基于交换机迁移的控制平面饱和攻击防御方法

管吉喆[1)] 程 光[1),2)] 周余阳[1)]
[1)](东南大学网络空间安全学院 南京 211189)
[2)](紫金山实验室 南京 211111)

摘 要 软件定义网络凭借着自身灵活的优势,被广泛应用在各类网络场景,但由于控制平面控制集中、资源有限,控制平面饱和攻击成为该网络中最大的安全威胁.控制平面饱和攻击作为DDoS攻击的特殊实践,其目标是消耗控制器的处理资源,影响控制器下发流规则,最终使网络瘫痪.与传统网络中攻击主要影响单一网络结点不同,在软件定义网络中,一旦控制器遭受攻击,子域下交换机都将失去工作能力,甚至引发更大范围的级联故障.近年来,学者们为了提高分布式软件定义网络的安全性作出了很多创新,但大多数研究采用弹性扩容和动态映射的方法,不断地增删设备和改变网络映射配置,在提高了防御成本的同时,还降低了网络的可靠性,且复杂的网络配置违背了软件定义网络的初衷,为网络管理增加了难度.本文针对控制平面遭受的饱和攻击,提出了一种基于交换机迁移的防御方法.首先,通过分析软件定义网络中控制器的负载来源,提出了由北向开销、南向开销和水平开销组成的计算负载,并利用阈值检测机制,监测控制器负载情况,实现了对控制平面的实时饱和检测.然后,综合流表统计信息和空间特征图,提出了与平均数据包、上下行流量比率和交换机重要程度相关联的迁移系数,以准确识别处于攻击路径上的交换机.最后,在交换机迁移策略的支持下,将其迁移至低载控制器子域当中,缓解受攻击控制器的单点故障问题,从而完成对攻击的防御效果.在实验验证阶段,本文首先证明了控制平面饱和攻击的攻防过程实质上是一场资源竞争博弈,可以通过资源的调配缓解攻击影响.然后,本文证明了所提方法所具有的灵活性和动态性,能够根据资源情况和网络情况动态选择目标,实现迁移操作.最后通过多组实验,本文证明了所提出的方法能够有效地缓解不同拓扑的控制平面饱和攻击,控制器饱和次数平均减少90%和65%,迁移目标首选率和单次迁移率都超过70%,有效避免了级联故障的产生,与其他方法相比,在64%左右的时间里负载标准差最低,迁移总次数最少降低11%,且迁移时间开销最少降低60%.

关键词 软件定义网络;控制平面;控制平面饱和攻击;主动防御;交换机迁移
中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2024.02889

Control Plane Saturation Attack Defense Method Based on Switch Migration

GUAN Ji-Zhe[1)] CHENG Guang[1),2)] ZHOU Yu-Yang[1)]
[1)](*School of Cyber Science and Engineering*, *Southeast University*, *Nanjing* 211189)
[2)](*Purple Mountain Laboratories*, *Nanjing* 211111)

Abstract Software defined network, with flexible advantages, is widely applied in various network scenarios. However, due to the centralized control and limited resources of the control plane, control plane saturation attacks have become the biggest security threat in this network. Control plane saturation attacks as the special approach to DDoS attacks, aim to consume the computing resource of the controller, affect the flow rules issued by the controller, and ultimately

收稿日期:2024-01-16;在线发布日期:2024-09-14. 本课题得到国家自然科学基金(62172093、62202097、U22B2025)、中国博士后科学基金资助项目(2024T170143、2022M710677)、江苏省卓越博士后计划(2022ZB137)资助.**管吉喆**,硕士研究生,中国计算机学会(CCF)学生会员,主要研究领域为软件定义网络、DDoS缓解和主动防御.E-mail:15864201808@163.com. **程 光**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)杰出会员,主要研究领域为网络安全、网络测量和流量行为分析.E-mail:chengguang@seu.edu.cn. **周余阳**,博士,助理研究员,中国计算机学会(CCF)会员,主要研究领域为移动目标防御、DDoS缓解、入侵检测和安卓恶意软件检测.

achieve network paralysis. Unlike traditional networks, DDoS attacks mainly affect a single network node. In software defined networks, once a controller is attacked, the switches in the subdomain will lose their ability to function, and even trigger larger cascading failures. In recent years, scholars have made many innovations to improve the security of distributed software defined network. However, most of the research adopts elastic expansion and dynamic mapping methods, constantly adding and deleting devices or changing network mapping configurations, which increases defense costs and reduces network reliability. Moreover, complex network configurations violate the original intention of software defined networks and increase the difficulty of network management. This article proposes a defense method based on switch migration for control plane saturation attacks. First, we meticulously analyze the sources of load on controllers within software-defined networks to devise a comprehensive computational load model. This model encapsulates northbound, southbound, and horizontal overheads, providing a holistic view of controller load. By employing a sophisticated threshold detection mechanism, we continuously monitor the controller load, enabling real-time saturation detection within the control plane. Next, we integrate flow table statistics with spatial feature maps to develop an innovative migration coefficient. This coefficient is intricately tied to factors such as average packet counts, up and down traffic ratios, and the criticality of individual switches. This approach allows us to accurately pinpoint switches that lie on potential attack paths. Finally, leveraging a strategic switch migration policy, we swiftly and efficiently relocate these identified switches to subdomains governed by controllers with lower loads. This migration not only alleviates the risk of single-point-of-failure issues in compromised controllers but also significantly enhances the overall resilience and defense capabilities against attacks. In the experimental validation phase, this paper conclusively demonstrates that the offensive and defensive dynamics of a control plane saturation attack fundamentally constitute a resource competition game. By strategically allocating resources, the impact of such attacks can be effectively mitigated. Moreover, the proposed method showcases remarkable flexibility and adaptability, enabling it to dynamically select targets and execute migration operations in response to real-time resource and network conditions. Through multiple sets of experiments, this paper proves that the proposed method can effectively alleviate control plane saturation attacks of different topologies. The average number of controller saturation decreases by 90% and 65%, respectively. Moreover, the preferred migration target rate and single migration rate both exceed 70%, effectively avoiding cascading failures. Compared with other methods, the load standard deviation is the lowest in about 64% of the time, the total number of migrations has been reduced by at least 11%, and the migration time cost is at least 60% lower.

Keywords software defined network; control plane; control plane saturation attacks; proactive defense; switch migration

1 引 言

软件定义网络(Software Defined Network, SDN)[1],凭借着其平面分离的特点和开放可编程的优势,实现了网络流量的灵活控制和网络设备的灵活部署.随着SDN网络规模的扩展和流量的几何式增长,集中式的控制中心难以处理巨大的数据流量,容易引发单点故障[2].研究人员纷纷开展了对分布式控制器的研究,来解决集中式控制器的可扩展性和可用性问题,并提出了在逻辑上集中、物理上分布的多控制器部署策略.这种策略提高了SDN网络的处理上限,能够保证交换机得到快速响应,但仍然存在容量上限的问题,过量的处理任务依旧会导

致单点故障，且随着多控制器的引入，单点故障会随着交换机控制权的转移而传播，引发了更严重的级联故障[3]，攻击者能够通过对一个控制器结点发起攻击，达到影响整个网络的效果，这在传统网络中是很难发生的，所以保护控制器免受攻击侵扰是提高SDN网络安全性的首要任务[4].

