面向SDN的攻击流量分配与负载均衡机制

李曼[1]，周华春[1]，徐琪[2]，邓双兴[1]，邹涛[2]，张汝云[2]
（1.北京交通大学电子信息工程学院，北京 100044；2.之江实验室，浙江 杭州 311121）

摘　要：为了解决软件定义网络（SDN）中传统流量分配算法无法有效识别分布式拒绝服务（DDoS）攻击的问题，提出了一种面向攻击流量的流量分配与负载均衡算法，将流量分配问题建模为马尔可夫决策过程，其中奖励函数考虑了资源消耗和时延。为了优化马尔可夫决策过程，利用基于演员-评论家网络的负载均衡算法，根据流量特征和网络特征，智能分配流量到不同安全路径，以减轻攻击影响，降低负载和时延。实验结果表明，在自生成数据集和公开数据集下，所提算法的奖励值高于对比算法的，表明其在负载均衡方面的性能更优。在吞吐量方面展现出了较高的稳定性，其变化范围相对较小，波动范围为12.95~14.83 Mbit/s；在流量分布方面，所有路径上的流量分布都比较平均；在检测性能方面，识别攻击的平均加权精准率、平均加权召回率和平均加权F1分数分别达到90%、92%和94%。

关键词：SDN；流量分配；负载均衡；DDoS攻击
中图分类号：TN92
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025034

Attack traffic allocation and load balancing mechanism for SDN

LI Man[1], ZHOU Huachun[1], XU Qi[2], DENG Shuangxing[1], ZOU Tao[2], ZHANG Ruyun[2]
1. School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
2. Zhejiang Lab, Hangzhou 311121, China

Abstract: To tackle the problem of traditional traffic allocation methods in software-defined networks (SDN) potentially failing to effectively detect distributed denial of service (DDoS) attacks, a traffic allocation and load balancing mechanism for attack traffic was proposed. The traffic allocation problem was modeled as a Markov decision process (MDP), where the reward function included both resource consumption and delay. To optimize the MDP, a load balancing algorithm based on actor-critic networks was developed. This algorithm allocated traffic to different paths based on traffic and network features with the goal of reducing load and latency. The experimental results demonstrate that, under self-generated and public datasets, the proposed method achieves higher reward than the baseline load balancing methods, indicating its superior performance in load balancing. In terms of throughput, it exhibits high stability with a relatively small variation range, fluctuating between 12.95 Mbit/s and 14.83 Mbit/s. Regarding traffic distribution, the traffic is relatively evenly distributed across all paths. In terms of detection performance, the average weighted precision, average weighted recall, and average weighted F1 score are 90%, 92% and 94%, respectively.

Keywords: SDN, traffic allocation, load balance, DDoS attack

收稿日期：2024–12–27；**修回日期：**2025–02–17
通信作者：徐琪，xuqi@zhejianglab.cn
基金项目：国家重点研发计划基金资助项目（No.2018YFA0701604）；国家自然科学基金资助项目（No.62341102）；浙江省重点研发计划基金资助项目（No.2024SSYS0001）；山东省自然科学基金资助项目（No.ZR2023LZH017）
Foundation Items: The National Key Research and Development Program of China (No.2018YFA0701604), The National Natural Science Foundation of China (No.62341102), The National Key Research and Development Program of Zhejiang Province (No.2024SSYS0001), The National Natural Science Foundation of Shandong Province (No.ZR2023LZH017)

0　引言

随着新型网络的迅速发展，新应用不断涌现，智能化程度日益提高，同时也带来了严重的安全威胁[1]。尤其是在软件定义网络（SDN, software defined network）面临网络攻击的环境下，如何实现有效的负载均衡和流量分配成为挑战。良好的负载均衡策略可以直接缓解网络拥塞，提高网络性能，确保业务流的高效传输。通过优化流量分配策略，可以平衡网络负载，降低网络时延，从而提升用户体验和服务质量。此外，构建安全可靠的流量分配机制，还可以有效应对网络攻击，保障网络的安全运行。

在海量攻击和SDN场景下进行流量分配和负载均衡面临着众多挑战。第一，分布式拒绝服务（DDoS, distributed denial of service）攻击的多样性和复杂性要求网络能够快速识别和响应，以减轻攻击带来的影响。同时，SDN环境中的流量调度需要考虑到网络的动态变化和不确定性，以确保在面对大规模攻击时，网络仍能维持高效和稳定的运行。第二，当网络遭受大量突发性的攻击时，传统的流量调度算法可能无法有效应对，因为它们通常基于静态的网络模型，难以适应快速变化的网络状态。例如，基于开放式最短路径优先（OSPF, open shortest path first）路由协议的传统算法可能在攻击流量激增时导致转发时延长和链路拥塞，从而无法满足安全性能的需求[2]。

此外，尽管一些研究提出了基于负载均衡的路由算法，但这些算法在处理海量业务流时计算效率低下，难以在大规模网络中实现理想的流量调度。机器学习技术的发展为解决这些问题提供了新的思路。例如，文献[3-5]利用深度神经网络和强化学习模型来预测网络流量并调整路由配置，以实现负载均衡。然而，这些算法在网络动态变化和攻击模式不断演变的情况下，可能面临训练时间长、模型难以适应等问题。因此，在海量攻击和SDN场景下，研究一种能够快速适应网络变化、有效抵御攻击、并实现高效流量分配和负载均衡的算法具有重要意义。

本文提出了一种面向攻击流量的流量分配与负载均衡（AALB, attack-oriented traffic allocation and load balancing）算法。所提算法利用深度强化学习的强大决策能力，动态调整网络流量路径，以实现负载均衡和优化网络性能。首先，针对时延、带宽和计算资源等多维度进行系统建模，综合考虑这些因素，形成一个以最小化资源消耗和时延为目标的负载均衡优化问题。把负载均衡优化问题转化为马尔可夫决策过程（MDP, Markov decision process），进一步，提出了一种基于演员-评论家网络的流量分配和负载均衡算法。该算法利用演员-评论家网络的强大学习能力，动态调整网络流量路径，流量路径由网络服务功能（NSF, network service function）组成，也称为服务功能链（SFC, service function chain）。把流级别的流量智能分配到不同的安全路径，确保每条路径上的流量分布均衡，避免某条路径过载而其他路径资源闲置，减轻攻击带来的影响，降低路径的负载和时延，同时保障安全性能。最后，在自生成数据集和公开数据集下，与基准负载均衡算法对比了负载均衡能力（LBC, load balancing capability）、吞吐量和流量分布。此外，展示了所提算法检测攻击的归一化混淆矩阵、平均加权精准率、平均加权召回率和平均加权F1分数。

1　相关工作

本节总结了负载均衡算法的研究现状，主要从SDN中负载均衡算法和各种负载均衡算法2个维度进行了深入探讨。

1.1　SDN中负载均衡的研究现状

不同的网络环境对流量分配有着不同的要求。负载均衡技术旨在优化网络资源的使用，提高网络的吞吐量和响应速度，同时降低时延和丢包率，确保用户体验和网络的稳定性。表1为SDN中负载均衡算法的评价指标以及优缺点。

文献[6]提出一种动态优先级多路径调度算法，实现了动态调度大象流和老鼠流。通过与基准算法相比，实验证明了所提算法能提高网络吞吐量和链路利用率，减少了平均流完成时间。该文献中重点关注大象流和老鼠流的调度性能，未验证攻击流量的适配性。

文献[7]介绍了一种基于强化学习（RL, reinforcement learning）的路由算法。该文献所提算法伸展性比对比算法小14%~16%、平均链路时延比对比算法平均低8%~50%、平均丢包率比Dijkstra算法的4种变体的丢包率平均低30%，最多低65%。链路吞吐量比对比算法平均低4%~27%。该

表1 SDN中负载均衡算法的评价指标以及优缺点

文献	算法	评价指标	优点	缺点
文献[6]	动态优先级多路径调度算法	链路利用率 网络吞吐量	适用数据中心网络	增加了网络负载
文献[7]	基于强化学习的路由算法	伸展性、链路吞吐量、平均丢包率和平均链路时延	保证低时延和高带宽利用率	侧重解决SDN中路由的问题
文献[8]	基于RL和深度神经网络的负载平衡路由算法	平均时延、平均利用率	提高了网络平均利用率和时延性能	缺乏验证安全性能
文献[9]	精英遗传算法	负载均衡程度	有效地平衡控制平面的负载	控制器资源利用率不足
文献[10]	优先级请求算法	总传输包数、仿真时间、服务器数量、时延、吞吐量、带宽	提高网络性能和改善用户体验	只适用于数据中心、云计算和物联网环境
文献[11]	混沌逻辑微分进化算法	能量消耗、网络生存能力	改善控制器的负载平衡和可靠性	只关注控制器的负载均衡问题
文献[12]	基于深度强化学习和贝叶斯网络的资源管理算法	处理时延负载均衡率	实现最优的负载平衡策略	缺乏在海量攻击流量下验证算法的性能
文献[13]	基于遗传算法和蚁群优化负载均衡的能量感知路由机制	响应时间和能量消耗	具有较强的负载均衡能力	
文献[14]	自适应可靠粒子群优化的负载均衡算法	时延、丢包率、吞吐量、平均往返时间和带宽利用率	解决SDN单控制器部署中的负载均衡问题	包数量不断增加时，会发生拥塞

文献所提算法侧重解决SDN中路由问题，未考虑负载均衡。

文献[8]提出了一种基于RL和深度神经网络的负载平衡路由算法，该算法可以利用SDN控制器的全局视图来制定路径策略，结果表示当负载超过80%时，该文献所提算法比传统算法的平均时延降低50~100 ms。该文献所提算法的网络平均利用率高达90%。文献[8]是把来自不同源的所有流量作为一个整体来考虑网络资源，但是来自不同源的攻击流量特征有差异，与文献[8]不同的是，本文研究综合网络状态和流量状态，考虑每个路径的流量分布、负载均衡性能，把攻击流量分配到合适的路径上。

文献[9]提出了一种精英遗传算法，旨在解决交换机与控制器之间映射关系导致的负载不均衡问题。实验结果表明，精英遗传算法能快速找到最优的部署方案，有效地平衡控制平面的负载。由于控制器存在资源利用率不足的情况，与文献[9]不同的是，本文更关注平衡数据平面中节点负载。

文献[10]提出了一种优先级请求算法，采用多个控制器来实现资源分配，使用优先级请求算法以确保关键请求获得更高的优先级，同时使用马尔可夫过程模型来最大化利用可用资源。这种算法只适用于数据中心、云计算和物联网环境，不适合SDN。与文献[10]不同的是，本文在SDN中用马尔可夫过程建模流量和网络环境解决负载均衡问题。

文献[11]提出了一种混沌逻辑微分进化算法。该算法分为确定控制器数量、控制器位置优化和贪婪算法的应用。实验结果表明，与基准算法相比，混沌逻辑微分进化算法的能量消耗降低了16%~27%、网络生存能力提高了57%~136%。但是文献[11]只关注控制器的负载均衡问题，通过优化控制器数量和位置来提升网络性能，面对海量攻击时，攻击流量会增加数据平面中节点负载，可能导致控制器与节点之间无法正常通信，因此，本文着重解决数据平面中节点负载的问题。

文献[12]提出了一种基于深度强化学习和贝叶斯网络的资源管理算法。实验结果表明，该算法能够有效地解决动态负载环境下的资源分配问题，提高网络性能和效率。文献[12]侧重解决资源管理的负载均衡问题。

文献[13]提出采用基于遗传算法和蚁群优化负载均衡的能量感知路由机制。旨在最小化能量消耗，同时保持用户流量的服务质量，并实现链路负载均衡。仿真结果表明，所提机制的响应时间比对比算法少3 ms，能量消耗比对比算法低1 J。但是，缺乏在海量攻击流量下验证算法的性能。

