中国科学: 信息科学 2023 年 第 53 卷 第 12 期: 2372–2385
SCIENTIA SINICA Informationis
论文

基于深度强化学习的算力网络主动防御方法

张焘[1,2,5,6], 许长桥[1,2*], 连一博[1,2], 康嘉文[3], 况晓辉[1,2,4]

1. 北京邮电大学计算机学院, 北京 100876
2. 网络与交换技术国家重点实验室, 北京 100876
3. 广东工业大学自动化学院, 广州 510062
4. 军事科学院系统工程研究所, 北京 100101
5. 北京交通大学软件学院, 北京 100044
6. 智能交通数据安全与隐私保护技术北京市重点实验室, 北京 100044

* 通信作者. E-mail: cqxu@bupt.edu.cn

收稿日期: 2023–01–03; 修回日期: 2023–03–28; 接受日期: 2023–06–04; 网络出版日期: 2023–12–07

国家自然科学基金杰出青年项目 (批准号: 62225105)、国家自然科学基金面上项目 (批准号: 61871048, 61872253)、国家自然科学基金青年项目 (批准号: 62102099)、北京交通大学人才基金项目 (批准号: 2023XKRC050) 和中国国家铁路集团有限公司科技研究开发计划项目 (批准号: N2023W012) 资助

摘要 算力网络旨在深度融合算力资源与网络资源, 实现多种资源的高效协同, 最大化资源利用率. 算力网络边缘部分通常采用分布式软件定义网络架构, 构建逻辑集中但物理分散的控制平面, 并将其与数据平面分离, 实现全网算力资源与网络资源的统一调度与编排. 然而, 攻击者极易将控制平面作为首要攻击目标, 发起分布式拒绝服务攻击 (distributed denial of service, DDoS), 使控制平面大面积失效, 严重影响计算任务的实时传输. 为了解决算力网络中的安全问题, 本文创新性地提出了基于深度强化学习的算力网络主动防御方法. 首先, 构建了马尔可夫决策过程 (Markov decision process, MDP) 模型来准确表征交换机与控制器映射关系的动态性, 并设计了一种基于节点介数的奖励函数来反映 DDoS 攻击对控制器部署方案的影响. 其次, 综合考虑多种网络约束, 将多控制器部署问题建模为约束满足问题, 其可行解空间即为 MDP 模型的动作空间. 最后, 提出了一种基于深度强化学习的主动防御算法, 迭代优化动作选择策略, 智能化选择多控制器部署方案. 实验结果表明, 该方法在网络性能几乎无损的前提下, 相比基准方法能够分别提升 13% 和 8% 的防御成功率.

关键词 算力网络, 分布式软件定义网络, 主动防御, 分布式拒绝服务攻击, 深度强化学习

1 引言

根据爱立信的预测 [1], 到 2050 年将会有 240 亿终端设备 (例如多样化的传感器、移动车辆及移动手机等) 接入互联网. 伴随着以人工智能为代表的智能应用快速发展, 海量的终端设备不再局限于

引用格式: 张焘, 许长桥, 连一博, 等. 基于深度强化学习的算力网络主动防御方法. 中国科学: 信息科学, 2023, 53: 2372–2385, doi: 10.1360/SSI-2023-0004
Zhang T, Xu C Q, Lian Y B, et al. Deep reinforcement learning-based moving target defense method in computing power network (in Chinese). Sci Sin Inform, 2023, 53: 2372–2385, doi: 10.1360/SSI-2023-0004

传统的数据感知与收集, 进一步产生了大规模智能任务的计算需求. 因此, 算力资源成为了当前网络中的关键因素. 以往算力资源通常集中在核心网络的云计算中心, 远离网络边缘的终端设备, 从而对网络传输造成了巨大的压力, 难以满足低时延、大带宽、高可靠性及确定性等需求. 此外, 计算资源部署分散并且资源分配与调度过程中缺乏高效协同, 导致计算任务完成实时性差, 计算资源利用率低. 为了有效地解决上述问题, 近年来提出了一种将算力资源与网络资源深度融合的新兴网络架构, 命名为算力网络, 其结合了边缘计算 [2]、软件定义网络 [3] 等前沿技术. 例如, 我国在 2021 年发布了《全国一体化大数据中心协同创新体系算力枢纽实施方案》, 实施了 “东数西算” 工程, 构建国家算力网络体系 [4].

算力网络具有高分布式计算资源的特性, 因此最具代表性的网络控制方案是采用分布式软件定义网络 (distributed software-defined network, distributed SDN), 将数据平面与控制平面分离, 其中控制平面具有全局视图, 能够集中管理整个网络的算力资源与网络资源, 并进行统一的调度编排 [5]. SDN 网络集中控制的特性使得控制器成为了网络架构中的安全薄弱环节, 极易遭受分布式拒绝服务攻击 (distributed denial of service, DDoS). 当控制链路被攻击流量淹没或者控制器的处理资源被耗尽时, 整个网络的时延与丢包率将会剧烈上升, 难以满足算力网络中实时传输与实时计算的需求, 破坏了网络与算力的深度融合. 因此, 在算力网络的分布式 SDN 架构中研究如何抵御针对控制层的 DDoS 是十分必要的.

目前抵御控制层 DDoS 的主要思路是识别 SDN 网络中的 DDoS 攻击流量, 进而将这些攻击流量过滤. 例如, 文献 [6] 提出一种基于信息熵的 DDoS 检测算法, 其通过提取流量特征并构建相应的特征矩阵, 进而计算联合熵来识别 DDoS 攻击流量. Zhang 等 [7] 设计了一种基于深度学习的 DDoS 攻击检测方案, 利用深度神经网络学习攻击流量的显著特征, 从而实现对 DDoS 攻击的识别. 但是这些 SDN 网络中的传统 DDoS 防御机制通常是在攻击发生之后进行防御响应, 意味着 DDoS 攻击已经对网络通信造成负面影响. 此外, 静态的防御部署给攻击者提供了充足的攻击准备时间, 造成攻击者在攻防博弈中具有明显的不对称优势. 面对前述挑战, 主动防御 [8] 是一种十分有前景的解决方案, 其通过周期性动态调整网络属性或者网络配置, 将动态性与不可预测性引入至网络系统, 无效化攻击者收集到的网络信息, 彻底扭转攻击者的不对称的优势. 然而, 目前主动防御主要关注网络路由跳变 [9]、网络地址跳变 [10]、服务功能链主动迁移 [11] 等方面, 难以面对控制层 DDoS 的威胁.