分布式拒绝服务攻击(Distributed Denial of Service Attack，DDoS)作为网络中最常见的攻击方式，具有高度隐蔽性和破坏性，攻击者通过控制大规模僵尸主机或虚拟主机来攻击主机组，消耗目标主机的计算和存储资源[5]. 目前，许多研究[6-8]都致力于在SDN网络中防御DDoS攻击，他们借助SDN网络的全局视野提取流量特征，阻止恶意流量流向目标主机. 虽然这些研究都有效降低了DDoS攻击对SDN网络的侵害，但本质仍是对传统DDoS攻击的防御方法. 针对SDN架构本身，产生了一种新型的DDoS攻击，这种攻击被称为控制平面饱和攻击[9]，其目标在于消耗SDN控制器的CPU、内存和控制信道带宽等有限资源，最终导致网络性能下降[10]. 研究者们针对控制平面饱和攻击，从检测和缓解两方面开展研究. 攻击检测方法被划分为基于阈值的攻击检测方法[11-17]和基于流量特征的攻击检测方法[18-21]. 前者通过提取统计特征设立相关阈值来检测攻击，这种方法实施简单，能够在攻击早期就检测到攻击的发生，但此类方法，以攻击过程为检测依据，不区别流量类型，容易被网络的动态变化而影响. 后者能够区别流量类型，但通用性较差，最致命的一点，此类方法需要频繁提取和处理流量特征，占用控制器的大量资源，这种缺点在面对控制平面饱和攻击时会被放大若干倍. 由于控制平面饱和攻击的攻防本质在于资源竞争博弈，弹性扩容成为了缓解此类攻击的有效方法. 部分研究者[22-24]从CPU数量、控制器数量和控制链路带宽入手，根据网络状态，动态添加组件资源，解决性能上限，但此方法都需要引入外部资源，增加了缓解成本. 另有研究[25-26]期望在不使用外部资源的情况下，通过集群将设备组件组合在一起，变换控制器与交换机之间的映射关系，缓解饱和控制器的工作负载，达到降低控制层处理开销的目的，但这些研究仅仅只关注了控制层的负载情况，网络配置复杂，为网络管理增加了难度.

为了解决当前研究所存在的问题、保护SDN网络中的控制器不受侵害，本文设计了一种基于交换机迁移的控制平面饱和攻击防御方法. 该方法既不需要引入额外资源，又满足SDN网络灵活配置的特性，以控制器饱和状态为攻击检测依据，不为控制器引入大量检测开销，最大限度保存合法的突发流量，并通过感知攻击路径，准确均衡控制器负载，有效地缓解了攻击影响. 首先，计算控制器的负载，并设立饱和检测阈值，当负载超过检测阈值时，标记饱和控制器，系统转向饱和缓解阶段. 饱和缓解阶段的任务是将饱和控制器的交换机迁移到低载控制器下，分担饱和控制器的负载，从而缓解攻击，该阶段分为迁移目标选择和迁入域选择两个步骤. 迁移目标选择是从饱和控制器下选择迁出交换机的过程，我们根据交换机结点的上下行流量比率、平均数据包大小和交换机重要程度来感知攻击路径，评估交换机的迁移优先级. 依据待迁移交换机所能产生的负载和优先级，在辅助控制器集群中进行最坏适应选择，寻找满足级联故障约束条件并且具有最大接纳能力的辅助控制器，实现迁入域选择. 最后，由辅助控制器们接替待迁移交换机们的原主控制器角色，处理他们的消息请求. 本文的主要贡献和创新如下：

(1)本文提出了一种基于交换机迁移的控制平面饱和攻击防御方法，该方法将攻击防御与负载均衡相结合，能够在不检测攻击流量特征的情况下执行缓解动作，预先保障控制器安全，既能够缓解恶意流量带来的攻击危害，又能兼顾突发流量形成的负载不均衡，保证网络安全和均衡.

(2)本文所采用的阈值检测方法偏重于饱和状态检测，确保缓解操作的执行是基于控制器状态而发生的，减少缓解操作的执行频率，实现非饱和状态不执行防御的目标，从而尽最大可能保留突发流量.

(3)本文通过对流量和交换机的特征分析，感知位于攻击路径上的交换机节点，确保了迁移目标选择的准确性，并在最坏适应匹配的约束下，降低了级联故障的发生概率.

(4)通过进行仿真实验，我们验证了基于交换机迁移防御方法的防御效果，综合防御的灵活性、均衡性和有效性等评价指标，与现有的迁移策略进行对比，进一步证明了本方法能够有效地抵御控制平面饱和攻击，并且实施防御后，网络负载状况保持均衡稳定，且迁移时间开销更低.

本文第2节对SDN场景下关于控制平面饱和攻击的相关研究进行了总结和分析. 第3节和第4节对控制平面饱和攻击和交换机迁移防御架构进行了简要的介绍. 第5节详述了对基于交换机迁移的

控制平面饱和攻击防御方法在饱和检测和饱和缓解的实现细节.第6节展示了对本文方法的实验分析.最后,在第7节对本文进行了总结.

2 相关工作

DDoS攻击具有高度的隐蔽性和破坏性,许多研究都希望借助SDN的特点,实现对传统DDoS攻击的防御.Zhou等人[6]针对在物联网SDN环境中的DDoS攻击,基于信号博弈理论和移动目标防御技术,建立了一种混合主动防御机制,以传播伪装信息来迷惑攻击者.Zheng等人[7]和Cao等人[8]都基于流的特征来定位攻击路径,从而实施靠近攻击源点的防御措施.

虽然SDN技术为传统DDoS攻击的防御提供了新思路,但是由于SDN架构的集中控制,针对SDN架构的DDoS攻击对网络的影响能够由单点控制器扩展到整个网络面,范围更广,破坏力更强.部分研究人员也从分布式控制器切入,将网络划分为多个子域集合,各个子域由单独控制器进行控制.分布式控制器虽然解决了单点故障问题,但由于分布式控制器之间存在主从关系,任务负载可在主从控制器间转移,为软件定义网络引入了级联故障的问题.因此,SDN架构自身的安全性更应该被重视.目前,部分研究者开展了针对SDN架构的攻击防御,这些防御过程可总结为具有两阶段的过程:攻击检测和攻击缓解.

2.1 控制平面饱和攻击的攻击检测

当前,控制平面饱和攻击的攻击检测方法大多被划定为两类:基于阈值的攻击检测方法和基于流量特征的攻击检测方法.基于阈值的攻击检测方法通过提取统计特征来设立阈值,超过阈值的流量会被判断为恶意流量.Li[11]等人首次研究了OpenFlow流量的自相似度,并利用正常和异常流量的Hurst指数作为检测攻击的阈值.Mousavi[12]等人基于目标IP地址熵变化来检测攻击,这是首次在SDN控制器中使用熵进行攻击检测.Huang[13]等人使用双阈值来提高检测的准确性,当Packet_In消息速率超过阈值且控制信道利用率的熵值低于阈值时,可以确定攻击的存在.Rui Wang[14]等人提出了一种基于熵的统计检测方案,依赖五元组(源IP、目的IP、源端口、目的端口、协议)定义流的熵值,达到检测攻击的目的,但这种方法需要设置特定的收集器,不支持基础OpenFlow环境下的元素收集.除此之外,Kumar[15]、Wang[16]和Mishra[17]分别通过目标IP和TCP标志、源IP地址和目的IP地址、流表信息来获取熵,并设立阈值来区分恶意流量和正常流量.基于阈值的检测方法能够在攻击早期就检测到攻击,但此类方法大都偏重于攻击过程的检测,难以区别攻击流量和突发流量,突发流量会被当作恶意流量丢弃.

基于流量特征的攻击检测方法通过机器学习方法,对流量进行分类,从而识别攻击流量.Jin[18]等人基于流表统计数据,使用SVM算法进行分类.Xu[19]与Dong[20]等人选择使用改进的KNN算法进行检测,前者提出了一种基于K-means++和K近邻算法的检测方法,并提出了模块化检测系统,而后者使用流长度、流持续时间、流大小和流比率来分析系统是否遭遇攻击,计算攻击程度后使用基于KNN的机器学习方法对正常和恶意流量进行分类.Gao[21]等人基于FCM算法根据数据包和端口特征检测攻击,并基于贝叶斯网络将交换机分为不同角色,从而采取不同策略.基于流量特征的攻击检测方法具有较高的准确率,但大多数检测与攻击协议相关,不具备通用性,最重要的是,此类方法需要频繁提取和处理流量特征,占用大量控制器的计算和存储资源,这在面对控制平面饱和攻击时是极其危险的.