文献[14]提出了一种自适应可靠粒子群优化的负载均衡算法，以解决SDN单控制器部署中的负载均衡问题。仿真结果表明，所提算法在时延方面比传统算法低6%~25%、丢包率降低11%~57%、吞吐量增加2%~12%、平均往返时间降低4%~20%，带宽利用率降低6%~22%。然而，当发送的包数量不断增加时，时延减小趋势会变缓慢，这种减少的原因是链路发生了拥塞。与文献[14]不同的是，本文把流量分配到不同的路径中进行检测和分类，降低单一路径的负载压力，确保每条路径能够提供安全服务。

文献[6-14]主要解决SDN中流量调度、路由和负载均衡问题。尽管这些研究主要关注正常网络环境下的性能优化，并未深入探讨在攻击场景下的负载均衡机制。在攻击场景下，数据平面中的节点负载会急剧增加，可能导致控制器与节点之间的通信受阻，甚至使整个网络陷入瘫痪状态。因此，如何在保证检测性能的同时提升网络性能，成为一个亟待解决的问题。

1.2　各种负载均衡算法研究现状

为了提高资源利用率、降低时延、增加吞吐量并保证网络稳定性，研究者们积极探索并应用启发式和机器学习算法来解决负载均衡问题。表2总结了各种负载均衡算法的评价指标以及优缺点。

文献[15]提出了一种时延感知加权等价路由算法，旨在针对给定的流量需求实现负载均衡。仿真评估表明，时延感知加权等价路由可以实现比传统等价路由更好的负载均衡，并有效降低网络中的整体端到端时延3 ms左右。文献[15]中把流量分割到多条路径中，这种方式能实现负载均衡，但会增加资源的消耗。与文献[15]不同的是，本文把完整的流量直接分配到不同的路径，而不是对流量进行分割，既能减少资源的消耗，也能保证负载均衡性。

文献[16]提出了一种深度优先搜索算法。实验结果表明，与单路径路由相比，该算法在时延方面与对比算法相比，降低20%~52%，抖动降低40%~43%，丢包率降低39%~58%。此外，该算法还允许将传入流量分配到所有可用路径上，从而实现更好的负载分布、资源利用率优化以及路径故障管理，然而，缺乏验证在攻击环境中性能。

文献[17]提出了一种多目标优化调度算法，将

表2　　各种负载均衡算法的评价指标以及优缺点

文献	算法	评价指标	优点	缺点
文献[15]	时延感知加权等价路由算法	端到端时延	降低时延	增加资源的消耗
文献[16]	深度优先搜索算法	端到端时延、抖动、丢包率和资源利用率	有效分配网络负载	缺乏验证在攻击环境中性能
文献[17]	多目标优化调度算法	完成时间、截止日期违规率、平均CPU利用率	实现了多目标优化	导致网络带宽碎片化
文献[18]	加权成本多路径算法的网络流量均衡分配算法	累积分布函数	更好的负载均衡效果和更高的带宽利用率	增加计算复杂度
文献[19]	基于改进蚁群的负载均衡算法	负载均衡	更好的负载均衡效果	预定义的规则
文献[3]	基于监督学习的路由算法	移动平均窗口	有效缓解网络拥塞	难以适应动态的流量情况
文献[20]	基于深度强化学习的流量工程算法	端到端时延、网络利用率	平衡端到端时延和网络利用率	缺乏评估多路径的性能
文献[21]	基于深度强化学习和图神经网络的路由优化方案	最大链路利用率、端到端时延和网络效用	具有较好的鲁棒性和灵活性	缺乏在实际网络环境下验证可行性
文献[22]	基于RL的网络流量管理算法	端到端时延、平均链路利用率、负载均衡性能	提供更好的时延性能和接近最优的负载平衡性能	没有评估安全性能
文献[23]	深度确定性策略梯度算法	网络吞吐量、平均时延	提高网络性能	未考虑多条路径负载均衡的问题
文献[24]	基于多智能体强化学习和图神经网络的流量粒度路由优化方案	平均时延和平均丢包率	优化网络性能	复杂度高，不适用资源受限的SDN环境
文献[25]	基于神经网络的负载均衡路由算法	平均丢包率、最坏吞吐量和平均时延	有效地分配网络资源，提高网络性能	不适用解决攻击流量下负载均衡的问题

流完成时间和用户预算成本作为优化问题的约束，实现性能和成本的多目标优化。实验结果表明违规率降低了34%，完成时间提高了56.6%，平均CPU利用率提高了4%。但是蚁群优化算法容易陷入局部最优问题，与文献[17]不同的是，本文提出的算法能够根据实际网络状态和攻击流量需求动态调整路径决策，寻找全局最优解。

文献[18]提出了加权成本多路径算法的网络流量均衡分配算法。该算法适合在网络拓扑变化的情况下平衡流量。与文献[18]不同的是，本文在固定的网络拓扑下，提出负载均衡算法，根据流量特征和网络特征，智能分配流量到不同路径中。

文献[19]提出了一种基于改进蚁群的负载均衡算法（IACA, load balancing algorithm based on improved ant colony），致力于在路径长度、负载以及处理效率的多约束条件下为待调度流找到最佳传输路径。仿真结果表明，在负载均衡能力方面，该算法优于其他传统算法。但当流量请求时间较长时，可能会导致服务器负载不平衡以及单台服务器过载的问题。与文献[19]不同的是，本文提出的算法平衡了每个服务器节点的负载。

文献[15-19]采用的启发式算法是基于一系列预定义的规则和策略，这些规则和策略往往是根据特定场景设计的。然而，在实际应用中，网络环境可能复杂多变，固定的启发式规则可能无法适应所有的负载变化，导致负载分配不准确或不均衡。启发式算法在处理动态负载变化时可能表现出一定的滞后性。它们是基于静态负载信息进行决策，当网络环境发生快速变化时，启发式算法可能无法及时做出调整，导致负载分配不及时或不合理。鉴于启发式算法在解决负载均衡问题时的不足，研究者们开始探索使用机器学习来解决负载均衡问题。

文献[3]提出了基于深度神经网络的路由（DNNR, deep neural network for routing）算法，将流量特征和链路利用率作为深度神经网络的输入，并为每条链路输出路径。实验证明该算法能够有效缓解网络拥塞。与文献[3]不同的是，本文通过深度强化学习以及环境交互和探索的方式，不断学习变化的流量特征，不断更新模型来实现负载均衡。

文献[20]提出基于深度强化学习的流量工程（DRL-TE, traffic engineering based on deep reinforcement learning）算法解决网络中的流量调度问题。该算法采用深度强化学习对网络中多个会话流量进行调度以降低端到端时延。仿真结果表明DRL-TE算法相比于其他负载均衡算法端到端时延降低了44.2%~70.5%，网络利用率提高了7.7%~26.4%。文献[20]中在代表性和随机生成的网络拓扑下验证了调度效果。与文献[21]不同的是，本文在固定拓扑下，设计了一种负载均衡算法，不仅实现了节点负载的平衡，还确保了流量的安全性。

文献[21]提出了一种基于深度强化学习和图神经网络的路由优化方案。实验结果表明相比于基准算法的最大链路利用率降低8.61%~35.98%，端到端时延下降8.72%~48.5%，网络效用提高0.33%~4.15%。然而，本文献的关注点在于调整路由策略，但并未考虑将路由分配到多个路径上以实现负载均衡。

文献[22]提出了一种基于RL的网络流量管理算法。该算法可以自动学习如何分配网络流量以满足不同优先级的需求，并优化网络性能。与传统的基于规则的算法相比，基于RL的网络流量管理算法可以根据实时的流量需求和网络拓扑动态地调整转发规则，从而更好地适应不断变化的网络环境，但是，缺乏验证在海量攻击流量下的性能。

文献[23]提出了一种智能驱动体验网络架构，通过深度确定性策略梯度（DDPG, deep deterministic policy gradient）算法来模拟人类的学习经验，并利用闭环网络控制机制和网络监控技术实现与网络环境的交互。该架构可以自动调整服务和资源以优化网络性能，提高网络吞吐量并降低平均时延。实验结果表明，相对于传统的方案，智能驱动体验网络架构显著提高了网络性能。文献[23]保证了流量转发到最优路径，在实际网络环境中，流量会经过多条路径提供服务，其没有详细讨论如何处理多条路径的负载均衡问题。

文献[24]提出了一个基于多智能体强化学习和图神经网络的流量粒度路由优化方案来提高端到端质量。实验结果表明，该方案使用强化学习优化路由的思路为解决负载均衡问题提供了一种可参考的解决方案。

文献[25]提出了基于神经网络的负载均衡路由算法，旨在解决大规模数据包交换中的负载不均衡问题。在平均丢包率、最坏吞吐量和平均时延等方面表现更优。在攻击流量场景中，流量特征和网络

状态随着攻击强度变化，文献[25]使用的单一攻击流量状态信息存在局限性，无法全面反映攻击环境的复杂性和多样性。相比之下，本文算法结合网络状态和流量状态来代表攻击环境，更全面地反映攻击环境的复杂性和多样性。

大规模攻击流量的随机性和动态性，导致网络状态难以被准确描述。此外，网络规模、网络资源以及流量资源造成状态与决策空间维度较高，进一步加剧了算法的收敛难度[26]。近年来，基于“演员-评论家”结构的DRL算法[27]，在求解具有高纬或连续性决策空间的优化问题方面取得了很多应用[28]。借助机器学习可以通过与环境的交互来学习和适应负载变化的优势，本文利用演员-评论家网络解决DDoS场景下流量分配和负载均衡问题。首先，针对时延、资源开销进行系统建模，并形成以最小化资源开销和时延的差为目标的负载均衡优化问题。其次，为简化问题求解难度，将所形成的优化问题转化为马尔可夫决策过程，提出了一种AALB算法，利用演员-评论家算法把流量分配到每条路径中。最后，与文献[3,19-20]对比了所提算法的性能。

2　应用场景和问题建模

本文首先描述了流量分配和负载均衡的应用场景，其次，为了清晰地阐述模型和算法，列举了本节所使用的主要数字符号和变量，然后详细介绍了系统模型、明确了优化目标和约束条件，最后，为了求解优化问题，引入MDP进行建模。

2.1　应用场景

图1为流量分配描述示例。具体地，将多条流量进行编排，依据流量分配和负载均衡策略，以及网络状态和流量状态的实时状况，智能地将流量分散至多个传输路径（如SFC_1和SFC_2）进行并行传输和检测，减少单一节点和单一SFC的负载，优化流量调度，提高网络资源可用性和整体的安全性。

