUEZ-NATAL A, CARNER J, et al. Knowledge-Defined Networking [J]. CM SIGCOM Computer Communication Review, 2017, 47(3): 2-10.
- [6] CLARK D. A knowledge plane for the internet[C]//Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003: 3-10.
- [7] DALAL S, DAHIYA N, SETH B, et al. Adaptive Traffic Routing Practice for Load Balance and Congestion Control in AdHoc Network in Cloud-MANET [J/OL]. https://www.researchgate.net/publication/353498753_Adaptive_Traffic_Routing_Practice_for_Load_Balance_and_Congestion_Control_in_AdHoc_Network_in_Cloud-MANET.
- [8] ZHANG L, CUI Y, WANG M, et al. DeepCC: Bridging the Gap Between Congestion Control and Applications via Multiobjective Optimization [J]. IEEE/ACM Transactions on Networking, 2022, 30(5): 2274-2288.
- [9] WANG Y, WANG L, DONG X. An Intelligent TCP Congestion Control Method Based on Deep Q Network [J]. Future Internet, 2021, 13(10): 261.
- [10] HM A, DU X A, YD B, et al. An intelligent scheme for congestion control: When active queue management meets deep reinforcement learning [J]. Computer Networks, 2021, 200: 108515.
- [11] ALIZADEH M, EDSALL T, DHARMAPURIKAR S, et al. CONGA: distributed congestion-aware load balancing for data centers [J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 503-514.
- [12] MOHAMMAD A, SIVASANKAR R, BARATH R, et al. Hedera: dynamic flow scheduling for data center networks [C]//Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10). USENIX Association, 2010.
- [13] ZAHER M, ALAWADI A, MOLNAR S. Class-based Flow Scheduling Framework in SDN-based Data Center Networks [C]//International Conference on Computing, Electronics & Communications Engineering (ICCECE), 2020.
- [14] LIU Z, LI K, WU L, et al. CATS: Cost Aware Task Scheduling in Multi-Tier Computing Networks [J]. Journal of Computer Research and Development, 2020, 57(9): 1810-1822.
- [15] AMPRATWUM I. An Intelligent Traffic Classification Based Optimized Routing in SDN-IoT: A Machine Learning Approach [D]. Ottawa: University of Ottawa, 2020.
- [16] MOUFAKIR T, ZHANI M F, GHERBI A, et al. Collaborative Multi-domain Routing in SDN Environments [J]. Journal of Network and Systems Management, 2022, 30(1): 1-23.
- [17] VAN ROSSEMS, TAVERNIER W, SONKOLY B, et al. Deploying elastic routing capability in an SDN/NFV-enabled environment [C]//IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), 2015: 22-24.
- [18] SHI X, LI Q, WANG D, et al. Mobile Computing Force Network (MCFN): Computing and Network Convergence Supporting Integrated Communication Service [C]//International Conference on Service Science (ICSS), 2022: 131-136.
- [19] RUELAS A M R, ROTHENBERG C E. A Load Balancing Method based on Artificial Neural Networks for Knowledge-defined Data Center Networking [C]//IFIP LANC, 2018.
- [20] PHAM Q T A, HADJADJ-AOUL Y, OUTTAGARTS A. Deep Reinforcement Learning Based QoS-Aware Routing in Knowledge-Defined Networking [M]//Social Informatics and Telecommunications Engineering, 2019: 14-26.
- [21] DUQUE-TORRES A, AMEZQUITA-SUAREZ F, RENDON O, et al. An Approach Based on Knowledge-Defined Networking for Identifying Heavy-Hitter Flows in Data Center Networks [J]. Applied Sciences, 2019, 9(22): 1-19.
- [22] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. arXiv:1409.1556, 2014.
- [23] JIAWEI H, MICHELINE K. Data Mining: Concepts and Techniques [J]. Data Mining Concepts Models Methods & Algorithms Second Edition, 2006, 5(4): 1-18.
- [24] JANE M, GREG G. Editorial for Journal of Field Robotics Special Issue on Machine Learning Based Robotics in Unstructured Environments [J]. Journal of Field Robotics, 2006, 23(11/12): 943-944.
- [25] DUAN Y, NI H, ZHU X, et al. A Single-Rate Multicast Congestion Control (SRMCC) Mechanism in Information-Centric Networking [J]. Future Internet, 2022, 14(2): 38.
- [26] LEE C, JUNG J, CHUNG J M. DEFT: Multipath TCP for High Speed Low Latency Communications in 5G Networks [J]. IEEE Transactions on Mobile Computing, 2020, 20(12): 3311-3321.



NIAN Yingpu, born in 2000, postgraduate. His main research interests include next generation Internet, intelligent routing, computing networking.



YI Bo, born in 1988, Ph. D, associate professor, master supervisor, is a member of CCF (No. 34223S). His main research interests include SDN/NFV, routing, service computing and computing networking.