2.2 控制平面饱和攻击的攻击缓解

攻击缓解是依据检测结果,采取相应缓解方法,来降低攻击对网络性能的影响.异常流量的清洗和过滤方法往往结合基于流量特征的检测方法,将识别到的异常流量通过安装流表规则的方式丢弃或者缓存.主动防御往往配合基于阈值的检测方法,在攻击造成影响之前,完成对网络的保护.Yuan等人[22]利用M/M/c排队模型来衡量SDN网络的抗饱和能力并实施动态的控制器资源投资方法.Advait[23]等人也提出了一种称为ElastiCon的弹性分布式架构,根据流量条件动态增长或收缩控制器资源,并依据交换机到控制器的跳数、不同子域的负载等条件平衡控制器之间的负载,确保网络保持良好的性能.Dai[24]等人通过动态增加虚拟交换机结点来弹性增加SDN中控制路径吞吐量,以减少饱和攻击对控制信道的耗尽威胁.此方法虽然能够有效地降低控制信道的拥塞程度,但交换机数量的增加,会使得控制器的子域范围变大,反而影响网络的传输性能.控制平面饱和攻击的攻防本质在于资源竞争博弈[27],弹性扩容虽然从本质上对此类攻击提出了缓解方法,但需要引入额外的资源成本,且这些额外资

源的使用率偏低. 因此，合理调配已有资源，完成攻击缓解是一项巨大的挑战. 动态映射在不使用外部资源的情况下，通过集群将设备组件组合在一起，能够最大程度地减少攻击影响. Wang等人[25]提出了一种重新映射控制器和交换机之间连接关系的方法，该方法基于控制器的剩余资源、控制器与交换机之间的跳数和控制器之间的同步成本作为度量指标，均衡控制器与交换机之间的映射关系. Ricardo[26]等人提出了一种多控制器网络下的攻击缓解协议PATMOS，该协议制定了缓解选举机制，为控制器赋予领导者、精英等角色，当攻击发生时，改变控制器映射，来缓解控制器的工作负载. 事实上，控制器与交换机往往借助带内传输的方式发送控制信息，两者之间的动态映射容易增加控制信息的传输开销，导致网络性能下降.

针对现有研究在检测和缓解所存在的缺点，本文提出了以控制器阈值检测和交换机迁移为主要策略的防御方法. 在攻击检测上，侧重于对攻击目标的检测，最大程度保留合法的突发流量；在攻击缓解上，能够识别处于攻击路径上的交换机，并且不引入额外资源. 该方法总体上具有检测开销低，控制灵活，阈值弹性高，能够满足不同控制器的功能要求等特点.

3　控制平面饱和攻击及威胁模型

SDN网络在控制上具有逻辑集中的特点. 在OpenFlow协议的支持下，流规则有两种安装方式：主动安装和被动安装. 在被动安装中，交换机将到达流量的数据包头部与匹配域字段匹配，如果匹配成功，交换机按照流表规则转发. 否则，交换机向控制器发送Packet_In消息，请求相关流规则. 如图1所示，控制平面饱和攻击是DDoS攻击在软件定义网络中的一种特殊实践，它依赖于OpenFlow协议的转发漏洞，以耗尽控制平面计算资源为手段，从而达到瘫痪网络，拒绝服务的目的. 攻击者利用SDN网络探测识别技术和流规则重构技术，探知目标网络是否为SDN网络，并知晓流表中的大部分匹配字段，再控制僵尸网络向SDN网络中注入大量经过伪造的数据包，以避免流表规则匹配成功，如①所示. 攻击迫使交换机将恶意流量的信息发往控制器，并占据请求队列，使得控制器花费大量计算资源进行处理，合法流量的请求处理迟迟得不到响应，最终导致服务延迟或超时丢弃，如②所示. 在集中式软件定义网络中，控制器饱和仅会造成单点故障，但随着多控制器软件定义网络的实施，发生单点故障的控制器将工作负载全部发送到其他控制器进行处理，由于协助处理的控制器无法容纳分发给它的工作负载，就会导致后者故障，从而不断传递，导致级联故障，如③所示. 在本文中，我们假设攻击者可以侦测流规则的部分匹配字段，并构造包含IP地址、Mac地址等假信息的数据包，且攻击流量包含多种协议信息. 但由于控制平面饱和攻击的攻防本质是一场资源竞争博弈，当攻击总强度超过控制器集群所能容纳的负载总量时，无论如何执行迁移都无法实现饱和缓解. 所以攻击者虽然能够通过控制僵尸网络从网络中的任意位置发起攻击，但攻击流量总强度不能超过控制器集群能够处理的流量总强度上限，且攻击者资源有限，没有能力对控制器全程实施不间断攻击.

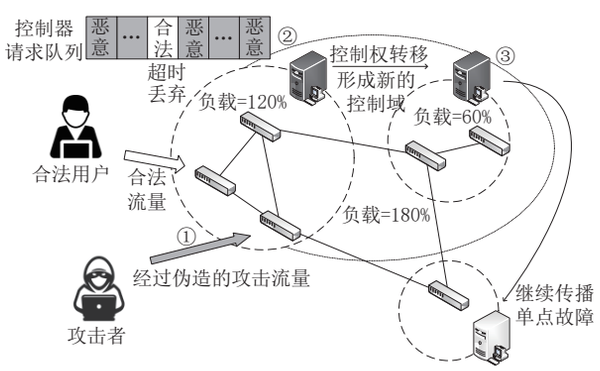


图1　控制平面饱和攻击

4　基于交换机迁移的三层防御架构

控制平面防御的目标是保护控制器的计算资源不被耗尽，能够持续且稳定地处理控制消息，并下发流表规则. 控制平面饱和攻击，与突发流量产生的饱和效果是相似的. 在实际中，往往采用负载均衡方法处理突发流量，而软件定义网络能方便地形成资源视图，适合进行负载均衡实施. 在多控制器的软件定义网络，我们尝试通过更改交换机的主控制器来缓解控制器的饱和情况，并利用控制器集群中未被充分利用的资源，来处理交换机产生的业务流量，这种更改交换机主控制器的策略被称为交换机迁移[28]. 依据此思想，我们提出了基于交换机迁移的控制平面饱和攻击防御方法，并实现了适用于多控制器软件定义网络的交换机迁移防御架构，该架构由状态同步模块、路由调度模块、饱和检测模块和

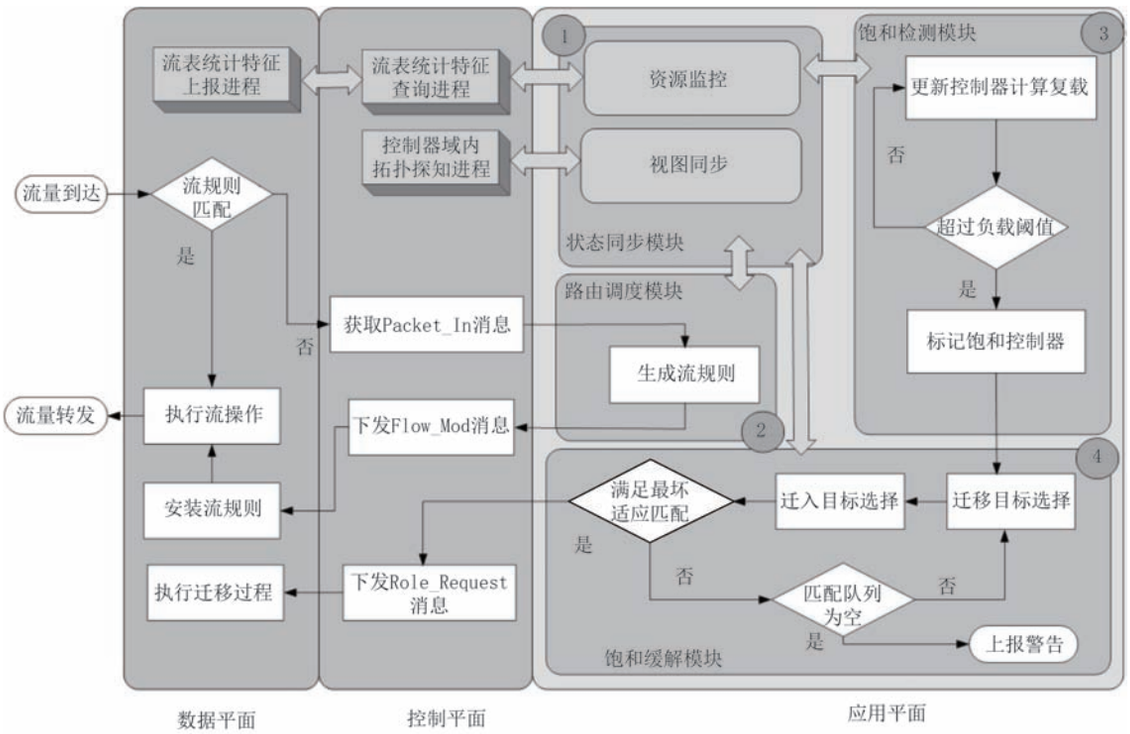


图2 基于交换机迁移的三层防御架构

饱和缓解模块构成，防御架构如图2所示.