特别地，每个SFC中都包含入口分类器、出口分类器和5个检测模块，每个检测模块识别分类一大

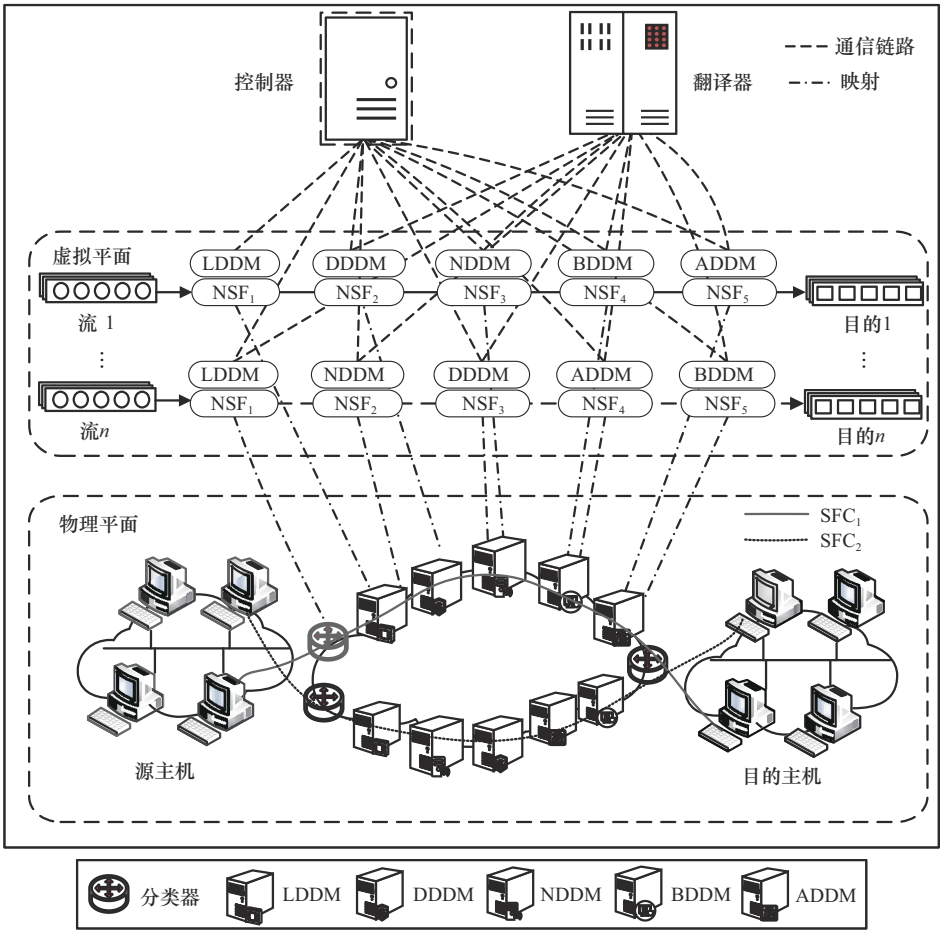


图1　流量分配描述示例

类攻击。具体包括低速率拒绝服务检测模块（LDDM, LDoS detection module）、反射放大拒绝服务检测模块（DDDM, DRDoS detection module）、网络层分布式拒绝服务检测模块（NDDM, network DDoS detection module）、僵尸网络检测模块（BDDM, botnet detection module）、应用层分布式拒绝服务检测模块（ADDM, application DDoS detection module）。这些模块分别针对不同类型的攻击，如低速率拒绝服务（LDoS, low-rate denial of service）攻击、反射放大拒绝服务（DRDoS, distributed reflection denial of service）攻击、网络层攻击、僵尸网络和应用层攻击，提供了精细化的分类和检测。

为了便于理解流量分配优化负载均衡的优势，进一步详细阐述SDN中流量分配和负载均衡机制。图2展示了流量分配优化适配机理，包括知识平面、控制平面、物理平面和虚拟平面。

控制平面能实现状态感知机制以及策略下发机制。状态感知机制通过策略控制器实时感知网络和流量状态信息（时延、资源消耗、流持续时间、流速、正向子流中的数据包平均数量）并将其作为知识平面的输入。

虚拟平面是节点资源以及链路资源的集合，并负责数据包的转发和检测。节点包括入口分类器、出口分类器、LDDM、BDDM、NDDM、DDDM、

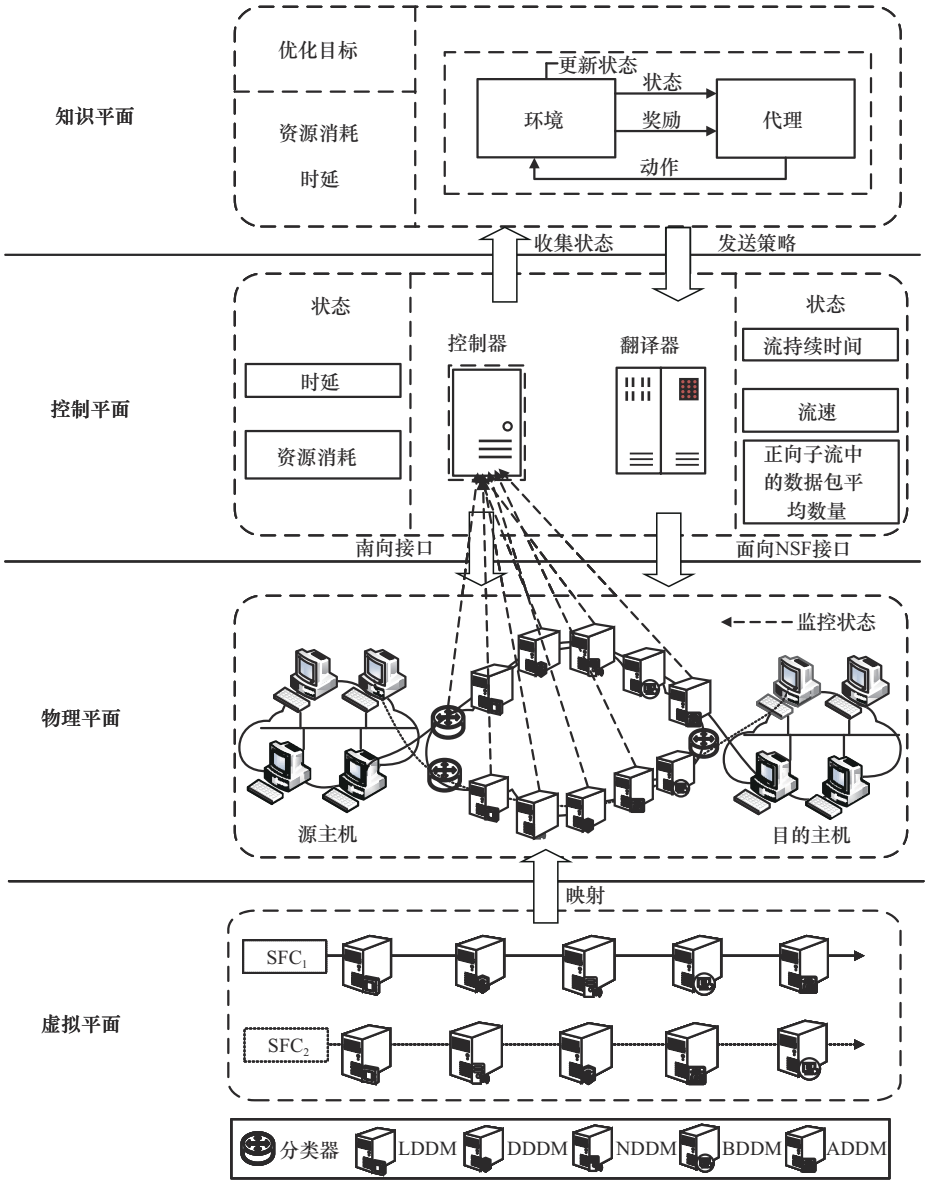


图2　流量分配优化适配机理

ADDM策略下发机制通过翻译器将流量分配的高级策略转译为详细的低级策略，通过面向NSF接口，下发低级策略中配置命令，传递给物理平面中各个网络组件，由其执行相应决策，实现相应的服务功能，比如特征提取和识别分类。低级策略中路径信息传递给控制器，由其负责转译成流表并下发。流表通过控制器下发至物理网络组件，保证安全功能之间的正常通信以及提供检测服务。当源主机发起攻击流量时，根据流表中所选的安全功能和路径索引，流量被分配到不同的路径中进行检测和分类，降低单一路径的负载压力。

知识平面中智能体利用演员-评论家网络深度分析网络状态和流量状态，制定精确的流量分配高级策略。当检测到某条路径可能面临过载风险时，迅速调整策略，将部分流量转移到其他路径上，以平衡负载并确保安全服务的连续性。

总之，在攻击流量变化时，知识平面发挥着重要的作用。它根据反馈的信息优化负载均衡模型，以确保网络流量分配到不同的路径中检测和分类，从而避免个别路径的资源浪费和过载风险。

本文重点围绕图2中知识平面优化流量分配、控制平面的状态感知和策略下发的功能，设计一种流量分配和负载均衡算法，该算法利用演员-评论家网络把流量分配到不同的路径中进行检测和分类，降低单一路径的负载压力，确保每条路径能够提供安全服务，提高网络性能的同时保证安全性能。

2.2　系统参数

在介绍负载均衡算法之前，本节先介绍系统模型和MDP建模过程，表3为本节所使用的主要数字符号和变量。

2.3　系统模型

底层网络模型：底层基础设施建模用有向加权图 $G=(N,L)$ 表示，$N=\{1,2,\cdots,\hat{N}\}$ 是物理网络节点的集合。物理网络节点N进一步分为3个不相交子集 $N=N_{\mathrm{c}}\cup N_{\mathrm{u}}\cup N_{\mathrm{p}}$，其中，$N_{\mathrm{c}}$表示入口分类器物理节点集合，$N_{\mathrm{u}}$表示服务功能物理节点集合，$N_{\mathrm{p}}$表示出口分类器物理节点集合。$L=(1,2,\cdots,L)$ 是物理链路的集合，其中，所有物理链路均是双向通信链路，即 $l_u^n(u,n\in N)$ 表示物理节点u到n的通信链路，l_n^u表示物理节点n到u的通信链路。链路 $l\in L$ 对应的总带宽资源为B_l。

表3　主要数字符号和变量

变量	含义
N	物理节点的集合
N_{c}	入口分类器物理节点的集合
N_{u}	服务功能物理节点的集合
N_{p}	出口分类器物理节点的集合
E	物理链路集合
l_u^n	物理节点u和n之间的通信链路
l_n^u	物理节点n和u之间的通信链路
B_l	链路l的带宽资源
$\tilde{N}$	虚拟节点的集合
$\tilde{N}_{\mathrm{c}}$	入口分类器虚拟节点的集合
$\tilde{N}_{\mathrm{u}}$	服务功能虚拟节点的集合
$\tilde{N}_{\mathrm{p}}$	出口分类器虚拟节点的集合
$\tilde{L}$	虚拟链路的集合
$\tilde{l}_{u'}^{n'}$	虚拟节点u'和n'之间的通信链路
C	SFC集合
c_i	第i条SFC
g	SFC的数量
v	一条SFC中服务功能的数量
r	一条SFC中虚拟节点的数量
C_n^m	虚拟节点的总计算资源
C_n^d	虚拟节点的总检测资源
C_n^f	虚拟节点的总转发资源
$f(\tilde{n}_u)$	服务功能的检测成本
f_{s}	流量的源节点
f_{d}	流量的目的节点
f_{v}	流量速率
f_{w}	正向子流中的数据包平均数量
f_{r}	流持续时间
σ_u	物理节点被启动和配置消耗的总物理资源开销
σ_o	流量检测过程中管理和维护所有物理节点需要的物理资源开销
$\bar{C}_n^m$	虚拟节点消耗的计算资源
$\bar{C}_n^d$	虚拟节点消耗的检测资源
$\bar{C}_n^f$	虚拟节点消耗的转发资源
$C(l_u^n)$	物理节点u和n之间的链路开销
W_{c}	资源消耗
D_{f}	时延成本
D_{t}	传输时延
D_{g}	传播时延
D_{q}	排队时延
D_{e}	检测时延
s_t	状态
a_t	动作
r_t	奖励

安全服务模型：虚拟节点集合定义为一个有向带权图$\tilde{G}=(\tilde{N},\tilde{L})$，其中$\tilde{N}$表示虚拟节点集合。$\tilde{N}_{\mathrm{c}}\in\tilde{N}$表示入口分类器虚拟节点集合，$\tilde{N}_{\mathrm{p}}\in\tilde{N}$表示出口分类器虚拟节点集合，$\tilde{N}_{\mathrm{u}}\in\tilde{N}$表示NSF集合。$\tilde{L}$表示虚拟链路集合，用$\tilde{l}_{u'}^{n'}(u',n'\in\tilde{N})$表示虚拟节点$u'$到$n'$的通信链路。假设每个SFC由入口分类器节点、出口分类器节点和5种NSF组成，由C表示SFC集合，集合中SFC的数量有g个。每个SFC$c_i\in C$中有v个服务功能、一个入口分类器节点和一个出口分类器节点，由$c_i=(\tilde{n}_{\mathrm{c}},\tilde{n}_1,\tilde{n}_2,\cdots,\tilde{n}_v,\tilde{n}_{\mathrm{p}})$表示，其中每个节点都具有总计算资源$C_n^m$和总转发资源$C_n^f$，但是服务功能除了有计算和转发资源，还有总检测资源C_n^d。