因此, 本文创新性提出了一种基于深度强化学习的算力网络主动防御方法, 周期性动态调整控制节点的部署位置及交换机节点与控制器节点的映射关系. 综合考虑多种实际网络约束, 将多控制器放置问题构建为约束满足问题, 移除 MDP (Markov decision process) 中的不可行动作. 接下来, 利用深度强化学习算法学习如何选择不同的多控制器放置方案, 并根据环境反馈迭代优化动作选择策略, 显著降低控制层 DDoS 对算力网络的破坏范围. 近年来, 强化学习已经被广泛使用来实现增强网络系统安全性的目标 [12]. 据我们所知, 本文是首个针对控制层 DDoS 的智能化主动防御工作.

本文的主要贡献包括 3 个方面:

(1) 构建了马尔可夫决策过程 (MDP) 模型来准确表征多控制器与交换机间映射关系的动态性, 并设计基于节点介数的奖励函数来描述控制层 DDoS 对多控制器部署方案的影响. 此外, 综合考虑了网络性能与能量消耗约束, 将多控制器放置位置及交换机节点与控制器节点的映射关系建模成约束满足问题, 从而将不可行的多控制器部署方案从 MDP 的动作空间移除.

(2) 提出了一种基于近端策略优化 (proximal policy optimization, PPO) 的主动防御算法, 周期性从前述移除不可行动作后的动作空间内选择多控制器部署方案, 并根据 DDoS 攻击造成交换机与控制器无法正常运行的情况计算奖励, 进而反馈优化动作选择策略实现感知攻击者的网络位置, 降低控制

层 DDoS 攻击的破坏范围.

(3) 在 Mininet-WiFi 网络仿真器中构建了实验环境, 实验结果证明, 与多控制器部署的基准方法和主动混合防御机制相比, 本文所提方法能够在网络性能基本无损的前提下, 分别提升大约 13% 和 8% 的防御成功率, 显著降低了 DDoS 攻击效果.

2 相关工作

本节对算力网络及目前 SDN 网络中抵御 DDoS 攻击的研究进展进行简要概述.

算力网络能够从全局角度统筹算力资源与网络资源, 合理分配不同需求的计算任务, 实现实时传输与实时计算, 达到算网资源的最优利用与高效协同. 近年来, 越来越多的学者开始关注算力网络的研究. 文献 [13] 设计了一种基于网络孪生的算力网络架构, 能够通过协同调度计算、存储和传输资源共同满足多种业务的网络质量需求. 为了充分利用闲置的网络资源来处理资源匮乏或者实时性要求高的计算任务, Di 等 [14] 提出了一种网络资源池化框架, 采用基于注意力机制的深度强化学习算法解决动态资源池中计算和缓存资源协同调度问题. 文献 [15] 提出了一种算力网络架构, 能够保障用户的适应性、网络的灵活性, 以及计算资源的可支配性. Liu 等 [16] 提出了一种多层次算力网络架构, 进而设计了一个云雾混合多层次计算卸载系统并构建混合付费模型, 解决代价感知的任务调度问题. 尽管有很多算力网络架构与计算任务分配调度机制的研究, 但仍缺乏算力网络安全方面的深入研究.

目前算力网络边缘部分通常采用分布式 SDN 架构, 将控制平面与数据平面分离, 构建逻辑集中但物理分散的控制平面, 协同调度全网计算资源. 当前已有一些研究关注分布式 SDN 下联合交换机与控制器的放置问题 [17]. 但在这种情况下, 控制器极易成为 DDoS 攻击的首要目标, 当遭到 DDoS 攻击时, 控制器无法正常提供网络调度与控制功能, 造成计算任务难以实时完成. SDN 网络中抵御 DDoS 的传统安全机制主要思路是识别网络流量中的 DDoS 流量并做出防御响应 [18]. 文献 [19] 提出了一种基于极限梯度增强的特征选择方法, 并通过混合卷积神经网络和长短期记忆神经网络结合来确定相关性最高的数据特征, 最终实现 DDoS 攻击分类. 为了解决针对控制器的 TCP SYN 泛洪攻击, Ravi 等 [20] 设计了一种命名为 AEGIS 的轻量级检测机制, 周期性检测是否存在由于持续的 SYN 泛洪而导致的控制器性能滞后, 如果检测到 DDoS 攻击, 则会触发相应的缓解机制. 为了解决 DDoS 攻击造成的 SDN 网络控制平面单点故障问题, 文献 [21] 提出一种基于长短期记忆神经与分级单元相结合的深度学习模型作为检测模块, 实现 DDoS 攻击识别. 然而, 这些传统安全机制均是在检测到 DDoS 攻击后进行防御响应, 一直处于被动防御的状态, 造成这些防御方法的有效性一直受到限制.

为了彻底扭转攻击者在攻防博弈中的不对称优势, 主动防御在安全领域得到了广泛的关注与研究, 部分工作已经研究了如何在 SDN 网络中抵御 DDoS 攻击. 例如, Zhou 等 [22] 设计了一种混合主动防御机制, 与网络欺骗技术相结合, 传播伪装的网络信息来迷惑攻击者, 并构建了一个防御者主导的信号博弈模型来优化防御部署策略. 文献 [23] 提出了一种基于 SDN 的成本感知边缘主动防御方法, 以低部署成本不断地洗牌与关键服务器具有最多连接数的主机, 实现 DDoS 攻击效果的减轻. 文献 [24] 提出了一种基于强化学习的路由跳变机制, 主动规避受到 DDoS 攻击的网络节点, 利用智能算法迭代优化跳变策略, 实现防御效果最大化. Zhang 等 [25] 在软件定义车联网中提出了一种基于深度强化学习的基站动态覆盖机制, 智能化减轻 DDoS 攻击影响, 车辆在多个基站之间切换时, 进一步评估车辆的信誉, 从而在源头阻断 DDoS 攻击. 然而, 上述主动防御方法均未考虑控制层面的 DDoS 攻击, 因此难以适用分布式 SDN 场景. 据我们所知, 本文工作是第 1 个为控制平面设计的基于深度强化学习的主动防御方法.