状态同步模块，由视图同步模块和资源监控模块组成. 在本方法中，攻击的防御和转发功能的实现依赖于对网络的全局视图，但是控制器仅能获取自身控制域内的视图信息和资源使用，所以需要该模块来实现多控制器的状态同步. 视图同步模块负责定期探询控制器所掌握的拓扑信息，并将不同控制器探知到的信息进行整合，形成完整的网络拓扑视图，以便于为路由调度模块提供拓扑依据. 资源监控模块根据OpenFlow协议所提供的查询请求，获取OpenFlow消息和流表统计信息，并将其提供给饱和检测模块和饱和缓解模块，以进行攻击防御动作，是整个防御架构最先执行的一个模块，并在整个防御方法运行期间不间断执行.

路由调度模块，下发流表规则. 靠近网络边缘的交换机，常常有新流量进入，无法提前安装能与流量匹配的流规则，路由调度模块根据网络拓扑和既定的路由算法，对边缘交换机采用被动下发方式，使其生成Packet_In消息，上传源地址、目的地址等路由基本信息. 计算流量的整条转发路由后，该模块主动向位于后续路径上的交换机下发Flow_Mod消息，安装流规则，减少流量转发的排队时延.

饱和检测模块，监测控制器负载并标记饱和控制器. 通过状态同步模块所收集的南向开销、北向开销和水平开销，计算控制器的计算负载，在Cbench[29]的帮助下，测量控制器单位时间所能处理的OpenFlow消息数量上限，依据上限设立饱和检测阈值. 当控制器的计算负载超过该阈值，该模块标记控制器，转向饱和缓解模块. 否则，继续检测控制器是否饱和.

饱和缓解模块，确定迁移目标并执行迁移过程. 由饱和检测模块转到饱和缓解模块后，系统根据攻击流量所具有的特征和交换机的重要程度，对饱和控制器下的交换机进行迁移目标选择，识别位于攻击路径上的交换机，构建待迁移交换机集. 未饱和的控制器按照剩余负载被确立为迁入目标，并与待迁移交换机进行最坏适应匹配，寻找满足约束的辅助控制器. 待迁移交换机与辅助控制器完成匹配后，逐个下发角色改变命令，完成交换机迁移过程. 若无法实现匹配，则上报警告，攻击强度已超过集群所能容纳的负载总和.

在网络运行之后，状态同步模块开始不断向控制平面发送状态同步指令，控制平面在收到指令之后激活流表统计特征查询进程或控制器域内拓扑探知进程，将数据平面上报的流表特征和控制器探知到的域内拓扑进行封装，发往应用平面进行资源监控和视图同步，来构建全局视图. 当有流量到达时，先在数据平面进行流规则匹配，如果匹配成功则执

行流操作，将流量从相应端口转发．如果流规则匹配失败，数据平面经过控制平面传递Packet_In消息中的路由信息给路由调度模块，再由路由算法生成流规则，最终生成Flow_Mod消息层层下发到数据平面，安装流规则，并执行流操作，完成流量转发过程．在收到Packet_In消息和下发Flow_Mod消息时，路由调度模块同时向状态同步模块更新控制器资源监控信息，在此时间周期结束后，激活饱和检测模块，更新控制器计算负载，判断饱和控制器并标记，然后向饱和缓解模块进行转移．在饱和缓解模块中，依次进行迁移目标选择、迁入目标选择和最坏适应匹配，最终生成迁移方案或上报警告．控制平面在收到迁移方案后，根据控制器索引和交换机编号下发Role_Request消息，引导交换机完成迁移过程．

5　基于交换机迁移的控制平面饱和攻击防御方法

在本节中，我们将攻击防御过程的攻击检测和攻击缓解与交换机迁移过程中的迁移触发判定和目标选择进行关联，提出了一种基于交换机迁移的控制平面饱和攻击防御方法，来检测和缓解攻击或突发流量所导致的控制器饱和现象．在饱和检测阶段，我们通过测量控制器在OpenFlow消息的处理过程中产生的北向开销、南向开销和水平开销，设立针对计算负载的检测阈值，以检测控制器目前的运行负载是否会导致单点故障的发生．在饱和缓解阶段，根据流表统计信息和空间特征图，对饱和域中的交换机结点进行迁移定级，确定交换机迁移顺序．此外，为降低交换机迁移发生级联故障的可能，我们对辅助控制器的选择过程进行约束，由待迁移交换机与辅助控制器的剩余负载进行最坏适应匹配，直到饱和控制器恢复或辅助控制器耗尽．

本节基于图论建立了多控制器软件定义网络模型，网络拓扑用无向图$G=(V,E)$表示，其中，V表示网络中的设备节点集合，E表示设备节点之间的链路集合．C是控制器集合，$C=\{C_0,C_1,\cdots,C_M\}$，S是交换机集合，$S=\{S_0,S_1,\cdots,S_N\}$．x_{ij}是二进制变量，表示交换机S_j与控制器C_i的主从控制关系，取值为1表示控制器C_i是交换机S_j的主控制器，取值为0表示控制器C_i是交换机S_j的从控制器，如式(1)所示．其余网络性能参数及说明如表1所示．

$$x_{ij}=\begin{cases}1, & C_i\text{是}S_j\text{的主控制器}\\ 0, & \text{其他}\end{cases} \tag{1}$$

表1　网络性能参数及含义

网络性能参数	参数含义
M	控制器数量
N	交换机数量
φ_j	交换机S_j产生的Packet_In消息数
ω_j	主控制器对交换机S_j产生的Flowd_Mod消息数
$\mu_{ii'}$	控制器C_i与控制器$C_{i'}$之间的同步消息数
u_j	交换机S_j已经安装的流表项数量
θ_l^j	交换机S_j的第l条流表项转发的数据量

5.1　饱和检测

当有业务流量进入到边缘交换机时，交换机会先检查流表空间，将流量头部(源MAC地址、源IP地址、端口号等)信息与流表项的匹配域进行逐条匹配，如果匹配成功，则执行流表项中所指示的动作信息(转发、丢弃等)；如果匹配失败，则暂存数据包，并保存数据包头部信息生成Packet_In消息上报主控制器．控制器将路由计算所需要的特征信息提取出来，按照既定的算法计算路由路径和执行动作，再将匹配域、路由信息和动作域信息打包为Flow_Mod消息返还给对应交换机，交换机按消息内容执行相关动作，并将其固化为流表项保存在流表空间，以便对接下来的流量快速反应．此外，控制器可以直接向路由路径上的后续交换机下发Flow_Mod消息，保证在流量到达交换机前，提前安装流表项，从而减少交换机的上报频率和转发等待时间．基于以上对SDN流量转发流程的分析，控制器在转发过程中的负载主要来自于处理Packet_In消息和Flow_Mod消息的开销．可以对此过程中控制器负载水平的相关参数进行定义．

定义1.　北向开销．北向开销是指单位时间内控制器C_i处理来自交换机S_j的Packet_In消息并依据路由特征信息进行路由计算产生的消息数量，如式(2)所示．

$$CN_i=\sum_{j=0}^{N}x_{ij}\times\varphi_j \tag{2}$$

定义2.　南向开销．南向开销是指单位时间内控制器C_i依据路由信息生成流表规则并向交换机S_j发送Flow_Mod消息所产生的消息数量，如式(3)所示．

$$CS_i=\sum_{j=0}^{N}x_{ij}\times\omega_j \tag{3}$$

除了南向开销和北向开销，域间的信息同步也是控制器负载的主要来源之一. 控制器之间需要进行拓扑信息、端口统计信息等信息的同步过程，来帮助控制器获取其他子域的拓扑结构、流表统计和负载水平等信息，并向其他控制器传递跨域路由信息，实现流表规则的主动安装等全局性操作，对此可定义关于控制器负载水平的另一参数.