流量模型：每条流量$f\in F$可以由一个五元组$f=\{f_{\mathrm{s}},f_{\mathrm{d}},f_{\mathrm{v}},f_{\mathrm{w}},f_{\mathrm{r}}\}$表示，其中$f_{\mathrm{s}}$和$f_{\mathrm{d}}$分别为流量的源节点和目的节点，$f_{\mathrm{v}}$表示流量速率，即每秒的数据包数量，$f_{\mathrm{w}}$表示正向子流中的数据包平均数量，$f_{\mathrm{r}}$表示流持续时间。

2.4　优化目标和约束条件

资源消耗W_{c}包括物理资源开销、虚拟节点资源开销和链路资源开销。在物理资源开销方面，流量检测前，所有路径中物理节点需要启动且配置，则其产生物理资源开销。在虚拟节点资源开销方面，当流量经过每条SFC时，SFC是由服务功能、入口分类器节点和出口分类器节点组成。其中每个虚拟节点会消耗相应的计算资源$\bar{C}_n^m$和转发资源$\bar{C}_n^f$，服务功能节点会消耗检测资源$\bar{C}_n^d$。在链路资源开销方面，当SFC中服务功能执行检测服务时，流量经过SFC中物理链路产生的链路资源开销，$C(l_u^n)$表示流量所占用物理节点u到物理节点n之间的链路资源开销。假设流量被分配到i条SFC中，每条SFC中有r个虚拟节点，其中i条SFC对应的总资源消耗公式化为

$$W_{\mathrm{c}}=(\sigma_{\mathrm{u}}+\sigma_{\mathrm{o}})+\sum_{c_1}^{c_i}\sum_{r=1}^{r=r}x_{\tilde{n}}^{n}(\bar{C}_n^m+\bar{C}_n^d+\bar{C}_n^f)+\sum_{c_1}^{c_i}\sum_{e'}y_{\tilde{u}\tilde{n}}^{un}C(l_u^n)\tag{1}$$

其中，σ_{u}表示所有物理节点被启动和配置消耗的总物理资源开销，σ_{o}表示在流量检测过程中管理和维护所有物理节点需要的物理资源开销，只要物理节点执行转发和检测功能就会产生该部分开销，与流经节点的流量大小无关，e'是一条SFC中物理链路总数，二进制变量$x_{\tilde{n}}^{n}=1$表示SFC所需的虚拟节点$\tilde{n}$部署在物理节点$n\in N$，$y_{\tilde{u}\tilde{n}}^{un}=1$表示虚拟链路$\tilde{u}\tilde{n}\in\tilde{L}$流经物理链路$un\in L$。

对于SFC$c_i\in C$，端到端时延包括传输时延D_{t}、传播时延D_{g}、排队时延成本D_{q}和检测时延D_{e}，第i条SFC对应的时延成本D_{f}数学表达式为

$$D_{\mathrm{f}}=\sum_{c_1}^{c_i}(D_{\mathrm{t}}+D_{\mathrm{g}}+D_{\mathrm{q}}+D_{\mathrm{e}})\tag{2}$$

其中，传输时延D_{t}是发送数据包大小与链路带宽的比值，计算式为$D_{\mathrm{t}}=\frac{f_z}{B_l}$。传播时延$D_{\mathrm{g}}$是链路物理长度$H$与信号在该物理链路介质中的传播速度$\upsilon$的比值，计算公式为$D_{\mathrm{g}}=\frac{H}{\upsilon}$，但是这个值很小，本文忽略不计。依据$M/M/1$排队模型[29]，排队时延$D_{\mathrm{q}}$取决于SFC的平均负载$\rho$，计算式为$D_{\mathrm{q}}=\frac{1}{\mu-f_v}\times\frac{\rho}{1-\rho}$。其中，$\mu$是服务率，服务率是服务功能完成检测的平均速率。检测时延D_{e}是指单位时间内每个数据流的平均检测时间，取决于每个服务功能的处理能力，计算式为$D_{\mathrm{e}}=\frac{1}{\bar{C}_n^d-\zeta}$。其中，$\zeta$是待检测的数据流的数量。

优化目标如式(3)所示，定义为时延和资源消耗的总和。

$$\begin{aligned}\mathrm{cost}=\min(W_{\mathrm{c}}-D_{\mathrm{f}})=\min[&(\sigma_{\mathrm{u}}+\sigma_{\mathrm{o}})+\\&\sum_{c_1}^{c_i}\sum_{r=1}^{r=r}x_{\tilde{n}}^{n}(\bar{C}_n^m+\bar{C}_n^d+\bar{C}_n^f)+\\&\sum_{c_1}^{c_i}\sum_{e'}y_{\tilde{u}\tilde{n}}^{un}C(l_u^n)-\sum_{c_1}^{c_i}(D_{\mathrm{t}}+D_{\mathrm{g}}+D_{\mathrm{q}}+D_{\mathrm{e}})]\end{aligned}\tag{3}$$

资源约束分为节点资源约束和链路带宽资源约束。节点资源约束保证SFC中安全功能所需的计算资源$\bar{C}_n^m$、转发资源$\bar{C}_n^f$和检测资源$\bar{C}_n^d$不应超过该节点的总计算资源C_n^m、总检测资源C_n^d和总转发资源C_n^f。链路带宽资源约束保证SFC所需的带宽资源不应超过流经链路l_n^u的可用带宽资源为B_l，如式(4)~式(7)所示。

$$\bar{C}_n^m\leqslant C_n^m,\forall n\in N\tag{4}$$

$$\bar{C}_n^d\leqslant C_n^d,\forall n\in N\tag{5}$$

$$\bar{C}_n^f\leqslant C_n^f,\forall n\in N\tag{6}$$

$$\sum_{\tilde{u}\tilde{n}\in\tilde{N}} y_{\tilde{u}\tilde{n}}^{un} C(l_u^n) \leqslant B_l, \forall l_u^n \in L \tag{7}$$

对于任意一条SFC$c_i \in C$，由式(8)可知，其实际经历的端到端时延不应超过该SFC的最大容忍时延$D_{c_i}^-$。

$$D_{\mathrm{t}} + D_{\mathrm{g}} + D_{\mathrm{q}} + D_{\mathrm{e}} \leqslant D_{c_i}^-, \forall c_i \in C \tag{8}$$

2.5　MDP建模

MDP是一种时序决策数据模型，用于在系统状态具有马尔可夫性质的环境中模拟智能体的随机策略与奖励。它可用于优化智能体的随机策略，并通过奖励机制来引导智能体的学习与决策。马尔可夫性质是指在一个随机过程中在给定当前状态的情况下，未来状态的概率分布只依赖于当前状态，与过去状态无关。当网络面临海量攻击时，为了获得随机网络环境下的实时最优决策，首先将流量分配问题转化为MDP，并提出基于演员-评论家网络的流量分配和负载均衡算法，以高效求解转化后的MDP[30]。通常，MDP包括状态、动作、策略和奖励等基本要素。在图2中，智能体被部署在知识平面中，可以为驱动系统运行作出实时负载均衡决策；除此以外的部分被视为MDP的环境。基于此，将智能体观察到的网络状态、制定的负载均衡决策以及获得的系统奖励等要素定义如下。

1) 网络状态s_t：在每个运行时隙中，数据平面中每条SFC中服务功能、入口分类器和出口分类器周期性地向控制器发送网络状态（包括时延成本D_{f}、资源消耗W_{c}）以及流量状态（流持续时间f_{r}、流量速率f_{v}和正向子流中的数据包平均数量f_{w}）。控制器通过应用程序接口（API）发送到知识平面。因此，知识平面中智能体可以建立相应网络状态空间S。在运行时隙t中，可将网络状态定义为

$$s_t = \{W_{\mathrm{c}}(t), D_{\mathrm{f}}(t), f_{\mathrm{v}}(t), f_{\mathrm{w}}(t), f_{\mathrm{r}}(t)\} \tag{9}$$

其中，$s_t \in S$表示当前网络状态集合。

2)负载均衡决策a_t：智能体根据策略π和状态s_t生成的负载均衡方案，流量被分配到合适的SFC，负载均衡决策表示策略π和状态s_t的函数，如式(10)所示。$a_t = \left(c_1, c_2, \cdots, c_g\right)$表示动作集合，集合中有$g$条SFC。在一个轮次中，代理会基于当前状态从动作集合中选取多个SFC，如果选择的多条SFC能满足约束条件（式(3)~式(8)）则该轮次终止。

$$a_t = \pi(s_t) \tag{10}$$

3) 奖励r_t：奖励是指环境针对智能体的负载均衡决策a_t做出反馈，用于评价该决策的有效性，根据获得的奖励对策略π进行迭代，直到获得最优决策能力。为了最小化时延和资源消耗之间的差，将运行时隙t中的奖励函数定义为

$$r_t = \begin{cases} W_{\mathrm{c}}(t) - D_{\mathrm{f}}(t), \text{约束条件能被同时满足} \\ 0, \text{约束条件不能被同时满足} \end{cases} \tag{11}$$

其中，$W_{\mathrm{c}}(t) - D_{\mathrm{f}}(t)$表示SFC在运行时隙$t$中产生的资源消耗和时延之差，资源消耗和时延的单位不同，为了确保奖励的合理性，将资源消耗和时延的值归一化到相同的范围。奖励函数包含2种情况：当所选择的负载均衡决策能够满足带宽和时延约束时，智能体可以获得系统奖励$r_t = W_{\mathrm{c}}(t) - D_{\mathrm{f}}(t)$；否则，则表示选择的SFC不满足时延约束和带宽约束，需要对智能体进行相应惩罚以指导其朝更好的方向优化，此时$r_t = 0$。

3　AALB算法

根据式(3)可知，所形成的流量分配优化问题为混合整数非线性规划问题。在该问题中，由于流量分配与预训练模型之间存在位置耦合关系，需要综合考虑网络资源和流量情况，采用传统优化算法难以求解该类型的问题。因此，本节提出了一种基于深度强化学习的算法来实现流量分配和负载均衡，从而获得最优的求解策略。具体而言，将所形成的优化问题转化为MDP过程，然后提出一种面向攻击流量的流量分配和负载均衡算法来高效求解MDP问题，依据实时的网络和流量状况将攻击流量分配到多条路径进行并行传输和检测。

如图3所示，面向攻击流量的流量分配和负载均衡算法是利用“双网络”机制和“演员-评论”结构来提升算法处理流量分配和负载均衡的稳定性和收敛性，而且使系统更加灵活和高效地响应网络环境的实时变化，包括攻击流量的突然峰值、攻击流量的重复和负载的波动。演员网络生成并执行流量分配策略，评论家网络评估策略价值并提供反馈。演员网络优化策略，评论家网络通过时间差分（TD, temporal difference）学习估计动作价值，指导策略更新。智能体根据网络和流量状况将攻击流量分配到多条路径传输和检测。