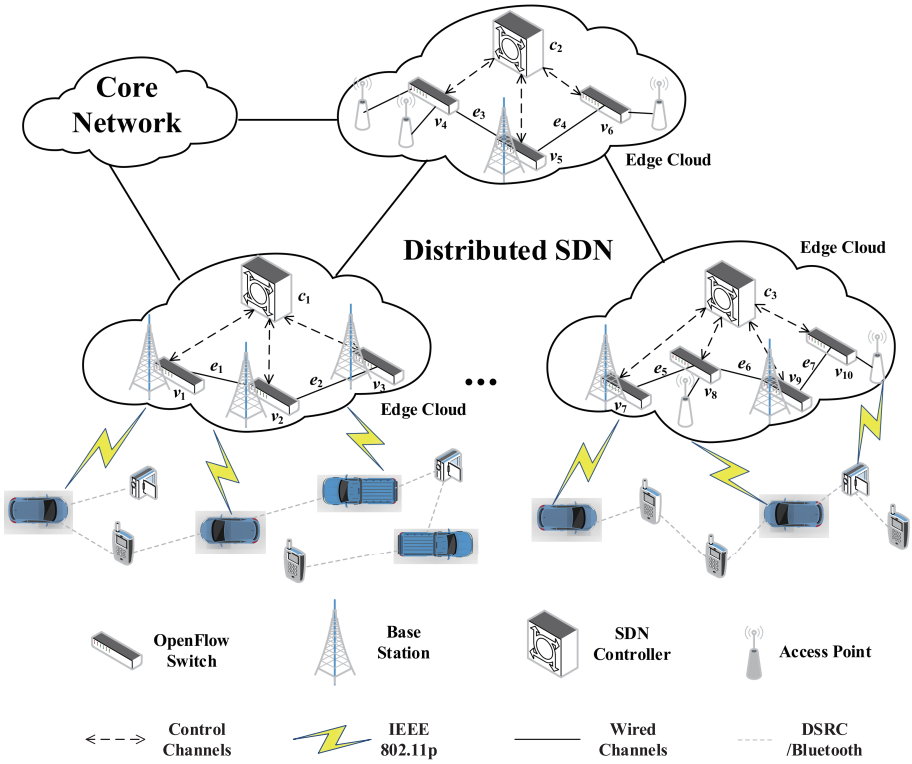

图 1 (网络版彩图) 基于分布式软件定义的算力网络架构示意图

Figure 1 (Color online) The framework of the distributed software-defined computing power network

3 算力网络主动防御的模型构建

在本节中, 我们分别介绍了算力网络的网络模型、攻击模型及 MDP 模型.

3.1 网络模型

如图 1 所示, 基于分布式 SDN 的算力网络主要包含 3 个层次, 即远端的云计算 (位于核心网)、分布式的边缘云计算, 以及泛在接入的终端设备 (例如, 多样化的传感器节点、移动车辆, 以及移动手机等). 其中, 边缘云计算采用了控制平面与数据平面分离的分布式 SDN 架构, 主要包括交换机节点和控制器节点, 交换机根据控制器下发的流表执行数据包转发, 控制器则具有网络资源与算力资源的全局视图, 可以实现网络与算力的全局统一编排调度. 此外, 分布式 SDN 构建的是逻辑上集中但物理上分散的控制平面, 体现了分布式结构的可扩展性和可靠性. 在这种网络架构下, 控制平面可以根据计算负载情况与网络通信质量动态地将任务传输到最佳网络节点进行计算, 从而满足终端用户的计算需求.

我们采用一个无向图 $G = (N, M, E)$ 来表示交换机间的网络拓扑, 其中 $N = \{v_1, v_2, \ldots, v_n\}$ 表示交换机节点的集合, $M = \{c_1, c_2, \ldots, c_m\}$ 表示控制器节点的集合, $E = \{e_1, e_2, \ldots, e_w\}$ 表示交换机节点间的通信链路. 根据现有研究工作中的结论 [26~28], 交换机节点迁移, 即动态地从连接某个控制器节点迁移到连接另一个控制器节点, 是广泛使用的负载均衡技术.

3.2 攻击模型

在基于分布式 SDN 的算力网络中, 多控制器承担着网络控制决策的重要作用, 因此同样成为了算力网络的安全脆弱点. 假设攻击者控制了多台僵尸主机, 并且这些僵尸主机已经接入算力网络, 其网络位置随机分布且保持固定, 但网络位置的分布对防御者是未知的. 对攻击者而言, 其无法获知控制器与交换机的映射关系, 因此无法主动选择特定的控制器作为攻击目标. 接下来, 攻击者指挥这些僵尸主机向各自关联的控制器发送大量伪造的攻击数据包, 尤其是伪造了目的 IP 地址及端口号等网络信息[29]. 这些数据流对交换机来说都是新到达的数据流, 因此会触发交换机向关联控制器发送询问请求, 从而淹没交换机与控制器之间的控制链路或严重消耗控制器的处理资源, 使得该控制器管理的所有交换机都无法正常运行. 控制层 DDoS 能够造成合法数据流被丢弃或者网络通信质量的严重下降, 难以满足算力网络中任务实时传输与计算的需求, 破坏网络与算力的深度融合.

3.3 马尔可夫决策过程 (MDP) 模型

首先, 将时间划分成相等的间隔, 每个间隔的长度为 ΔT, 那么时间可以表示为 $t \in \{0,1,2,\ldots\}$. 为了准确表达多控制器与交换机间映射关系的动态性, 将其建模成 MDP 模型, 主要包含了以下几个关键的元素.

(1) 状态 (state). 算力网络的网络状态可以表示为一个 n 维向量 $S_t = \{s_1, s_2, \ldots, s_n\}$, 其中 s_i $(1 \leqslant i \leqslant n)$ 表示第 i 个交换机节点是否能够正常运行, 如果它能够正常地提供数据转发, 那么 s_i 的值为 1, 否则为 0. 根据前述定义, 状态空间将包括 2^n 个不同的网络状态.

(2) 动作 (action). 控制器执行的跳变主要包括两个部分, 一是将控制器节点放置到恰当的交换机节点上, 二是确定交换机节点与控制节点的映射关系, 因此动作可以表示为一个多维向量 $A_t = \{x_{1,1}, \ldots, x_{i,j}, \ldots, x_{n,m}, y_{1,1}, \ldots, y_{i,j}, \ldots, y_{n,m}\}$, 其中 $x_{i,j} = 1$ $(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$ 表示第 i 个控制器节点放置在了第 j 个交换机节点上, 否则 $x_{i,j} = 0$. 此外, $y_{i,j} = 1$ $(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$ 表示第 j 个交换机节点分配给了第 i 个控制器节点, 否则 $y_{i,j} = 0$. 根据前述定义, 动作空间将包括 2^{2nm} 个不同的动作, 但其中绝大多数动作都是不可行动作, 因为它们不满足实际的网络约束, 这部分内容将在第 4 节展开详细的介绍.

(3) 状态转移 (state transitions). 根据网络状态的定义, 交换机节点运行状态的变化将被视为状态转移. 例如, 如果交换机节点关联的控制器节点遭到了 DDoS 攻击, 造成控制器节点无法正常响应来自交换机节点的数据包询问请求, 那么该交换机节点的运行状态将从 1 转移至 0.