定义3. 水平开销. 水平开销是指单位时间内控制器C_i向其他控制器$C_{i'}$发送或接收同步消息所产生的消息数量，如式(4)所示.

$$CH_i=\sum_{i'=0,i'\neq i}^{M}\mu_{ii'} \tag{4}$$

通过对多控制器下SDN场景中业务流量转发过程的分析，控制器在工作阶段的负载主要来自于处理Packet_In消息的北向开销、下发Flow_Mod消息的南向开销和进行控制器间状态同步的水平开销. 因此，可对控制器的负载水平进行合理准确的定义表示.

定义4. 计算负载. 计算负载是指在单位时间内控制器C_i进行主要业务工作所消耗的资源开销，如式(5)所示. 其中α、β、γ分别为北向开销、南向开销和水平开销的单位负载.

$$Load_i=\alpha\times CN_i+\beta\times CS_i+\gamma\times CH_i \tag{5}$$

由于控制平面饱和攻击的特殊性，多级阈值的计算复杂度会大大加剧对网络资源的影响. 所以，本文以计算负载作为单级检测阈值，表示某一时间段内控制器的资源使用情况. 当控制器遭受到攻击时，控制器的资源消耗情况会快速攀升，所以可以通过对控制器的资源评估，设立相应的阈值δ，来检测控制器是否发生了饱和情况，其计算方式如式(6)所示，其中$CPUR$表示实际业务中对于控制器CPU的最大使用率，$Load_{max}$表示控制器的OpenFlow消息处理上限.

$$\delta=CPUR\times Load_{max} \tag{6}$$

当控制器计算负载小于δ时，表征控制器未遭受攻击，控制器能够发挥正常的业务处理能力，及时地处理业务流信息. 当控制器计算负载大于δ时，表征控制器可能遭受到了攻击，消息的到达速度超过了控制器的处理速度，会导致流量不能及时转发，甚至出现数据包大量丢失等问题.

在交换机迁移过程中，控制器的负载表示是极其重要的一部分，它关系到了迁移操作的触发条件，是确定迁出域的判断依据. 我们对比了与本方法具有相似性的其他6种方法，如表2所示. 在这些方法当中，Rebalance、TSSM、Q-learning和ESMLB方法只针对Packet_In消息来表征控制器负载情况，前三者通过测量Packet_In数据包的速率来判断是否执行迁移，该指标瞬时性较强，在突发流量的影响下容易错误判断饱和情况，ESMLB方法采用统计指标，记录单位时间内Packet_In消息的数量，相较于前三种方法具有较高的稳定性. 但是，控制器不仅仅只负责处理Packet_In消息，还负责下发Flow_Mod消息，其占比能够达到控制器收发的OpenFlow消息数的一半，所以仅仅以Packet_In消息这样的北向开销表征控制器负载缺乏准确性. Balcon策略相较于前四种策略对北向开销进行了细化，将Packet_In消息分为流量在域内传递产生的Packet_In消息和流量由域外进入域内产生的Packet_In消息，两者采取不同的计算公式. 此外，Balcon策略增加了南向开销的计算方法，进一步完善了对控制器负载的表征维度，使得负载表征更加贴近实际负载. 但是，随着网络范围的逐渐扩大，控制器之间需要频繁地交换状态信息，实现流量的跨域转发，该方法并未考虑到这部分的消息负载. Highly-efficient策略考虑到了控制器之间的同步开销，以Packet_In消息大小和速率、交换机之间的跳数、流表大小等多种特征进行各类开销计算，最终加和求得控制器的负载. 但是面对控制平面饱和攻击，复杂的计算会占用控制器的额外开销，导致对控制器的抗攻击能力减弱，所以Highly-efficient策略的计算方式并不适合应用于攻击环境中. 对于现有研究中负载表征指标所存在的缺陷，我们采用Packet_In消息、Flow_Mod消息和控制器间同步消息的计数统计来作为负载表征的指标. 以消息数表征与控制器相关的南向、北向和水平开销，使得负载表征随着消息的接收和发送自然完成统计，能够降低对控制器的负载影响. 并且控制器能够处理的OpenFlow消息数量上限可以由Cbench测量获得，我们的办法可以更好地关联控制

表2 负载表征相关研究比较

相关研究	负载表征指标			指标类型
	北向开销	南向开销	水平开销	
Rebalance[23]	√	×	×	测量指标
TSSM[28]	√	×	×	测量指标
Q-learing[30]	√	×	×	测量指标
ESMLB[31]	√	×	×	统计指标
Balcon[32]	√	√	×	计算指标
Highly-efficient[33]	√	√	√	计算指标
本方法	√	√	√	统计指标

器负载与饱和检测阈值之间的关系，从而更加稳定、准确地识别饱和状态，确保迁移防御执行的有效性.

5.2　饱和缓解

5.2.1　迁移交换机选择

当控制器遭受到攻击时，通过计算负载与阈值δ的比较，能够确定饱和控制器，但一个控制器的子域中往往包含多个交换机，所以每个交换机都可能是潜在的迁移目标. 本小节研究的重点问题是期望能够通过流表统计特征和交换机的空间特征图准确地选择位于攻击路径上的交换机节点，以达到快速降低饱和控制器负载的效果.

攻击者通过僵尸网络发送大量伪造MAC地址的数据包，使得流表项无法与流量进行匹配，交换机不得不向控制器上报大量路由特征信息，迫使控制器花费大量时间处理伪造流量，无暇顾及正常业务流量. 正是由于这种攻击手段，造成了攻击流量和业务流量在交换机的流表空间中具有不同的统计特征. 对于伪造的地址信息，每条流表项转发的数据总量少，而正常地址，由于业务多次交互的原因，一条流表项数据包的统计量大很多，这就造成了攻击路径上的交换机呈现出了流表项多，但大多数流表项转发的数据包总量少的特点.

定义5.　平均数据包. 平均数据包是指经过交换机S_j转发的全部路由路径的平均数据包多少，如式(7)所示. 平均数据包越多，表明交换机S_j所在的拓扑位置，进行正常业务流量转发的行为越多，每条流表项被匹配的次数越多，遭受攻击的可能性越小.

$$P_j=\frac{\sum_{l=0}^{u_j}\theta_l^j}{u_j} \tag{7}$$

在传统网络架构中，DDoS攻击所产生的流量与正常的业务行为产生的流量在对称特征上具有巨大的差异. 正常情况下，服务器接收和发送的流量之比约等于1. 服务器在DDoS攻击的影响下，服务器向外发出的流量数量相比于接收到的流量数量要少很多，其接收和发送的流量之比越大，意味着服务器遭受到的攻击强度越高. 这种接收和发送流量的差异，被称为DDoS攻击的不对称特征. 控制平面饱和攻击也具有这种特征，其不对称性主要表现在控制器接收的Packet_In消息数量与下发的Flow_Mod消息数量之间的不对称.

定义6.　上下行流量比率. 上下行流量比率是指单位时间内交换机S_j向主控制器C_i发送Packet_In消息与接收主控制器C_i下发的Flow_Mod消息数量之比，如式(8)所示. 上下行比率的数值越远离1，表示交换机未被响应的Packet_In消息数量越多、攻击者借助交换机S_j向控制器C_i发起攻击的概率越大. 非边缘交换机采用主动方式下发流表项，此类交换机的上下行流量比率会趋于0，由此还能进一步区分边缘交换机与非边缘交换机.

$$R_j=\frac{\varphi_j}{\omega_j} \tag{8}$$

在业务网络的拓扑中，主干网络的存活状态是影响该网络服务质量的一个重要因素. 相比于边缘网络交换机，主干网络中的交换机在路由路径中的占比更高，更容易被攻击者选择为实施DDoS攻击的入口. 因此，在迁移过程中，主干网络上的交换机应当处于优先位置.