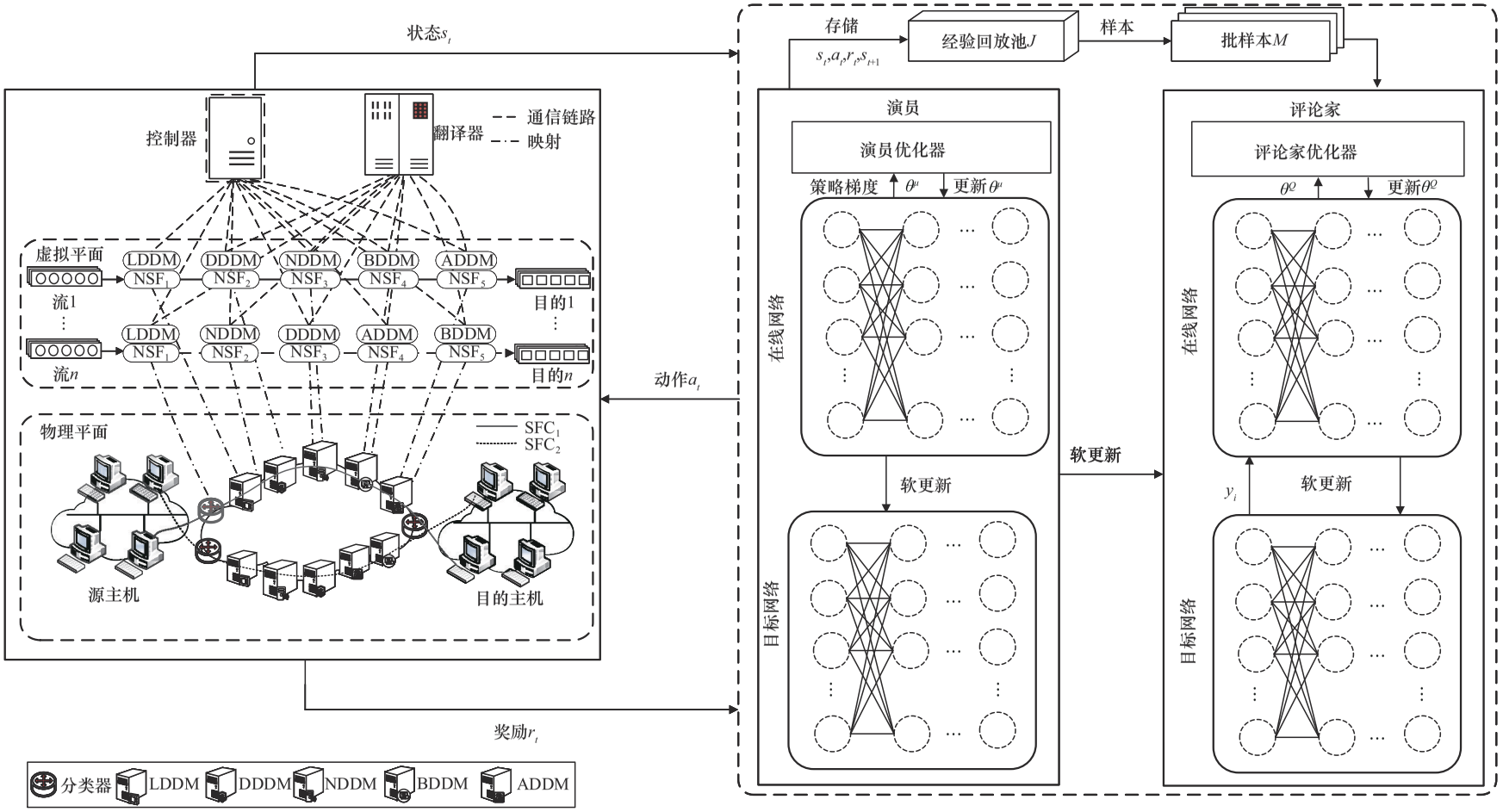


图3　面向攻击流量的流量分配与负载均衡算法结构

演员-评论家网络的工作过程：时隙t、主网络演员网络根据网络状态s_t，智能体通过与环境进行交互，结合策略函数$\pi(a_t|s_t;\theta)$输出流量分配动作a_t，应用到环境中获得奖励r_t。主网络评论家网络的动作价值函数$Q(s_t,a_t;\theta^{\mu})$，负责评估当前状态s_t和流量分配动作a_t。

网络状态从当前状态s_t转移到下一个时隙状态s_{t+1}，目标网络演员网络依靠策略函数$\pi'(a_t|s_t;\theta')$计算并输出下一个流量分配动作值a_{t+1}。目标评论网络利用网络状态s_{t+1}和流量分配动作值a_{t+1}计算并输出目标动作价值$Q'(s_t,a_t;\theta^{\mu'})$。主网络评论网络利用奖励$r_t$和下一个状态-动作对的预期回报$Q(s_{t+1},a_{t+1})$估计状态-动作对的价值函数$y_t$，如式(12)所示。

$$y_t = r_t + \gamma Q(s_{t+1},a_{t+1};\theta') \tag{12}$$

其中，γ是折扣因子，在0和1之间，表示未来奖励的当前价值。

然后利用TD[30]误差更新主网络评论网络价值函数的估计，减少当前估计值和目标值之间的差异，TD误差如式(13)所示。

$$\varsigma_t = \alpha_{\text{critic}}(y_t - Q(s_t,a_t;\theta)) \tag{13}$$

其中，α_{critic}是主网络评论网络的学习率，决定了新信息对旧估计的影响程度。

为了更新主网络评论网络的参数，使用TD的均方误差作为损失函数，定义如式(14)所示。

$$L(\theta^{\mu}) = \frac{1}{2}\sum_t \varsigma_t \tag{14}$$

其中，$L(\theta^{\mu})$是均方误差损失函数，$\sum_t \varsigma_t$表示对所有时隙的TD误差进行求和。

为了提高算法的学习效率，在演员-评论家网络中采用了经验回放技术。在智能体与SDN环境互动时，在每个运行时隙中会生成一系列的经验样本$e_t =< s_t,a_t,r_t,s_{t+1} >$，这些经验样本被存储在一个回放缓冲区$J =< e_1,e_2,\cdots,e_t >$中。当训练智能体时，系统会从回放缓冲区中随机抽取一批样本M。智能体使用这些随机抽取的批样本M进行迭代优化。因此，实际训练过程中的损失函数定义为

$$L(\theta^{\mu}) = \frac{1}{2M}\sum_{m=1}^{M}\sum_t \varsigma_t^{\,2} = \frac{1}{2M}\sum_{m=1}^{M}\sum_t [\alpha_{\text{critic}}(y_t - Q(s_t,a_t;\theta))]^2 \tag{15}$$

主网络评论网络的目标是最小化TD误差，从而提高估计动作价值函数的准确性。主网络演员网络则根据主网络评论网络提供的反馈更新其策略，找到最优策略，即在给定状态下选择能够最大化预期回报的动作。

主网络演员网络是通过采用策略梯度算法学习和更新演员网络的参数，得到一个最佳策略。策略梯度算法的目标函数$J(\theta)$表示为

$$J(\theta) = E_{s\sim\rho^{\pi},a\sim\pi}[G_t] \tag{16}$$

其中，ρ^{π}是遵循策略π下的状态分布，$a\sim\pi$表示动作是根据策略π随机选择的，G_t是从时间t开始的累积折扣回报，定义为

$$G_t = \sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \tag{17}$$

其中，r_{t+k+1}是在时间$t+k+1$获得的即时奖励。

策略梯度算法的目标是找到一组参数θ，使得目标函数$J(\theta)$最大化。为了实现这一点，对目标函数进行梯度并沿着梯度的方向更新参数。策略梯度的更新表示为

$$\theta_{\text{new}} = \theta_{\text{old}} + \alpha_{\text{actor}}\nabla_{\theta}\ln\pi_{\theta}(a_t|s_t)G_t \tag{18}$$

其中，α_{actor}是主网络演员网络的学习率，$\ln\pi_{\theta}(a_t|s_t)$是在状态s_t下采取动作a_t的策略的对数概率。式(18)确保了当策略π产生高回报的流量分配动作时，策略参数θ会朝着增加该动作概率的方向更新。通过这种方式，策略网络逐渐学习到一个能够最大化预期回报的策略。

目标网络的演员网络和评论网络采用软更新的算法进行更新。软更新是一种加权平均的方式，它通过平滑地更新目标网络的参数，将主网络的参数以一定的比例复制到目标网络，使目标网络的参数逐渐接近主网络，以稳定训练过程并提高收敛性。软更新为

$$\theta' = \alpha\theta + (1-\alpha)\theta' \tag{19}$$

$$\theta^{\mu'} = \alpha\theta^{\mu'} + (1-\alpha)\theta^{\mu} \tag{20}$$

选择一个合适的目标网络学习率α对模型的更新稳定性和收敛速度都比较重要。学习率α越小，模型更新越平缓，算法收敛速度越慢，学习率α越大，更新的抖动越明显，收敛速度越快。

综上所述，AALB算法的基本过程是从网络环境中采样状态s，利用经验回放池中数据更新主网络评论家的参数θ^{μ}，根据目标函数更新主网络演员

参数θ，采用软更新的算法更新目标网络中演员网络参数θ'和评论家网络的参数$\theta^{\mu'}$，最终获得最优的策略函数。

AALB算法包括离线训练过程和在线流量分配过程。离线训练过程伪代码如算法1所示。首先，初始化主网络演员网络和评论家网络的学习率α_{actor}和α_{critic}，神经网络参数θ和θ^{μ}，目标网络演员网络和评论家网络参数θ'和$\theta^{\mu'}$，目标网络学习率α，折扣因子γ，经验回放池J，批样本大小M，代理训练轮次数N_{episode}，时间差分误差ς_t。其次，在每次循环开始时，从SDN中收集流量状态和网络状态。输入收集的信息，初始化系统状态s_t。基于当前状态s_t，使用策略函数$\pi(a_t|s_t;\theta)$选择动作a_t（即选择的多个SFC上），并根据选择的多个SFC是否满足约束条件（式(3)~式(8)）计算即时奖励r_t，代理读取下一个状态s_{t+1}（第5行）。将当前状态对应的经验数据存入经验回放池J，从经验回放池J中随机提取M个批样本（第6、7行）。利用批样本M、价值函数y_t和时间差分误差ς_t更新评论家网络的参数θ^{μ}，通过损失函数优化时间差分误差ς_t（第8、9行）。采用策略梯度算法更新主网络演员网络的参数θ，在更新演员网络后，用软更新的算法同步更新目标网络的参数θ'和$\theta^{\mu'}$（第10~12行）。当损失函数收敛或到达最大训练次数时，离线训练过程终止。

在线流量分配过程中，智能体通过控制器和翻译控制器协同实时感知网络状态和流量状态，这些信息经过预处理后，被输入已经训练好的策略网络中，生成流量分配的路径，并获得相应反馈奖励。智能体通过试错法修改其网络参数以训练策略网络模型，并做出在线流量分配决策，从而最大化系统长期积累性能。只有当策略网络的准确度急剧下降时，策略网络模型才需重新训练。

算法1 离线训练过程

初始化折扣因子γ，目标网络学习率α，主网络演员网络和评论家网络学习率α_{actor}和α_{critic}，主网络演员网络和评论家网络的神经网络参数θ和θ^{μ}，目标网络演员网络和评论家网络参数θ'和$\theta^{\mu'}$，经验回放池J，批样本大小M，代理训练轮次数N_{episode}，时间差分误差ς_t

1) for $N_e \leqslant N_{\text{episode}}$ do
2) 从SDN中采集流量状态和网络状态
3) 初始化系统状态s_t
4) 输入当前状态s_t，依靠当前的策略函数$\pi(a_t|s_t;\theta)$选择动作a_t，计算即时奖励r_t
5) 代理读取下一个状态s_{t+1}
6) 将当前状态对应的经验数据(s_t,a_t,r_t,s_{t+1})存入经验回放池J
7) 从经验回放缓存区J随机提取批样本M
8) 根据式(12)计算第t个经验数据的状态−动作的价值函数
9) 根据式(13)计算目标值与当前估计值之间的时间差分误差
10) 利用式(15)更新评论家网络的参数
11) 采用式(18)更新演员网络的参数
12) 根据式(19)~式(20)更新目标网络的参数θ'和$\theta^{\mu'}$
13) 输出：根据当前状态选择最优动作的神经网络和评估给定状态、动作的价值的神经网络
14) end for