(4) 奖励 (reward). 在网络管理者选择多控制器放置方案后, 环境信息将会给予反馈, 因此需要构建合适的奖励函数来对被选择的动作进行评价. 考虑终端节点对控制器节点持续性发动 DDoS 攻击的情况下, 选择不同的控制器与交换机映射关系将会对网络性能产生不同程度的影响, 主要考虑了两个方面的因素, 一是遭到 DDoS 攻击的控制器所关联的交换机节点数量, 二是无法正常提供数据转发功能的交换机在网络拓扑中的重要程度, 用节点介数表示, 交换机节点的介数越大, 代表其在网络连通性方面越重要. 综上所述, 单个控制器的奖励函数被定义为

$$R_i = \begin{cases} \displaystyle\sum_{v_j \in Q} \sum_{v_k, v_l \in N} -\xi \times \frac{f_{k,l}(v_j)}{f_{k,l}}, & \text{如果控制器遭到 DDoS 攻击,} \\ C^+, & \text{其他,} \end{cases} \tag{1}$$

其中, Q 表示与遭到 DDoS 攻击的控制器关联的交换机节点集合 (在每个时隙网络管理者通过被攻击的控制器集合以及控制器和交换机的映射关系获知 Q), ξ 表示正系数, $f_{k,l}$ 表示交换机节点 v_k 与节点

v_l 的最短路径的个数, $f_{k,l}(v_j)$ 表示交换机节点 v_k 与节点 v_l 之间的最短路径中经过交换机节点 v_j 的个数, C^+ 表示正数. 上述奖励函数说明当控制器节点遭到 DDoS 攻击时, 单个控制器的奖励是由其管理的交换机节点的节点介数之和所决定, 遭受攻击的控制器管理的交换机节点越多, 获得奖励值越低. 在每个时刻, 总的奖励即为所有控制器的奖励之和. 值得注意的是, 动作被定义为整体上选择控制器部署位置及控制器和交换机之间的映射关系, 并非每个控制器单独选择, 因此奖励由网络管理者统计所有的控制器情况进行计算.

4 多控制器与交换机映射关系约束建模

3.3 小节指明了 MDP 的动作空间内仍然存在很多不可行动作, 因此本节将详细介绍如何根据实际网络约束对多个控制器与交换机的映射关系进行建模从而移除不可行动作, 通常此类问题被称为多控制器放置问题[5]. 不同于传统多控制器放置问题为最优化问题, 本节利用可满足性模理论 (satisfiability modulo theories, SMT) [30] 构建为约束满足问题, 该问题的可行解即为 MDP 模型中的可行动作. 多控制器放置问题的本质就是在部署多控制器的算力网络中确定控制器的放置位置及与交换机的映射关系, 从而满足所要考虑的多种实际网络约束.

在本文中, $x_{i,j}=1$ 表示第 i 个控制器节点放置在第 j 个交换机节点上, 否则 $x_{i,j}=0$. $y_{i,j}=1$ 表示第 j 个交换机节点分配给第 i 个控制器节点, 否则 $x_{i,j}=1$. 此外, 用 $D(i,j)$ 表示第 i 个交换机节点与第 j 个交换机节点的最短路由跳数. 首先, 多控制器与交换机的映射关系需要满足基本的分配约束, 定义为

$$\sum_{j=1}^{n} x_{i,j}=1,\ \forall c_i\in M, \tag{2}$$

$$\sum_{i=1}^{m} y_{i,j}=1,\ \forall v_j\in N, \tag{3}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} x_{i,j}=m, \tag{4}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j}=n, \tag{5}$$

其中, 式 (2) 保证了任何一个控制器节点只能放置在一个交换机节点上. 式 (3) 保证了任何一个交换机节点只能分配给唯一的控制器节点. 式 (4) 保证了当前网络中控制器节点的总数量为 m. 式 (5) 保证了当前网络中的交换机节点总数量为 n.

其次, 因为交换机与控制器均具有各自处理数据流的最大容量, 多控制器与交换机的映射关系需要满足处理容量约束. 因此, 容量约束被定义为

$$\sum_{j=1}^{m} h_j y_{i,j}\leqslant H_i,\ \forall c_i\in M, \tag{6}$$

其中, h_j 表示第 j 个交换机节点所能处理最大数量的数据包, H_i 表示第 i 个控制器节点所能处理最大数量的数据包. 式 (6) 保证了分配给同一个控制器节点的交换机节点的处理容量总和不超过相应控制器节点的最大处理容量.

此外, 多控制器与交换机的映射关系还需要满足网络延迟约束. 网络延迟主要包括交换机到控制器间的传输时延及控制器的处理时延. 那么, 延迟约束可以被定义为

$$\sum_{j=1}^{n} y_{i,j} \sum_{k=1}^{n} D(j,k) x_{i,k} d_l + \sum_{j=1}^{n} y_{i,j} d_p \leqslant T_i,\ \forall c_i \in M, \tag{7}$$

其中, d_l 表示单位路由跳数的传输时延, d_p 表示单位负载的处理时延, T_i 表示时延阈值. 式 (7) 保证了所有分配给第 i 个控制器节点的交换机节点的网络时延总和小于时延阈值.

最后, 因为多控制器之间需要同步每个网络域的控制信息, 多控制器与交换机的映射关系需要满足控制器之间的通信能耗约束, 该约束可以被定义为

$$\sum_{h=1}^{m} \sum_{k=1}^{n} \sum_{j=1}^{n} x_{i,j} x_{h,k} D(j,k) e \leqslant E,\ \forall c_i \in M, \tag{8}$$

其中, e 表示控制器跨域通信时单位跳数的通信能耗, E 表示通信能耗阈值. 式 (8) 保证了任何控制器节点与其他控制器节点的通信能耗总和不超过能耗阈值.

5 基于深度强化学习的主动防御算法

本节首先介绍深度强化学习的基本概念, 其次详细说明所提出基于深度强化学习的主动防御算法.

根据在 3.3 小节中构建的 MDP 模型的奖励函数, 网络管理者在每次选择动作后都会获得即时回报, 其不断地迭代优化动作选择策略, 最终目标是累积长期的收益回报. 那么, 状态价值函数可以被定义为

$$V^{\pi}(S_t) = \mathrm{E}_{\pi}\left[R_t + \sum_{\mu=1}^{\infty} \gamma^{\mu} R_{t+\mu}\right], \tag{9}$$