定义7.　交换机重要程度. 交换机重要程度是指该交换机在业务网络中有流量通过的概率，如式(9)所示. 在网络拓扑中，最重要的交换机就是位于主干路径上的交换机，该路径上的交换机有更多的流量经过，是保证网络服务质量的重要节点. 以交换机重要程度作为迁移交换机的度量之一，有助于优先保证主干路径产生的请求被处理，确保大多数服务能够被使用.

$$I_j=\frac{\sum_{l=0}^{u_j}\theta_l^j}{\sum_i^N u_i\times P_i} \tag{9}$$

通过对流表统计特征和空间特征图的分析，得到了交换机目标选择的三个度量标准：平均数据包、上下行流量比率和交换机重要程度，综合以上的三个指标，可以对交换机的迁移系数进行定义.

定义8.　迁移系数. 迁移系数表示交换机在迁移过程中所具有的优先级，如式(10)所示. 其中，a、b、c是各个特征值影响迁移系数的权重参数. 迁移系数越高的交换机，在迁移过程中的优先级越高.

$$\begin{aligned}Grade_j=&a\times\left(1-\frac{P_j-P_{min}}{P_{max}-P_{min}}\right)+\\&b\times\left(\frac{R_j-R_{min}}{R_{max}-R_{min}}\right)+c\times\left(\frac{I_j-I_{min}}{I_{max}-I_{min}}\right)\end{aligned} \tag{10}$$

通过对饱和控制器子域内的所有交换机进行迁移评估，获得每个交换机的迁移系数并排序，选取具有$Grade_{max}$的交换机作为本次迁移过程的迁移交换机. 通过式(10)我们可以看出，迁移系数与平均数据包数负相关，与其他两个指标呈正相关，并且每个

特征值对迁移系数的影响与权重系数有关. 上下行流量比率是与控制器饱和最直接相关的特征表现，其主要是为了迁移出产生最多Packet_In消息数量的交换机，以快速降低饱和控制器负载，所以该特征对迁移系数的权值应是最大的. 平均数据包和交换机重要程度，都偏向于判断该交换机所处的位置，它们主要是为了保障首先迁移位于主干网络上的交换机，从而优先恢复大多数路由路径的通信，提高服务质量，所以对迁移系数的影响应略低于上下行流量比率.

5.2.2 辅助控制器选择

交换机迁移的最小单位是一个交换机整体，这种方式能够快速地使迁出域负载下降，也会使得迁入域负载快速上升，所以可能会导致迁入域的控制器立刻出现饱和的情况，从而导致再次迁移的情况，这种情况被称作控制器乒乓问题，是多控制器架构下引发级联故障的根本原因. 因此，在进行迁入域选择时，根据交换机所能产生的负载与控制器的剩余负载之间进行最坏适应匹配，能够有助于避免控制器乒乓问题的产生.

定义9. 交换机贡献负载. 交换机负载是交换机S_j所产生的消息导致其主控制器产生的计算负载，如式(11)所示，其中包含上报的Packet_In消息和接收的Flow_Mod消息所产生的负载，以及跨域情况下，主动安装流表项所产生的部分水平开销.

$$Load\_Switch_j=\alpha\times\varphi_j+\beta\times\omega_j+\gamma\times\mu_j \tag{11}$$

所有未饱和控制器都会被标记为辅助控制器，接纳具有最高迁移系数的交换机. 由于仅有单个交换机需要接收，所以可对辅助交换机按照剩余负载能力进行排序，优先选择剩余负载能力最高的控制器接收迁移目标，并通过式(12)为级联故障约束条件决定辅助控制器是否接收迁移目标.

$$Load\_Switch_j\leqslant\delta_{target}-Load_{target} \tag{12}$$

5.3 交换机迁移算法

综上所述，我们提出了交换机迁移算法，该算法涵盖饱和检测模块和饱和缓解模块内容，主要功能是判断控制器是否饱和，并在判断饱和之后执行饱和缓解过程，生成迁移方案. 在网络建立之后，网络管理人员设立饱和检测阈值并第一次执行该算法，在后续网络运行过程中随着时间自动地重复执行. 交换机迁移算法的伪代码如算法1所示. 首先，收集控制器的负载，检测攻击是否发生. 当某时刻，控制器受到攻击，通过收集交换机和控制器的资源情况，获取饱和控制器的计算负载和其子域内交换机的负载贡献度并建立各个初始设备集合(第1～6行). 然后，进行迁移目标选择. 依据饱和域内交换机的平均数据包、上下行流量比率和交换机重要程度获取所有饱和控制器下交换机的迁移系数，并降序排序迁移系数，得到迁移交换机的优先级排序(第8～15行). 最后，进行迁入域目标选择. 将控制器集群中的非饱和控制器划分为辅助控制器，并按剩余负载进行降序排序，选举具有最大剩余负载的控制器为迁入目标控制器(第16～17行). 由于辅助控制器的剩余负载不一定能够接纳具有高优先级的交换机，因此需要从优先级最高的交换机开始遍历，直至找到能够被辅助控制器所接纳的最高优先级交换机，返回迁移目标交换机和迁入目标控制器，执行迁移过程(第18～20行). 若遍历交换机结束后，仍未找到合适的交换机，则上报给协调器，声明本次迁移失败，控制器严重饱和.

设网络中共有n个控制器，饱和域内有m个交换机. 在饱和检测阶段，需要遍历所有控制器，时间复杂度为$O(n)$. 在饱和缓解阶段，首先需要遍历饱和控制器下的交换机结点，计算各个交换机的迁移优先级，最坏时间复杂度为$O(m)$. 然后对待迁移交换机和辅助控制器进行选择排序，最坏时间复杂度分别为$O(m^2)$和$O(n^2)$. 最后，进行辅助控制器与待迁移交换机之间的匹配过程，由于进行最坏适应匹配，只需将具有最多剩余负载的控制器与各个待迁移交换机进行匹配即可，所以辅助控制器选择最坏时间复杂度为$O(m)$. 由于在实际网络中，控制器的数量与饱和域内交换机的数量大小关系并不固定，所以交换机迁移算法的最大时间复杂度为$O(max(n^2,m^2))$.

算法1. 交换机迁移算法.

输入：控制器计算负载$Load$；阈值δ；交换机负载贡献$Load_Switch$

输出：迁移目标交换机S；迁入目标控制器C

```
BEGIN
1.  初始化饱和控制器集合List_over = [ ]
2.  初始化迁移交换机集合List_migrate = [ ]
3.  初始化辅助控制器集合List_assist = [ ]
4.  FOR i = 1 TO M DO
5.    IF Load_i ≥ δ THEN
6.  List_over ← List_over + C_i
7.  IF List_over ! = NULL THEN
8.    FOR C_i∈List_over DO
9.      FOR S_j∈C_i DO
```

10. $P_j \leftarrow$ 计算交换机S_j的平均数据包P_j
11. $R_j \leftarrow$ 计算交换机S_j的上下行流量比率R_j
12. $I_j \leftarrow$ 计算交换机S_j的交换机重要程度I_j
13. $Grade_j \leftarrow$ 计算交换机S_j的迁移系数$Grade_j$
14. $List_{migrate} \leftarrow List_{migrate} + (S_j, Grade_j)$
15. $List_{migrate} \leftarrow SORT(List_{migrate}, Grade_j)$
16. $List_{assist} \leftarrow S / List_{over}$
17. $List_{assist} \leftarrow SORT(List_{assist}, \delta - Load_i)$
18. FOR $S_j \in List_{migrate}$ DO
19. IF $Load\_Switch_j \leqslant$ MAX($List_{assist}$) THEN
20. RETURN S_j, $C_{\max\_assist}$
21. RETURN警告信息