AALB算法复杂度分析。首先，初始化阶段是初始化变量，复杂度是$O(\theta+\theta^{\mu})$。其次，在主循环阶段（第2~6行），当前状态、动作、奖励存入经验回放池，复杂度为$O(1)$，从经验回放池中随机提取批样本（第7行），复杂度是$O(M)$，然后根据给定的式(12)、式(13)计算状态−动作的价值函数和目标值与当前估计值之间的时间差分误差，这部分需要计算每个样本，复杂度是$O(M)$（第8、9行）。最后，更新评论家、演员网络和目标网络参数，复杂度是$O(M\times(\theta+\theta^{\mu})+(\theta+\theta^{\mu}))$，因为每个轮次$N_{\text{episode}}$都需要更新演员和评论家网络的参数，并且目标网络参数也会更新，所以，简化后总复杂度$O(N_{\text{episode}}\times M\times(\theta+\theta^{\mu}))$。

4 实验评估

本节介绍实验设置情况，在各种参数下验证所提算法的性能并选择最佳参数，以及使用公开数据集和自生成数据集验证负载均衡能力、吞吐量和流量分布情况。

4.1 实验设置

4.1.1 原型平台设置

本节一共使用8个物理机，采用VMware ESXi搭建虚拟平台，在物理机上用Vmware vCenter Server创

建和管理42台虚拟机，每个虚拟机都分配了8个CPU虚拟核数、500 GB硬盘、16 GB内存，并安装了Ubuntu 18.04 LTS系统。用1台虚拟机实现图2中的知识平面，已训练好的AALB算法部署在这台虚拟机中。用2台虚拟机分别实现控制平面的控制器和翻译器，采用OpenDaylight实现控制器，采用python3.8实现翻译器。在35个虚拟机中用Open vSwitch实现物理平面中入口、出口分类器和检测模块，在实现检测模块的交换机中，利用docker虚拟化部署检测模块。假设有5条SFC（简称路径1、路径2、路径3、路径4和路径5），每条路径中都由入口分类器、LDDM、DDDM、NDDM、BDDM、ADDM和出口分类器组成，根据网络状态和流量状态，把流量分配到5条路径上，先保证负载均衡再验证各个路径的检测性能。

在物理平面，源主机中用2个虚拟机分别安装正常流量主机和攻击流量主机，在目的主机中用2个虚拟机分别安装被攻击主机和正常流量到达主机。在正常流量主机中模拟5G场景的流量生成模型并生成正常流量，模拟的场景有公共服务、智能家居、用户个人计算机上网和机器类型通信等9种场景[31]。在攻击流量主机中通过Tcpreplay工具以重放的形式生成DRDoS、网络层DDoS、LDoS、应用层DDoS、僵尸网络攻击流量以及多种攻击的混合流量。

4.1.2　AALB算法实现

在实验中，使用3层的多层感知机作为演员网络，第一层输入5维特征，第二层使用200个神经元，前2层的激活函数用ReLU，输出层的激活函数用Tanh[27]。使用一个3层全连接前馈神经网络作为评论家网络，第一层和第二层分别包含200个神经元，并使用ReLU作为激活函数。它的最后一层是一个全连接的线性层，不包含激活函数，输出相应动作的Q值。此外，设置了其他重要超参数：演员网络的学习率α_{actor}为0.001，评论网络的学习率α_{critic}为0.01，目标网络的学习率α为0.001，经验回放池J的大小为1 000，训练轮次数N_{episode}为20 000[32]。

4.1.3　数据集

为了评估所提算法，采用2个数据集：公开数据集CICDDoS2019[33]和自生成数据集。

CICDDoS2019数据集旨在为研究DDoS攻击提供真实且多样的数据样本，以支持网络安全领域的深入研究和实际应用。数据集中的攻击流量通过B-profile模拟人类抽象行为交互生成，确保了数据的真实性和可靠性。在攻击流量方面，数据集进行了细致的划分和标注。攻击数据分为训练和测试两部分，以便研究者能够评估其检测算法的性能。训练数据中包含12种不同的攻击类型，测试数据则包含7种攻击类型，这些攻击类型覆盖了多种网络协议和攻击手法，为研究者提供了丰富的攻击样本。

自生成数据集中攻击流量通过运用多种脚本和工具生成，涵盖了19种不同的攻击类型，旨在模拟现实世界中可能遇到的各种网络攻击情况。本节在攻击流量主机中使用Tcpreply工具重放公开数据集CICDDoS2019中简单服务发现协议（SSDP, simple service discovery protocol）、chargen、用户数据报协议（UDP, user datagram protocol）、同步序列号（SYN, synchronize）攻击，重放自生成数据集中确认字符（ACK, acknowledge character）、slowheaders攻击、slowbody攻击3种攻击，表4为具体攻击类型和数据条数。

表4　攻击类型和数量条数

数据集	攻击类型	数据条数/条
公开数据集	SSDP	14 508
	chargen	13 563
	UDP	33 695
自生成数据集	SYN	356 496
	ACK	131 072
	slowheaders	100 793
	slowbody	110 044

本节使用3种常见的基准算法IACA[19]、DNNR[3]和DRL-TE[20]和与本文所提算法进行对比。

4.1.4　评估指标

评估指标包括奖励、损失、负载均衡能力、吞吐量、流量分布情况、平均加权精准率、平均加权召回率、平均加权F1分数和归一化混淆矩阵，具体描述如下。

1) 奖励和损失分别反映模型的折扣回报和收敛性能。

2) 负载均衡能力：本节用奖励函数代表负载均衡能力，奖励函数中资源成本反映了节点资源的消耗情况，资源成本和负载成正比，因此，用路径负载P_{load}代替资源成本。路径负载P_{load}是由于转发和检测流量产生的路径的所有负载，使用路径中所有检测模块的流量速率总和来近似路径负载，如式(21)所示。

$$P_{\text{load}}=\frac{f_{\text{v}}}{\max f_{\text{v}}} \tag{21}$$

3) 吞吐量：流量从源节点成功到达目的节点的数据量。

4) 流量分布情况：用来衡量不同路径上流量的分布情况，表示在路径上流量的相对大小，以百分比形式表示。

5) 平均加权精准率：对每个类型的精准率加和求平均。

6) 平均加权召回率：对每个类型的召回率加和求平均。

7) 平均加权F1分数：对各个类别的精确率和召回率求加权平均值。

8) 归一化混淆矩阵：分析检测模型的分类结果以及预测类型与实际类型之间的匹配程度。

4.2　评估结果

本节首先通过对比不同折扣因子γ和批样本大小M的模型性能，选择折扣因子γ和批样本大小M参数值。其次，在自生成数据集和公开数据集下对比所提算法与基准算法的性能。最后，验证所提算法选择的路径对应的检测性能。

4.2.1　选择超参数

本节通过对比不同超参数的模型性能，选择了折扣因子γ和批样本大小M的参数值。

1) 折扣因子和奖励

图4为不同折扣因子对应的奖励。由图4可知，折扣因子影响模型的收敛速度和稳定性，当折扣因子为0.9时，未来的回报在流量分配决策过程中的影响将增大，模型更倾向于考虑长期利益。当折扣因子较小（即折扣因子为0.7）时，模型对未来奖励的重视程度降低，更多地依赖于即时奖励进行决策。这导致模型在探索和学习过程中更注重短期收益，需要更长时间才能找到最优策略，因此收敛速度较慢。图4显示，当折扣因子为0.9时，模型的性能最佳，奖励函数能快速收敛。

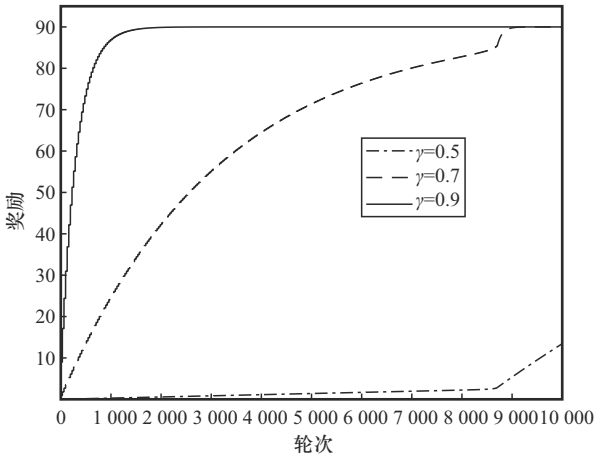


图4　不同折扣因子对应的奖励

2) 批样本大小和损失

图5为不同批样本大小对应的损失。由图5可知，训练批样本大小会影响算法的收敛速度。较小的批样本大小（即批样本大小为16和32）导致经验池中旧的数据训练网络重复使用次数增多，每次更新都基于较小的数据集，算法收敛较慢，损失值波动较大；较大的批样本大小（即批样本大小为64）使损失值变化更平稳。因此，设置批样本大小设置为64。

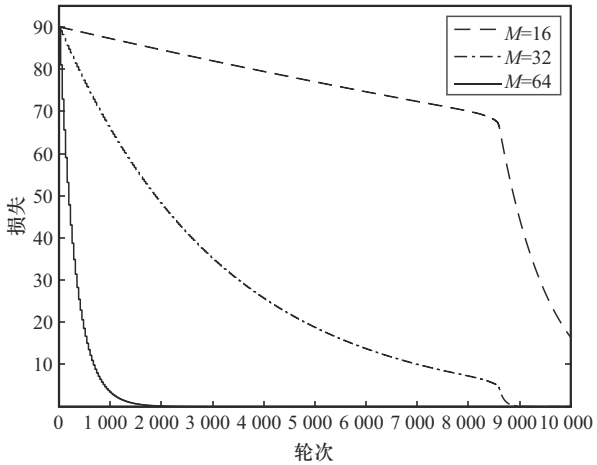


图5　不同批样本大小对应的损失

4.2.2　AALB算法的性能

本节在自生成数据集和公开数据集下对比所提算法与基准算法的性能。注意：本节的所有实验结果经过10次重复测试获得，并取其均值作为最终数据。

1) 重放自生成数据集场景下性能

图6展示了重放自生成数据集中slowheaders攻

击、slowbody 攻击和 ACK 攻击下 AALB、DRL-TE、DNNR 和 IACA 这 4 种算法对应的 LBC 随流量速率的变化。对于上述 4 种算法，随着流量速率的增加，LBC 均呈现不同程度的上升趋势。这是因为随着流量速率的增加，把攻击流量分配到满足网络资源约束的路径上，实现了负载均衡的目的，提升了网络性能。

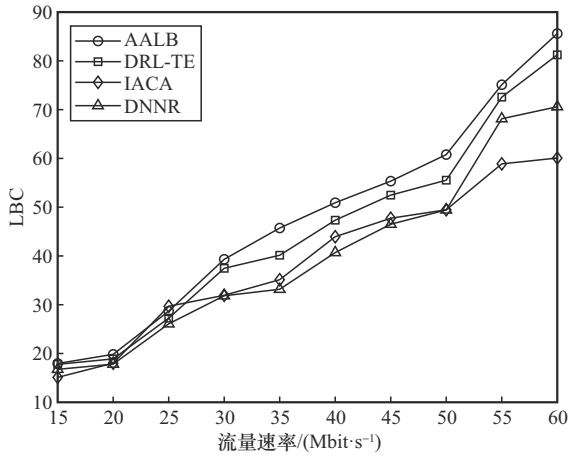


图 6　4 种算法对应的 LBC 随流量速率的变化

具体地，当流量速率低于 25 Mbit/s 时，IACA 的负载均衡能力较强，这是因为 IACA 以负载均衡为优化目标，将流量分配到了不同路径上。当流量速率超过 25 Mbit/s 时，IACA 的负载均衡能力增幅减缓，这是因为 IACA 专注于寻找瞬时最优解，未考虑长期收益。基于 DRL 框架的 AALB 和 DRL-TE 相较于 IACA 和 DNNR 具有更高的负载均衡能力，这是因为基于 DRL 框架的算法通过与环境的不断交互来学习最优策略，因此，AALB 和 DRL-TE 能够根据实际网络状态和攻击流量需求动态调整路径决策，用贪婪算法找到全局最优解，实现更高效的资源利用率。相比之下，IACA 和 DNNR 是基于固定的规则且只关注当前状态，无法适应复杂多变的攻击环境和无法实现长期最优。AALB 的奖励值高于 DRL-TE，说明 AALB 的负载均衡能力强于 DRL-TE。