其中, π 表示动作选择策略, E 表示执行期望运算, γ 表示折扣因子, 则最优动作选择策略 π^* 定义为

$$V^{\pi^*}(S_t) = \max_{\pi} V^{\pi}(S_t), \tag{10}$$

依据式 (9) 与 (10), 当状态价值函数可以通过理论计算时, 那么完全遍历所有的网络状态, 则可以获得最优动作选择策略. 然而, 因为攻击者控制的僵尸主机在网络中的分布位置是未知的, 状态转移轨迹难以通过理论方法进行追踪, 从而无法获得准确的状态价值函数. 在这种情况下, 强化学习是广泛用来获取最优动作选择策略的方法, 但随着状态空间或动作空间的规模扩大, 传统强化学习例如 Q-learning 算法 [31] 难以应对维度爆炸的挑战. 为了处理大规模状态空间或动作空间, 深度强化学习利用深度神经网络近似表征策略或价值函数. 目前主流的深度强化学习算法主要分为基于值与基于策略的两类. 基于值的深度强化学习例如 DQN (deep Q-network) 算法 [32] 利用深度神经网络近似表征值函数, 能够处理大规模状态空间, 但不能够处理大规模动作空间. 为了弥补前述不足, 基于策略的深度强化学习算法利用深度神经网络近似参数化策略. 考虑到本文所构建的 MDP 模型中的状态空间与动作空间均随着交换机与控制器的数量增加而指数增长, 因此需采用基于策略的深度强化学习算法. 近年来, PPO 算法 [33] 作为基于策略的深度强化学习被广泛使用, 其中 actor 网络的损失函数为

$$L(\theta) = \mathrm{E}_t\left[\hat{A}_t \min(r_t(\theta), \mathrm{clip}(r_t(\theta), 1-\delta, 1+\delta))\right], \tag{11}$$

其中, $\hat{A}_t$ 表示在时刻 t 的优势函数, $r_t(\theta)$ 表示新旧策略的概率比例, δ 是控制裁剪函数裁剪范围的超参数, 优势函数与概率比例分别定义为

$$r_t(\theta) = \frac{\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta_{\text{old}})}, \tag{12}$$

$$\hat{A}_t(S_t,A_t) = \sum_{q=0}^{\infty}(\gamma\lambda)^q\delta_{t+q}, \tag{13}$$

$$\delta_t = R_t + \gamma V(S_{t+1},\phi) - V(S_t,\phi), \tag{14}$$

其中, γ 表示值在 0 和 1 之间的折扣因子, λ 用来平衡方差和偏差, $V(S_t,\phi)$ 表示状态价值函数的近似, 则状态价值目标为 $\hat{V}_t(S_t) = \hat{A}_t(S_t,A_t) + V(S_t,\phi)$, 因此 critic 网络的损失函数为

$$L(\phi) = \mathrm{E}_t\big(\hat{V}_t(S_t) - V(S_t,\phi)\big)^2, \tag{15}$$

最后根据文献 [34], 利用 Adam 优化器通过 $\nabla_\theta L(\theta)$ 和 $\nabla_\phi L(\phi)$ 分别更新 actor 网络的参数 θ 和 critic 网络的参数 ϕ. 在本文中, actor 和 critic 网络均设置为拥有 4 个全连接层的深度神经网络, 其中输入层维度为状态空间的维度, 第 2 和 3 层均为隐藏层, 隐藏层神经元的数量将在第 6 节实验参数设置部分给出. actor 网络的输出层维度是动作空间的维度, 而 critic 网络的输出维度是 1, 表示对当前状态的价值估计, 每个全连接层之间都使用 tanh 作为激活函数. 此外, 在 actor 网络的输出层后面添加 softmax 函数, 将输出转化为概率分布. 具体的主动防御算法如算法 1 所示.

算法 1 基于 PPO 的主动防御算法

输入: 参数 α,γ, batch 大小 T, minibatch 大小 K;

1: 初始化经验回放池 Φ;
2: 初始化 critic 网络 $V(S_t,\phi)$;
3: 初始化 actor 网络 π_θ;
4: **for** $q = 1$ to Q **do**
5: **for** $k = 1$ to K **do**
6: **for** $t = 1$ to T **do**
7: 观测当前网络状态 S_t;
8: 根据策略选择动作 A_t, 执行交换机与控制器的映射重分配;
9: 获得即时奖励 R_t 及观测下一个状态 S_{t+1};
10: 将样本 (S_t,A_t,R_t,S_{t+1}) 存入经验回放池 Φ, 计算优势函数及动作价值函数;
11: **end for**
12: **end for**
13: **for** $u = 1$ to U **do**
14: 根据式 (11) 计算梯度并更新 actor 网络的参数 θ;
15: 根据式 (15) 计算梯度并更新 critic 网络的参数 ϕ;
16: **end for**
17: $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$;
18: **end for**

输出: 防御选择策略 π.

算法 1 最开始初始化经验回放池 Φ, critic 及 actor 网络 (第 1~3 行). 在样本获取阶段, 首先观测当前网络状态, 执行控制器放置并重分配交换机与控制器的映射关系, 获得 DDoS 攻击影响的即时奖励及观测下一个网络状态, 最后将样本存入经验回放池并计算优势函数及动作价值函数 (第 5~12 行).

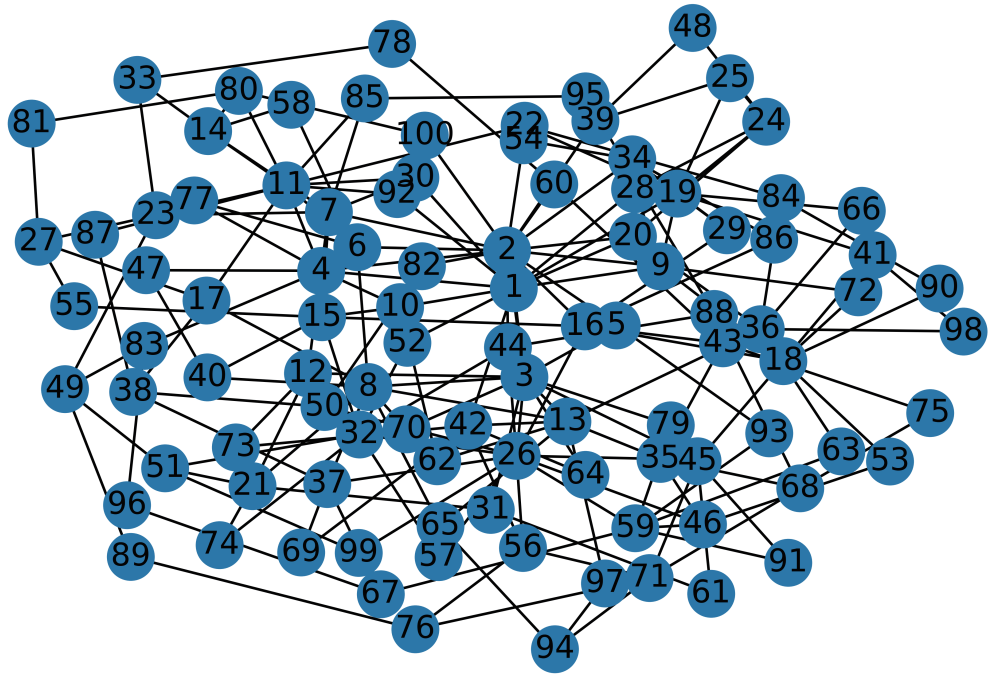

图 2　(网络版彩图) 100 个 OpenFlow 交换机的网络拓扑图

Figure 2　(Color online) Network topology with 100 OpenFlow switches

在参数更新阶段, 根据式 (11) 分别计算梯度并更新 actor 与 critic 网络 (第 13~16 行). 最后, 更新动作选择策略 (第 17 行).