END

6　实验分析

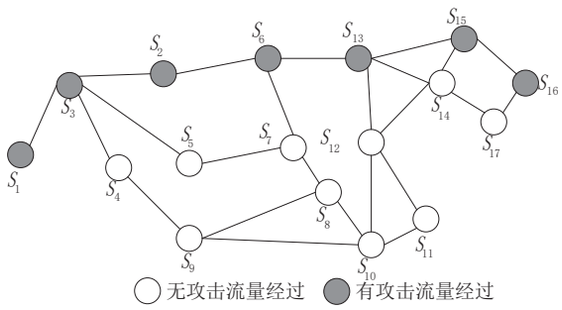
6.1　实验环境建立

本文的实验部署在由8个Intel酷睿2双核T7700型号的CPU、16 GB内存组装的服务器之上，该服务器运行的操作系统为Ubuntu 18.04.1，并通过Open vSwitch虚拟交换机和RYU控制器来搭建拓扑，拓扑结构如图3所示，这两种拓扑结构分别是通信网络拓扑和数据中心网络拓扑的经典拓扑结构. 两种网络拓扑分别具有不同的流量特点，网状拓扑转发路径数量多，不同节点间通信时转发路径重合率低，交换机负载贡献差异性大，而树状拓扑转发路径有限，不同节点间通信时转发路径重合率高，交换机负载贡献相对类似. 主机节点中通过运行基于Scapy模块实现的脚本程序发送正常业务流量，为网络环境提供正常业务背景流量，而攻击主机节点中通过Hping3发起攻击，攻击的有效性在文献[34]和文献[35]中已经证实. 具体软件配置如表3所示.

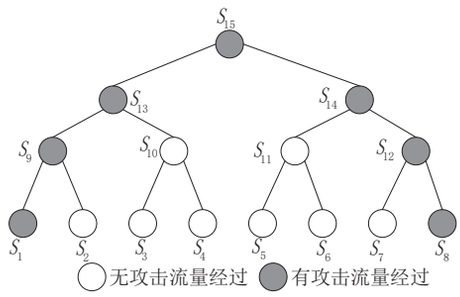
实验中所有控制器具有相同的处理性能和资源配置，为去除虚拟环境下的扰动因素，通过Cbench测得控制器所能处理的OpenFlow消息数量，并以此设置控制器饱和阈值为1000，为控制器最大处理消息的50%左右. 对于计算负载和交换机负载中的权重系数，依据实验中处理Packet_In消息、Flow_Mod消息和同步消息的CPU变化和文献[32]的先验知识，设置$\alpha:\beta:\gamma=1:1:0.5$. 对于目标选择中的权重系数，依据多次重复实验，设置$a:b:c=0.3:0.4:0.3$.

6.2　实验结果分析

为了证明方法的有效性，我们对比了当前几种



（a）网状拓扑



（b）树状拓扑

图3　实验拓扑

表3　软件配置信息

软件名称	版本
Docker	20.10.17
Ryu	4.34
Open vSiwtch	2.13.8
OpenFlow	1.3
Hping3	3.0.0
Scapy	2.5.0
Cbench	0.5
Python	3.6.9

性能较好的防御策略. 对比策略有OpenFlow[1]策略、Balcon策略[32]和Rebalance策略[23]. 其中，OpenFlow策略为无行动的静态策略，控制器不采用任何防御措施；Balcon策略将交换机迁移视为图划分问题，该策略由最大数据率交换机开始，构建强连接交换机群，不断吸纳与该交换机群内具有最大数据交换率的其他交换机，以此交换机集群为迁移目标进行迁移评估，不断迭代优化，直至控制器之间的负载标准差达到最小化；Rebalance策略基于贪心算法，寻找各个时刻控制器负载所能达到的最小标准差，并与此时刻控制器之间的标准差进行比较，若差值大于阈值，则按照最小标准差方案进行交换机迁移，否则，保持不变.

6.2.1　资源竞争博弈

本实验首先获取控制器的CPU使用情况，并与控制器的计算负载进行对比，由于控制器的计算负

载与拓扑结构无关，所以我们经过实验后，选择以网状网络进行本节的实验分析，实验结果如图4所示．需要说明的是，在本小节及后续实验结果当中，纵坐标常常出现的负载是指控制器的计算负载，该坐标是由公式(5)计算所得，是一个计算量，并无具体实意单位．通过实验结果的对比，各个控制器的计算负载与CPU使用率虽然在具体数值上有所差异，但其在整体趋势上具有高度的一致性，且在实验中发现，CPU使用率多数情况下比计算负载所得的控制器负载情况延后1～2个监测周期，并且在无攻击环境下，CPU使用率易受其他因素影响，出现剧烈的抖动．所以对比了各个时刻的计算负载与相同时刻或后续1～2个时刻的CPU使用率变化趋势，结果显示各个控制器在超过93%的时间，计算负载与CPU使用率的变化趋势是一致的．

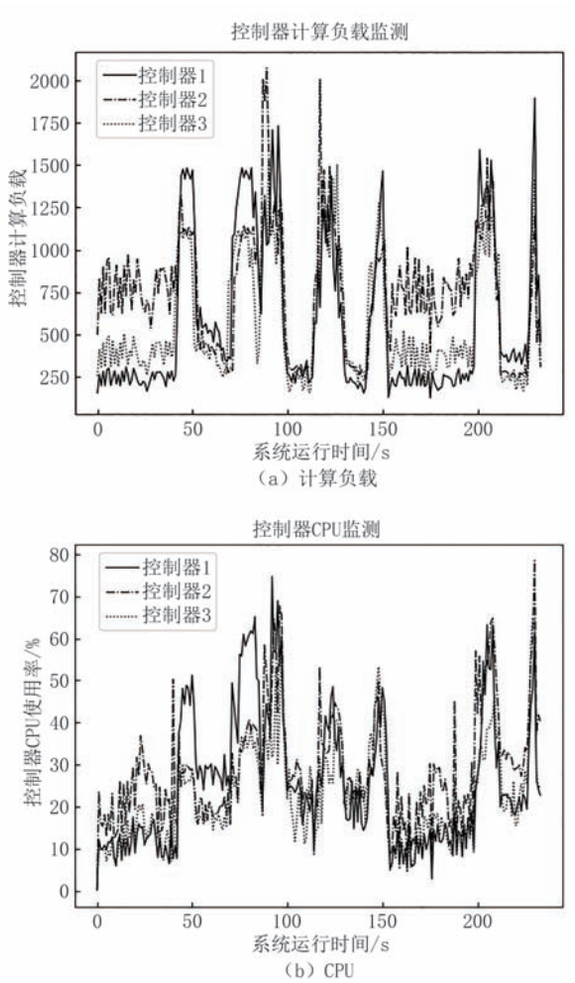


图4 控制器资源监测

此外，为验证计算负载能够有效地表征CPU使用率，将控制器的计算负载除以计算负载上限，再求得每时刻计算负载比率与CPU使用率的差值．在超过80%的时间内，两个数据之间的差距小于15%，这在CPU使用率的不断波动与延后下是可以容忍的．

根据以上的实验结果，我们可以认为文中所定义的计算负载与控制器的CPU占用情况呈正相关，从而能够有效地表征控制器的负载并对控制器达到饱和的情况做出准确判断．

其次，借助实验证实控制平面饱和攻击的攻防过程是一场资源竞争博弈，防御者可以加注资源来缓解攻击带来的影响．交换机迁移是将交换机迁移到另外的控制器子域中，来减少原控制器的负载，其等价于借用迁入目标控制器的部分资源来扩充原控制器的资源上限．因此，可以将交换机迁移看作另一种类型的控制器资源扩容，不过它并没有对控制器集群增加额外的资源，只是将集群中的资源进行整合使用，实验结果如图5所示．柱状图显示了不同控制器处理能力下控制器的饱和次数，而折线图显示受攻击控制器的平均负载与阈值之间的负载差，表明了当前攻击对控制器的危害程度．结果可以看出，攻击强度不变的情况下，随着控制器的资源投入不断升高，控制器的饱和次数快速地下降，并且控制器的处理资源越多，攻击者想要继续使得控制器长时间饱和，需要投入更多的攻击资源才能达到效果，这对攻击者来说成本高且风险大．