图 7 为不同算法的吞吐量。具体地，DNNR 对应的吞吐量变化范围仅次于 DRL-TE 和 AALB。IACA 对应的吞吐量变化在 12.5~14.73 Mbit/s，变化范围最广。刚执行 IACA 时，流量被分配到合适的路径，以保障流量正常到达目的主机的数量，当算法执行到一定时间时，算法陷入局部最优解，流量不再被分配到负载较低的路径，导致有些路径会出现吞吐量下降的情况。

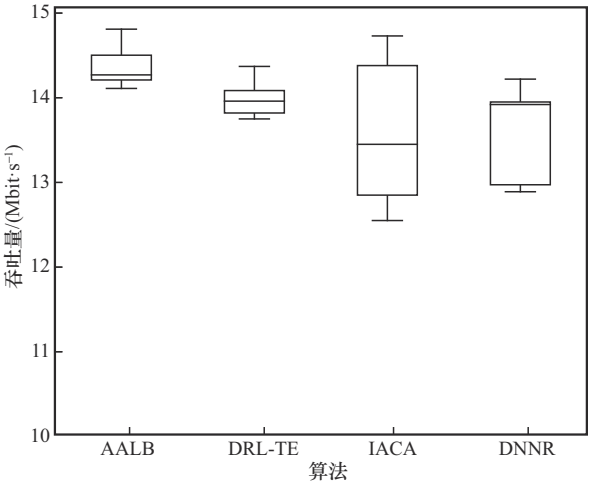


图 7　不同算法的吞吐量

DNNR 在执行正向传播和反向传播过程中，经过多个轮次的迭代，模型会避开局部最优解，使流量分配到合适的路径。因此，DNNR 和 IACA 对应的吞吐量变化范围比较大。DRL 框架是基于最大化长期系统累积奖励，所以 AALB 和 DRL-TE 对应的吞吐量比 DNNR 和 IACA 的吞吐量变化范围小。AALB 对应的吞吐量变化为 14.1~14.81 Mbit/s，吞吐量变化范围相对较小，表明该算法在对比算法中表现出较好的稳定性。DRL-TE 对应的吞吐量变化为 13.75~14.37 Mbit/s，其吞吐量变化略大于 AALB，这也表明 DRL-TE 在攻击流量分配中表现出一定的稳定性。然而，相较于 AALB，DRL-TE 的平均吞吐量降低了 0.40 Mbit/s。当网络中有攻击流量时，AALB 具有较强的负载能力和稳定性。

图 8 展示了在混合攻击流量下执行不同算法的过程中不同路径的流量分布。由图 8 可知，AALB 能使攻击流量均匀分配到 5 个路径中，没有明显的瓶颈或空闲路径，这意味着即使没有任何网络的先验知识，AALB 也可以平衡时延和网络资源利用的多目标优化，以保证正常用户 QoS 需求。与 AALB 相比，DRL-TE 在路径 5 上的流量明显更高，是在该路径上有潜在的拥塞风险。在路径 1 上的流量最低，说明路径 1 的利用率较低。DNNR 在路径流量分布上略显逊色，路径 5 上的流量最高，说明机器学习算法在训练期间学习到了特定流量模式，但实

际应用中攻击流量模式发生变化，可能会导致流量分配不均衡。IACA对应的流量分布波动较大，路径1的流量最低，路径5的流量最高，容易导致一些路径的资源未被充分利用，而其他路径面临拥塞。

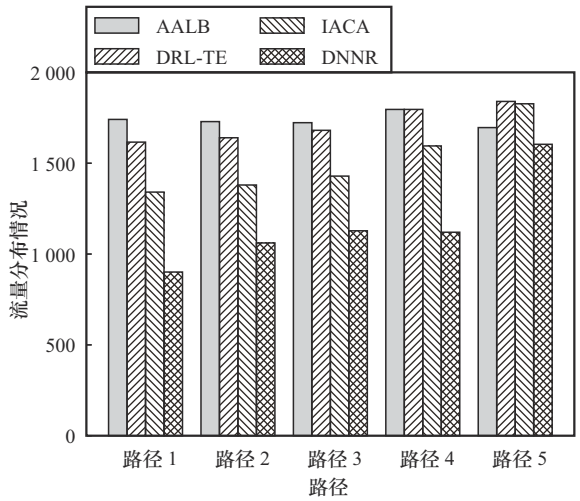


图8　不同路径的流量分布

2) 重放公开数据集场景下的性能

图9展示了重放公开数据集中SSDP、chargen、UDP、SYN攻击下执行流量分配操作后，不同算法的LBC随流量速率变化情况。除IACA外，其他算法随着流量速率的增加，整体LBC在一定程度上有所上升。

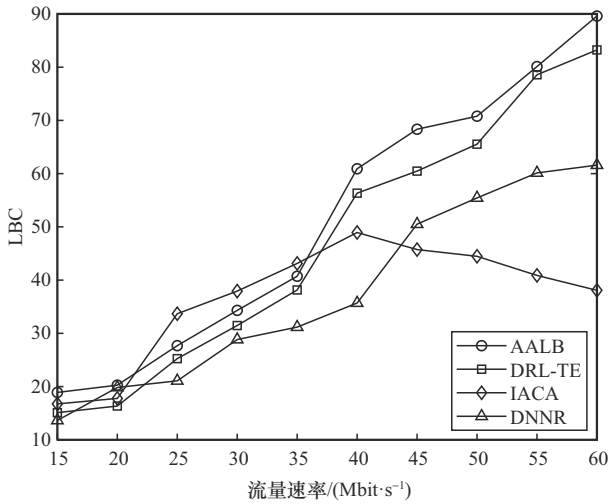


图9　不同算法的LBC随流量速率变化情况

由图9可知，当流量速率为15~40 Mbit/s时，IACA能为流量分配到合适的路径，当流量速率超过40 Mbit/s时，该算法注重求解瞬时最优解，导致当流量速率不断提高，分配的路径存在拥塞的情况，增加流量到达目的主机的时间。随着流量速率的增加，AALB比DRL-TE的负载均衡能力始终较高，说明AALB能把流量分配到路径，更能找到负载平衡的最优解。

图10展示了执行流量分配到不同路径中，不同算法的吞吐量。具体地，IACA对应的吞吐量变化范围最大，其对应的吞吐量变化为11~14.72 Mbit/s。AALB对应的吞吐量变化为12.95~14.83 Mbit/s，DRL-TE对应的吞吐量变化为12.85~14.78 Mbit/s，DNNR对应的吞吐量变化为12.02~14.7 Mbit/s。

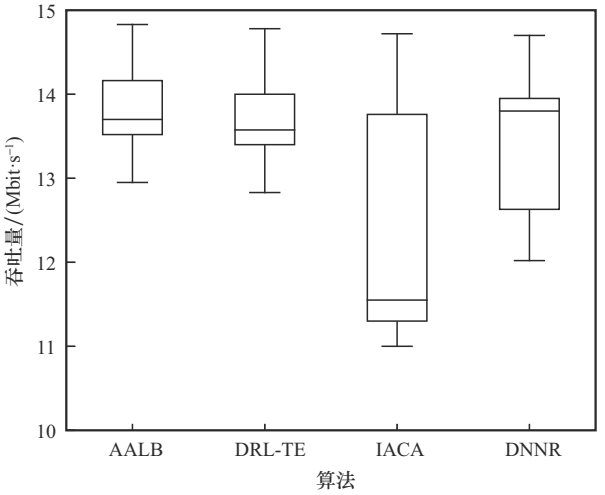


图10　不同算法的吞吐量

由图10可知，DNNR和IACA的吞吐量具有较大的波动性，因为其算法分配的路径出现了拥塞，而AALB和DRL-TE在吞吐量方面具有较好的稳定性。图10中结果显示，AALB具有更好的稳定性。

图11展示了在公开数据集下执行不同算法过程中，不同路径的流量分布。IACA基于规则信息寻找流量分配问题。IACA在路径4上的流量数显著高于其他路径的，这可能表明IACA在这个特定路径上表现非常出色。然而，在其他路径上，流量分布相对均匀。DNNR在路径1和路径2上有较高的流量数，但在路径3~路径5上的表现较差。在DRL-TE中，流量能被分配到路径2~路径5，但是，路径1的流量相对较低，表明该路径可能出现拥塞。AALB在所有路径上的流量分布都比较均匀，表明它在流量分配方面表现较好。

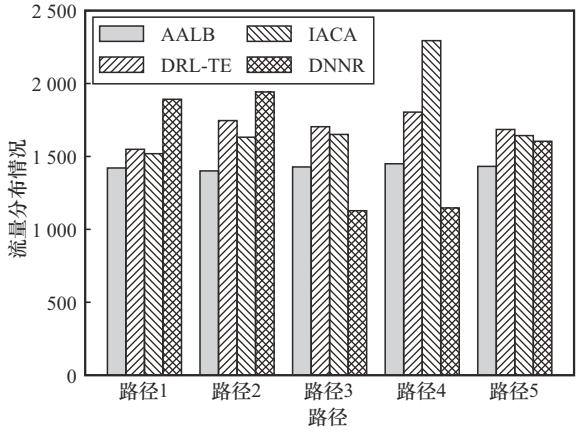


图11　不同路径的流量分布

3) 算法性能

把AALB应用在图2所示的系统中，重放公开数据集和自生成数据集中混合攻击流量被分配到5个路径后，用归一化混淆矩阵和性能指标直观地表示每种攻击类型的分类情况。表5为不同攻击的检测性能。

表5　　不同攻击的检测性能

类型	平均加权精准率	平均加权召回率	平均加权F1分数
chargen	0.99	0.99	0.99
SSDP	0.99	0.96	0.97
UDP	0.99	0.95	0.95
SYN	0.99	0.92	0.96
slowheaders	0.90	0.98	0.94
slowbody	0.93	0.99	0.96
ACK	0.99	0.98	0.99

图12为自生成数据集的归一化混淆矩阵，矩阵中斜对角线代表召回率，识别出所有攻击类型的召回率都高于0.96，说明这些攻击流量中存在少量攻击流量误判为其他攻击类型。具体地，由于slowheaders和slowbody攻击的恶意行为相似，都利用了HTTP协议的特性和机制来消耗服务器的连接和内存资源，从而达到攻击的目的，因此，2种攻击类型容易相互误判。

图13为公开数据集的归一化混淆矩阵，矩阵中识别所有攻击类型的召回率都高于0.95，具体地，chargen、SSDP和UDP之间容易相互误判。其主要原因包括协议相似性、攻击特征重叠和易于模仿。在协议相似性方面，chargen攻击和SSDP攻击都是基于UDP的，与UDP洪水攻击共享相同的传输层协议。这种协议层面的共通性使得在流量分析时难以仅凭协议类型区分攻击类型。在攻击特征重叠方面，3种攻击都可能表现为高频率的UDP数据包交换，都有大量的短数据包发送和接收，这种流量模式的相似性可能导致在流量检测中被混淆。在易于模仿方面，UDP洪水攻击可以通过控制数据包的大小和发送频率来模仿chargen或SSDP攻击的流量特征，从而在一定程度上规避检测。