6　实验与结果

本文利用 Mininet-WiFi 网络仿真器 [35] 构建了一个分布式 SDN 网络, 并使用基于 Python 的开源 Ryu 作为 SDN 控制器. 此外, 利用 Networkx 库根据 Waxman 模型 [36] 生成 100 个 OpenFlow 交换机节点的网络拓扑, 其拓扑图如图 2 所示. 攻击者所控制的僵尸主机具有固定数量, 但僵尸主机在 SDN 网络中的分布位置在每次实验开始前随机设置. 在 PPO 中, actor 与 critic 网络均通过 Pytorch 框架进行训练. 上述实验均在具有 Nividia GeForce RTX 5000 GPU 的工作站 (Intel(R) Core(TM) i9-10920X CPU @3.5 GHz, 64 GB RAM) 上运行. 本文实验具体参数如表 1 所示.

6.1　约束求解性能

第 4 节构建了一个多控制器部署的约束满足问题, 已经被证明是 NP 完全问题 [10]. 我们利用微软最新提出的理论求解器 Z3 solver [37] 求解前述约束满足问题, 其能够解决计算数万个约束和数百万个变量. 这种情况下, 求解时间被认为是重要的评估度量. 在 OpenFlow 交换机的数量设置为 50, 60, 70, 80, 90, 100, 并将 SDN 控制器的数量设置为 $10, 15, 20$, 利用 Z3 slover 计算获得 500 个可行解的求解时间.

如图 3 所示, SMT 求解时间随着 OpenFlow 交换机和 SDN 控制器的数量上升而显著上升. 当交换机数量为 100, 控制器数量为 20 时, SMT 求解时间达到了 519 s, 原因是交换机和控制器数量的增加会使约束和变量的数量同样增加, 从而造成 Z3 solver 的求解时间同样增长. 此外, 从图中可以发现,

表 1 实验参数设置

Table 1 Simulation parameters

Parameter	Value or range	Parameter	Value or range
Parameters of Waxman model	$\alpha = 0.2, \beta = 0.15$	Capacity of switches	50000 PIMs [27]
Number of OpenFlow switches	$[50, 60, 70, 80, 90, 100]$	Capacity of controllers	5000 PIMs
Number of controllers	$[10, 15, 20]$	Delay threshold	40 ms
Communication range of switches	250 m	Energy threshold	100 J
Bandwidth of links	5 Mbps	Number of episodes	1000 episodes
Number of bot hosts	10	Coefficient of clip δ	0.2
Actor network hidden layers	$[256, 256]$	Critic network hidden layers	$[256, 256]$

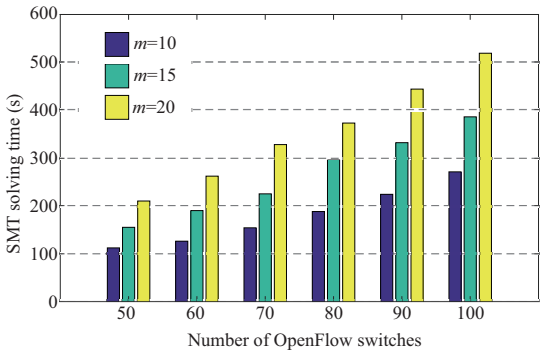

图 3 (网络版彩图) 不同数量交换机与控制器 SMT 求解时间对比

Figure 3 (Color online) SMT solving time with different number of switches and controllers

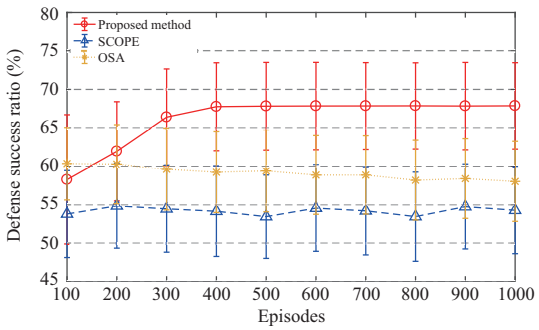

图 4 (网络版彩图) 本文所提方法与基准方法的防御成功率对比

Figure 4 (Color online) Defense success ratio comparison between the proposed methods, SCOPE and OSA

控制器数量增长造成 SMT 求解时间增长趋势高于交换机数量增长造成 SMT 求解时间增长趋势. 这种现象是因为控制器数量增加引发的约束数量增加是高于交换机数量增加引发的约束数量增加.

6.2 防御性能

防御性能是衡量本文所提主动防御方法的重要方面. 采用防御成功率作为评估指标, 其定义为

$$\text{防御成功率} = (1 - n_{\text{attack}}/n_{\text{total}}) \times 100\%, \tag{16}$$

其中, n_{attack} 表示受 DDoS 攻击影响而无法正常运行的 OpenFlow 交换机数量, n_{total} 表示 OpenFlow 交换机的总数量. 我们将所提主动防御方法与多控制器部署的基准方法 (SCOPE) [17] 以及混合主动防御机制 (optimal strategy selection algorithm, OSA) [22] 进行防御性能对比. 3 种算法分别进行 5 轮实验, 并求得平均值与均方差. 此外, 针对本文考虑的多控制器部署场景, 我们对 SCOPE 方法与 OSA 方法分别进行适当的调整与修改, 并且为了充分测试对比算法的防御性能, 在每个 episode 重新执行 DDoS 攻击并计算防御成功率.

如图 4 所示, SCOPE 方法的防御成功率为 55%, OSA 方法的防御成功率为 60%, 这两种算法的防御成功率随着 episode 的增加基本保持不变, 并且均方差基本固定, 说明各自防御成功率都在 5% 左右的范围内波动. 另一方面, 本文所提方法的防御成功率初始为 58%, 随着 episode 的增加而逐渐上升,

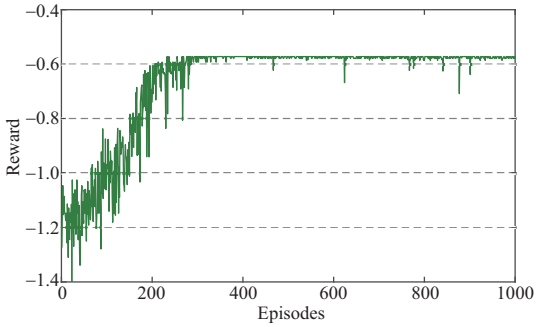

图 5 (网络版彩图) 本文所提方法的收敛性能

Figure 5 (Color online) Convergence performance of the proposed method

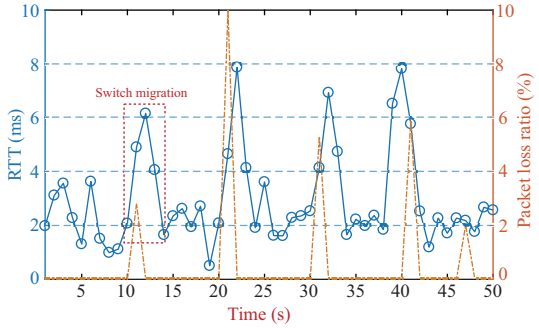

图 6 (网络版彩图) 本文所提方法的网络性能

Figure 6 (Color online) Network performance of the proposed method

直至收敛至 68%, 并且本文所提方法的防御成功率在 8% 左右的范围内波动. 原因是本文所提方法基于深度强化学习, 能够根据 DDoS 攻击对多控制器部署方案造成影响的反馈不断优化动作选择策略, 最终选择最优的多控制器部署方案, 将 DDoS 攻击的负面影响降至最低. 因此, 多次实验结果表明, 本文所提方法的防御成功率与 SCOPE 方法和 OSA 方法相比, 分别提升了大约 13% 与 8% 的防御成功率.

6.3 收敛性能

我们收集算法训练过程中的奖励数据来评估本文所提方法的收敛性, 同样进行了 5 轮实验, 每轮实验共 1000 个 episode, 并求多次实验的平均值.

如图 5 所示, 奖励数据从最开始的 −1.2 随着 episode 的增加逐渐上升, 最终收敛至 −0.6. 此外, 所提方法大约在 300 个 episode 后收敛, 与 0~300 个 episode 训练阶段一直波动不同的是, 在收敛后奖励数据相对平稳, 不再产生剧烈的波动. 多次实验结果表明本文方法具有良好的收敛性能.

6.4 网络性能

周期性地调整控制器的部署位置及交换机与控制器之间的映射关系将会对网络性能造成影响. 因此, 本文采用往返时间 (round trip time, RTT) 和丢包率来评估本文所提方法对网络性能造成影响的程度. 我们进行 50 s 的实验, 并将映射关系的调整周期设置为 10 s, 同样进行了 5 轮实验, RTT 与丢包率均为多次实验的平均值.

如图 6 所示, 当不改变交换机与控制器之间的映射关系时, RTT 大约在 2 ms, 丢包率为 0%. 当为了调整映射关系而发生交换机迁移时, RTT 将会增加至 6~8 ms, 丢包率将会增加至 2%~10%. 此外, 红框表示交换机迁移阶段, 根据实验结果可以发现, 交换机迁移对网络性能造成的负面影响将会持续 1~2 s, 此后 RTT 与丢包率均会恢复至正常. 因此, 多次实验表明, 本文所提的方法仅会对网络性能造成轻微影响.

7 总结

本文提出了一种基于深度强化学习的算力网络主动防御方法. 首先, 将控制器部署位置及交换机与控制器之间映射关系的动态性用 MDP 模型表示. 其次, 考虑了多种网络约束, 构建了一个约束满足

问题, 将不可行的多控制器放置方案从 MDP 模型的动作空间中移除. 最后, 提出了一种基于 PPO 的主动防御算法, 能够根据 DDoS 攻击对交换机造成影响的范围, 智能化地调整多控制器部署方案, 将 DDoS 攻击对分布式 SDN 网络造成的破坏效果降至最低.

未来的工作包括 3 个方面, 第 1 个方面是如何在真实分布式 SDN 中部署本文所提出的主动防御方法. 第 2 个方面是如何追溯 DDoS 攻击的源头, 从而阻断 DDoS 攻击的持续性. 第 3 个方面是未来将进一步探索如何依据算力网络主动防御场景优化深度强化学习的神经网络结构.

参考文献

1 Ericsson. Why IoT changes everything. 2022. https://www.ericsson.com/

2 Yao Z X, Xia S C, Li Y. Task offloading and resource allocation in an uncertain network. Sci Sin Inform, 2022, 52: 1349–1361 [姚枝秀, 夏士超, 李云. 不确定网络环境下的任务卸载和资源分配算法. 中国科学: 信息科学, 2022, 52: 1349–1361]

3 Lv P, Liu Q R, Wu J X, et al. New generation software-defined architecture. Sci Sin Inform, 2018, 48: 315–328 [吕平, 刘勤让, 邬江兴, 等. 新一代软件定义体系结构. 中国科学: 信息科学, 2018, 48: 315–328]

4 Jia Q M, Hu Y J, Zhang H Y, et al. Research on deterministic computing power network. J Commun, 2022, 43: 55–64 [贾庆民, 胡玉姣, 张华宇, 等. 确定性算力网络研究. 通信学报, 2022, 43: 55–64]

5 Das T, Sridharan V, Gurusamy M. A survey on controller placement in SDN. IEEE Commun Surv Tut, 2020, 22: 472–503

6 Kalkan K, Altay L, Gur G, et al. JESS: joint entropy-based DDoS defense scheme in SDN. IEEE J Sel Areas Commun, 2018, 36: 2358–2372

7 Zhang L, Wang J S. DDoS attack detection model based on information entropy and DNN in SDN. J Comput Res Dev, 2019, 56: 909–918 [张龙, 王劲松. SDN 中基于信息熵与 DNN 的 DDoS 攻击检测模型. 计算机研究与发展, 2019, 56: 909–918]

8 Cho J H, Sharma D P, Alavizadeh H, et al. Toward proactive, adaptive defense: a survey on moving target defense. IEEE Commun Surv Tut, 2020, 22: 709–745

9 Zhang T, Xu C Q, Zhang B C, et al. Toward attack-resistant route mutation for VANETs: an online and adaptive multiagent reinforcement learning approach. IEEE Trans Intell Transp Syst, 2022, 23: 23254–23267

10 Jafarian J H, Al-Shaer E, Duan Q. An effective address mutation approach for disrupting reconnaissance attacks. IEEE Trans Inform Forensic Secur, 2015, 10: 2562–2577

11 Zhang T, Xu C Q, Zhang B C, et al. Towards attack-resistant service function chain migration: a model-based adaptive proximal policy optimization approach. IEEE Trans Depend Secure Comput, 2023, 20: 4913–4927