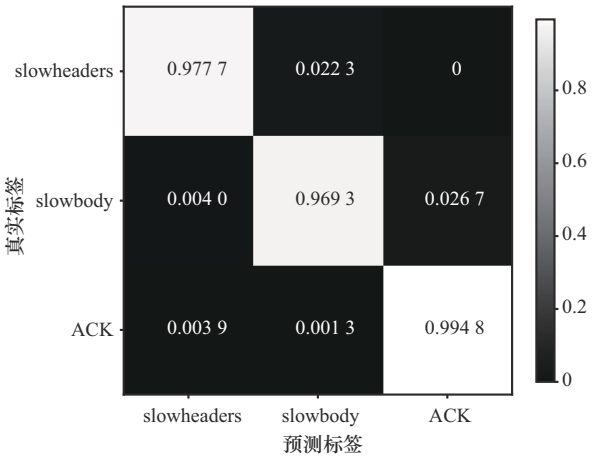


图12　自生成数据集的归一化混淆矩阵

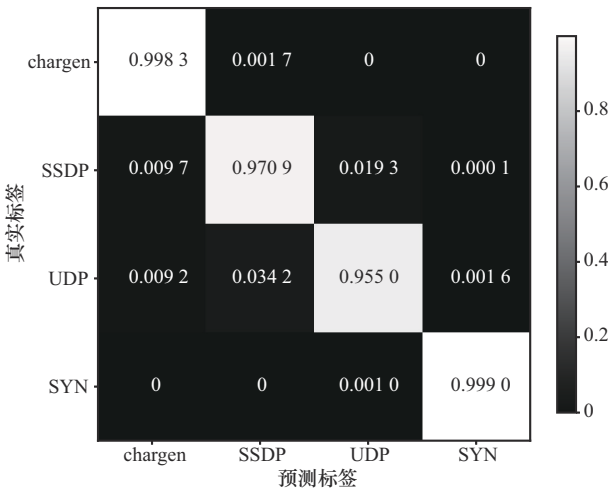


图13　公开数据集的归一化混淆矩阵

综上所述，在自生成数据集下，在流量速率低于25 Mbit/s时，AALB的负载均衡能力较强，能够将流量均匀分配到不同路径上。当流量速率超过25 Mbit/s时，AALB依然保持较高的负载均衡能力。在公开数据集中，在流量速率低于40 Mbit/s时，AALB能够为流量分配到合适的路径，确保网络资源的高效利用。当流量速率超过40 Mbit/s时，

AALB仍有较高的负载均衡能力值，通过调整路径决策，避免路径拥塞。

此外，在自生成数据集和公开数据集下，AALB都表现出较稳定的吞吐量和流量分布，在检测性能方面，识别攻击的平均加权精准率、平均加权召回率和平均加权F1分数分别达到0.90、0.92和0.94。

5 结束语

随着网络规模的不断扩大和业务需求的日益复杂，SDN和网络攻击场景下流量分配和负载均衡成为关键问题。为此，本文提出了一种面向攻击流量的流量分配与负载均衡机制。将流量分配问题建模为MDP模型，综合考虑资源消耗和时延2个关键因素。为了优化MDP，进一步设计了一种基于演员-评论家强化学习框架的负载均衡算法，该算法能根据网络状态和流量特征，智能地为流量分配到不同的路径，确保网络负载的均衡性和安全性。实验结果表明，相较于基准负载均衡算法，所提算法在自生成数据集和公开数据集下均展现出更强的负载均衡能力，同时在吞吐量和流量分布稳定性方面表现更佳。此外，实验结果证明所提算法能够在不同路径之间实现流量分配，并保持了一定的检测性能。本文所提算法主要侧重于优化网络负载均衡，根据网络和流量状态将流量分配到不同的路径中，以提升网络效率。然而，所提算法在网络安全性能的考虑上尚显不足，并且缺乏对其他常见DDoS攻击类型的检测性能验证。因此，未来的研究可在此基础上进行扩展，综合考虑网络状态、流量管理和安全防护，进一步提升所提算法的网络安全能力与负载均衡能力，从而为DDoS攻击的检测提供更加全面和有效的解决方案。

参考文献：

[1] DONG S, SU H D, XIA Y J, et al. A comprehensive survey on authentication and attack detection schemes that threaten it in vehicular ad-hoc networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(12): 13573-13602.

[2] AULIA M A, SUKMANDHANI A A, OHLIATI J. RIP and OSPF routing protocol analysis on defined network software[C]//Proceedings of the 2022 International Electronics Symposium (IES). Piscataway: IEEE Press, 2022: 393-397.

[3] REIS J, ROCHA M, PHAN T K, et al. Deep neural networks for network routing[C]//Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN). Piscataway: IEEE Press, 2019: 1-8.

[4] RUSEK K, SUÁREZ-VARELA J, MESTRES A, et al. Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN[C]//Proceedings of the 2019 ACM Symposium on SDN Research. New York: ACM Press, 2019: 140-151.

[5] VALADARSKY A, SCHAPIRA M, SHAHAF D, et al. Learning to route[C]//Proceedings of the 16th ACM Workshop on Hot Topics in Networks. New York: ACM Press, 2017: 185-191.

[6] 肖军弼, 程鹏, 谭立状, 等. 基于SDN的数据中心动态优先级多路径调度算法[J]. 计算机与现代化, 2020(7): 21-26.
XIAO J B, CHENG P, TAN L Z, et al. Data center dynamic priority multipath scheduling algorithm based on SDN[J]. Computer and Modernization, 2020(7): 21-26.

[7] CASAS-VELASCO D M, RENDON O M C, FONSECA N L S D. Intelligent routing based on reinforcement learning for software-defined networking[J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 870-881.

[8] HUONG T T, KHOA N D D, DUNG N X, et al. A global multipath load-balanced routing algorithm based on Reinforcement Learning in SDN[C]//Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC). Piscataway: IEEE Press, 2019: 1336-1341.

[9] SUN S M, WANG M Y, ZHOU L, et al. Load balancing algorithm for SDN multi-controller with elite genetic algorithm[C]//Proceedings of the 2023 IEEE 6th International Conference on Electronics and Communication Engineering (ICECE). Piscataway: IEEE Press, 2023: 146-150.

[10] SURABHI S, PADMAJA PREK SHA D, MUKHOPADHYAY A. Load balancing in software defined network using multiple controllers[C]//Proceedings of the 2023 International Conference on Computer Science and Emerging Technologies (CSET). Piscataway: IEEE Press, 2023: 1-7.

[11] ABEDINI BAGHA M, MAJIDZADEH K, MASDARI M, et al. ELA-RCP: an energy-efficient and load balanced algorithm for reliable controller placement in software-defined networks[J]. Journal of Network and Computer Applications, 2024, 225: 103855.

[12] SHAIKH M R R. Bayesian network based optimal load balancing in software defined networks[C]//Proceedings of the 2023 International Conference on Emerging Smart Computing and Informatics (ESCI). Piscataway: IEEE Press, 2023: 1-5.

[13] DAFDA J, SUBHEDAR M. Dynamic load balancing in SDN using energy aware routing and optimization algorithm[C]//Proceedings of the 2022 IEEE Bombay Section Signature Conference (IBSSC). Piscataway: IEEE Press, 2022: 1-6.

[14] BELGAUM M R, MUSA S, ALI F, et al. Self-socio adaptive reliable particle swarm optimization load balancing in software-defined networking[J]. IEEE Access, 2023, 11: 101666-101677.

[15] ZHANG J J, XI K, ZHANG L R, et al. Optimizing network performance using weighted multipath routing[C]//Proceedings of the 2012 21st International Conference on Computer Communications and Networks (ICCCN). Piscataway: IEEE Press, 2012: 1-7.

[16] CHAHLAOUI F, DAHMOUNI H, EL ALAMI H. Multipath-routing based load-balancing in SDN networks[C]//Proceedings of the 2022 5th Conference on Cloud and Internet of Things (CIoT). Piscataway: IEEE Press, 2022: 180-185.

[17] ZUO L Y, SHU L, DONG S B, et al. A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing[J]. IEEE Access, 2015, 3: 2687-2699.

[18] ZHOU J L, TEWARI M, ZHU M, et al. WCMP: weighted cost mul-

tipathing for improved fairness in data centers[C]//Proceedings of the Ninth European Conference on Computer Systems. New York: ACM Press, 2014: 1-14.

[19] SHI H Q, HAO Z L, SHI H Q. A dynamic load balancing strategy based on improved ant colony algorithm[J]. Journal of Physics: Conference Series, 2021, 1871(1): 012140.

[20] XU Z Y, TANG J, MENG J S, et al. Experience-driven networking: a deep reinforcement learning based approach[C]//Proceedings of the IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. New York: ACM Press, 2018: 1871-1879.

[21] HE Q, WANG Y, WANG X W, et al. Routing optimization with deep reinforcement learning in knowledge defined networking[J]. IEEE Transactions on Mobile Computing, 2024, 23(2): 1444-1455.

[22] YE M H, HU Y, ZHANG J J, et al. Reinforcement learning-based traffic engineering for QoS provisioning and load balancing[C]//Proceedings of the 2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS). Piscataway: IEEE Press, 2023: 1-10.

[23] HU Y X, LI Z Y, LAN J L, et al. EARS: Intelligence-driven experiential network architecture for automatic routing in software-defined networking[J]. China Communications, 2020, 17(2): 149-162.

[24] ZHANG L M, CHENG H R, PENG S Q, et al. Flowlet-level routing optimization with GNN-based multi-agent deep reinforcement learning[C]//Proceedings of the GLOBECOM 2023 - 2023 IEEE Global Communications Conference. Piscataway: IEEE Press, 2023: 5214-5219.

[25] YAO H P, YUAN X, ZHANG P Y, et al. Machine learning aided load balance routing scheme considering queue utilization[J]. IEEE Transactions on Vehicular Technology, 2019, 68(8): 7987-7999.

[26] PENG H X, SHEN X M. Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks[J]. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2416-2428.

[27] BAI M M, ZHU R, GUO J X, et al. Energy-efficient power control strategy of the delay tolerable service based on the reinforcement learning[J]. Computer Communications, 2023, 210: 102-115.

[28] CHENG N, LYU F, QUAN W, et al. Space/aerial-assisted computing offloading for IoT applications: a learning-based approach[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(5): 1117-1129.

[29] CHOWDHARY A, HUANG D J. SDN based network function parallelism in cloud[C]//Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC). Piscataway: IEEE Press, 2019: 486-490.

[30] ZHANG W T, YANG D, PENG H X, et al. Deep reinforcement learning based resource management for DNN inference in industrial IoT[J]. IEEE Transactions on Vehicular Technology, 2021, 70(8): 7605-7618.

[31] NAVARRO-ORTIZ J, ROMERO-DIAZ P, SENDRA S, et al. A survey on 5G usage scenarios and traffic models[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 905-929.

[32] XU J, HOU Z M, WANG W, et al. Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks[J]. IEEE Transactions on Industrial Informatics, 2019, 15(3): 1658-1667.

[33] SHARAFALDIN I, LASHKARI A H, HAKAK S, et al. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy[C]//Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST). Piscataway: IEEE Press, 2019: 1-8.

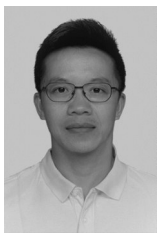
[作者简介]

李曼（1997– ），女，河南濮阳人，博士，北京交通大学工程师，主要研究方向为入侵检测、软件定义网络、网络功能虚拟化等。

周华春（1965– ），男，安徽池州人，博士，北京交通大学教授、博士生导师，主要研究方向为未来互联网体系架构、移动互联网、网络安全、空间网络等。

徐琪（1992– ），男，浙江台州人，博士，之江实验室副研究员，主要研究方向为多模态网络、软件定义网络、服务功能链等。

邓双兴（1998– ），男，河北唐山人，北京交通大学硕士生，主要研究方向为深度强化学习、服务功能链等。



邹涛（1974– ），男，重庆人，博士，之江实验室研究员，主要研究方向为多模态网络、工业互联网、信息安全等。

张汝云（1973– ），男，山东聊城人，博士，之江实验室研究员，主要研究方向为智能计算、多模态网络、网络安全、工业互联网等。