12 Lu X Z, Xiao L, Li P M, et al. Reinforcement learning-based physical cross-layer security and privacy in 6G. IEEE Commun Surv Tut, 2023, 25: 425–466

13 Yu Q, Ren J, Fu Y J, et al. Cybertwin: an origin of next generation network architecture. IEEE Wireless Commun, 2019, 26: 111–117

14 Di Z, Luo T, Qiu C, et al. In-network pooling: contribution-aware allocation optimization for computing power network in B5G/6G Era. IEEE Trans Netw Sci Eng, 2023, 10: 1190–1202

15 Wang X F, Ren X X, Qiu C, et al. Net-in-AI: a computing-power networking framework with adaptability, flexibility, and profitability for ubiquitous AI. IEEE Netw, 2021, 35: 280–288

16 Liu Z N, Li K, Wu L T, et al. CATS: cost aware task scheduling in multi-tier computing network. J Comput Res Dev, 2020, 57: 1810–1822 [刘泽宁, 李凯, 吴连涛, 等. 多层次算力网络中代价感知任务调度算法. 计算机研究与发展, 2020, 57: 1810–1822]

17 Maity I, Misra S, Mandal C. SCOPE: cost-efficient QoS-aware switch and controller placement in hybrid SDN. IEEE Syst J, 2022, 16: 4873–4880

18 Yi L Z, Yin M, Darbandi M. A deep and systematic review of the intrusion detection systems in the fog environment. Trans Emerg Tel Tech, 2023, 34: e4632

19 Zainudin A, Ahakonye L A C, Akter R, et al. An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks. IEEE Int Things J, 2023, 10: 8491–8504

20 Ravi N, Shalinie S M, Lal C, et al. AEGIS: detection and mitigation of TCP SYN flood on SDN controller. IEEE Trans Netw Serv Manage, 2021, 18: 745–759

21 Jagtap M M, Saravanan R D. Intelligent software defined networking: long short term memory-graded rated unit enabled block-attack model to tackle distributed denial of service attacks. Trans Emerging Tel Tech, 2022, 33: e4594

22 Zhou Y Y, Cheng G, Yu S. An SDN-enabled proactive defense framework for DDoS mitigation in IoT networks. IEEE Trans Inform Forensic Secur, 2021, 16: 5366–5380

23 Javadpour A, Ja'fari F, Taleb T, et al. SCEMA: an SDN-oriented cost-effective edge-based MTD approach. IEEE Trans Inform Forensic Secur, 2023, 18: 667–682

24 Xu C Q, Zhang T, Kuang X H, et al. Context-aware adaptive route mutation scheme: a reinforcement learning approach. IEEE Int Things J, 2021, 8: 13528–13541

25 Zhang T, Xu C Q, Zou P, et al. How to mitigate DDoS intelligently in SD-IoV: a moving target defense approach. IEEE Trans Ind Inf, 2023, 19: 1097–1106

26 Xu Y, Cello M, Wang I C, et al. Dynamic switch migration in distributed software-defined networks to achieve controller load balance. IEEE J Sel Areas Commun, 2019, 37: 515–529

27 Lai W K, Wang Y C, Chen Y C, et al. TSSM: time-sharing switch migration to balance loads of distributed SDN controllers. IEEE Trans Netw Serv Manage, 2022, 19: 1585–1597

28 Sahoo K S, Puthal D, Tiwary M, et al. ESMLB: efficient switch migration-based load balancing for multicontroller SDN in IoT. IEEE Int Things J, 2020, 7: 5852–5860

29 Chen K Y, Liu S, Xu Y, et al. SDNShield: NFV-based defense framework against DDoS attacks on SDN control plane. IEEE ACM Trans Netw, 2022, 30: 1–17

30 de Moura L, Bjørner N. Satisfiability modulo theories: introduction and applications. Commun ACM, 2011, 54: 69–77

31 Jang B, Kim M, Harerimana G, et al. Q-learning algorithms: a comprehensive classification and applications. IEEE Access, 2019, 7: 133653

32 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533

33 Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. 2017. ArXiv:1707.06347

34 Guan Y, Ren Y, Li S E, et al. Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization. IEEE Trans Veh Technol, 2020, 69: 12597–12608

35 Fontes R R, Afzal S, Brito S H B, et al. Mininet-WiFi: emulating software-defined wireless networks. In: Proceedings of International Conference on Network and Service Management (CNSM), 2015. 384–389

36 Waxman B M. Routing of multipoint connections. IEEE J Sel Areas Commun, 1988, 6: 1617–1622

37 de Moura L, Bjørner N. Z3: an efficient SMT solver. In: Proceedings of International conference on Tools and Algorithms for the Construction and Analysis of Systems, 2008. 337–340

Deep reinforcement learning-based moving target defense method in computing power network

Tao ZHANG[1,2,5,6], Changqiao XU[1,2*], Yibo LIAN[1,2], Jiawen KANG[3] & Xiaohui KUANG[1,2,4]

1. *College of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing* 100876*, China*;
2. *State Key Laboratory of Networking and Switching Technology, Beijing* 100876*, China*;
3. *College of Automation, Guangdong University of Technology, Guangzhou* 510062*, China*;
4. *National Key Laboratory of Science and Technology on Information System Security, Beijing* 100101*, China*;
5. *School of Software Engineering, Beijing Jiaotong University, Beijing* 100044*, China*;
6. *Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing* 100044*, China*
* Corresponding author. E-mail: cqxu@bupt.edu.cn

Abstract Computing power networks aim to deeply integrate computing resources and network resources to obtain efficient collaboration of multiple resources and maximize resource utilization. The edge of computing power networks usually adopts the framework of a distributed software-defined network (SDN), in which the control plane is logically centralized and physically distributed, and separated from the data plane to unify the scheduling and orchestration of computing and network resources. However, the attacker regards the control plane as the target and launches distributed denial of service (DDoS) attacks, making the control plane fail in a large area and severely affecting the real-time transmission of computing tasks. To solve the security problem in computing power networks, this paper proposes a novel moving target defense method based on deep reinforcement learning. First, a Markov decision process (MDP) model is formulated to accurately represent the dynamic mapping relationship between switches and controllers, and a reward function based on betweenness is designed to reflect the impact of DDoS attacks on the control plane. Second, considering multiple network constraints, the multiple controller placement problem is modeled as a constrained satisfaction problem, and feasible solutions are considered the action space of the MDP. Finally, an active defense algorithm based on deep reinforcement learning is designed to iteratively optimize the selection strategy of actions and intelligently select the deployment of multiple controllers. The experimental results show that compared with baseline methods, our method can improve the defense success ratio by approximately 13% and 8% while slightly affecting network performance.

Keywords computing power network, distributed software-defined network (SDN), moving target defense, distributed denial of service (DDoS), deep reinforcement learning