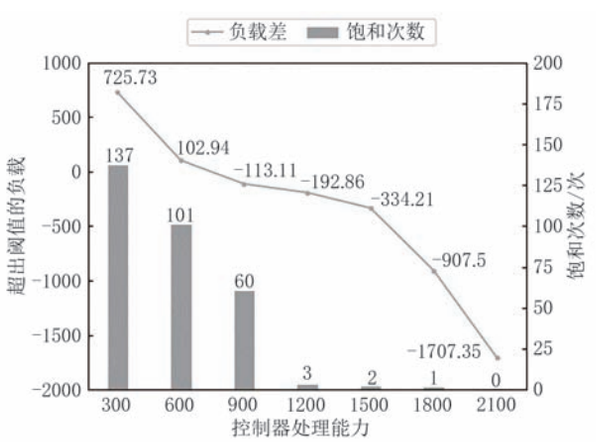


图5 不同资源投入下控制器的负载情况

6.2.2 动态目标选择

我们通过改变攻击的强度和辅助控制器之间的资源配比，来验证交换机迁移算法具有较好的动态性和灵活性．经过实验，本方法在网状拓扑和树状拓扑中有着相似的结果，所以本节所有实验同样以网状网络作为实验分析的代表，结果如图5所示．实

验中将$\{S_1,S_3,S_2,S_6,S_{13},S_{15},S_{16}\}$作为攻击路径，并且在实验过程中，保持背景流量强度不变，使得攻击流量强度与背景流量强度之比分别为0.5∶1、1∶1和1.5∶1，并记录在第i个时间周期T_i时的交换机优先级序列．在T_{30}处向网络内目标发起攻击，并在T_{60}处停止攻击，所得实验结果如表4所示．通过实验结果可以看出，在攻击发起前和结束后，交换机S_7总是处于迁移优先级最高的位置，并且其他正向重要交换机的迁移优先级也处于较高的位置．当网络内出现攻击流量时，交换机的迁移优先级发生变化，攻击路径上的交换机迁移优先级提升，无论攻击强度与背景流量强度的比率为多少，在攻击持续的时间段内，攻击路径上的交换机都占据着较高的迁移优先级，并且攻击强度越大，迁移优先级越高，迁移算法对攻击路径上交换机的选择可能性越高．

表4　部分交换机的迁移优先级排序

攻击强度	交换机(是否处于攻击路径上)	迁移优先级(1为最高，13为最低)				
		T15	T30	T45	T60	T80
攻击流量＜背景流量	S_3(是)	11	11	4	5	11
	S_7(否)	1	1	5	4	1
	S_9(否)	5	8	11	11	8
	S_{13}(是)	13	13	7	10	13
	S_{16}(是)	2	2	1	1	2
攻击流量＝背景流量	S_3(是)	11	11	3	3	11
	S_7(否)	1	1	10	7	1
	S_9(否)	6	8	11	11	8
	S_{13}(是)	13	13	7	8	13
	S_{16}(是)	2	3	1	1	2
攻击流量＞背景流量	S_3(是)	3	11	11	3	11
	S_7(否)	1	1	11	10	1
	S_9(否)	5	6	10	11	7
	S_{13}(是)	13	13	5	6	13
	S_{16}(是)	2	3	1	1	2

除此之外，实验通过改变辅助控制器能够用于接纳交换机的剩余负载，来讨论交换机迁移算法在迁入域选择方面的表现．本防御方法中的迁入域选择依据辅助控制器的剩余负载，也就是意味着辅助控制器的剩余负载越多，它能接纳的交换机和负载应该越多，如图6所示，实验结果验证了上述思想．辅助控制器在不同剩余负载的情况下，接纳的交换机数量和承担的负载不同．在辅助控制器的剩余负载相近的情况下，迁移过程会在辅助控制器之间交替进行，使得辅助控制器承担几乎相同的负载．当辅助控制器之间的接纳能力具有巨大的差距时，一方控制器可能会占据所有交换机，这将会导致控制器乒乓问题的产生概率大幅度上升．因此，无论是否发生交换机迁移，控制器之间的负载应该保持均衡、稳定，而本方法在此方面的表现将在之后的部分中阐述．通过上述实验，可以明显地看出，本方法在不同网络条件下的目标选择具有动态性和灵活性，这对攻击防御是极为重要的一点，迁移目标和迁入目标的动态变化能够有效地提高网络的安全水平．

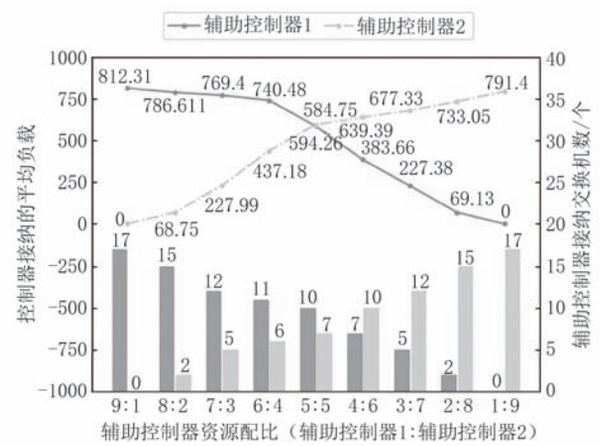


图6　辅助控制器接纳交换机情况

6.2.3　交换机迁移防御架构的防御有效性和迁移表现

我们监测了在152个时间周期内(攻击在第10个时间周期发生，并在第140个时间周期结束)四种防御策略下，不同控制器和控制器集群的负载和迁移表现，目标是验证本方法在使得控制器集群饱和最低化方面具有较好的表现，即能够尽可能防止控制器饱和并且能够尽可能消除迁移带来的负面影响．为了实验的真实性和准确性，我们重复进行了10次实验，采取了实验数据的平均结果．除此之外，为了能够更加直观地对比不同策略下的迁移效果，我们依据防御目的和相关研究的评判标准，从四方面对迁移效果作出了定义．

定义10.　饱和次数．饱和次数是指控制器C_i在实验过程中计算负载超过阈值的总次数，当集群内任意控制器饱和时，控制器集群同样被标记为饱和状态．该度量表征着控制器集群对于饱和攻击的抗打击能力，饱和次数越低，意味着防御策略的防护性能越好．

定义11.　首选率．首选率是指在迁移目标选择过程中，攻击路径上的交换机S_j被确定为最高迁移优先级目标的次数占迁移次数的比率．该度量表

征着迁移策略在迁移目标选择方面的表现，首选率越高，意味着迁移策略对攻击路径上交换机的选择越准确.

定义 12. 单次迁移率[36]. 单次迁移率是指在所有迁移操作中，交换机S_j仅进行过一次迁移的比率. 该度量表征着迁移策略在迁移目标和迁入域目标选择方面的表现，单次迁移率越高，单个交换机在多个控制器间来回迁移情况的发生次数越少，意味着该迁移策略下发生控制器乒乓问题的可能性越低，级联故障的发生概率越低.

定义 13. 保持度[37]. 保持度是指在攻击结束后，在经历了交换机迁移操作的重新映射下，现在的控制器与交换机的主从关系与原主从关系相同的保持率. 业务网络初始映射关系往往是基于路径质量、传输时延、通信负载等度量权衡下的最优结果，因此，迁移能够保持的映射关系应该尽量多，这样才能在保证安全性的同时确保网络的高质量通信.

如图7和表5所示，OpenFlow方法并没有采取任何的迁移方法，由于实验中控制器的负载能力是一致的，即饱和检测阈值是相同的，控制器1和控制器3控制的交换机较少，能够容纳攻击产生的负载，而在控制器2子域中含有更多的交换机，容易产生饱和现象. 因此，控制器1和控制器3发生饱和的次数为0，而控制器2发生饱和的次数都超过了总时间的65%. 使用交换机迁移的三种防御策略，通过将控制器2子域下的交换机向外迁移，来达到缓解控制器2饱和情况的目的. 通过实验结果可以看出，三种迁移方案都达到了上述的目的，控制器2的饱和次数都有了明显的下降.

对于Balcon策略，迁移方式是以强连接交换机集群为迁移目标的集群迁移策略. 该集群是由饱和域中产生负载最高的交换机开始扩展的，这类交换机在攻击的场景下往往有攻击流量经过，与此类交换机具有大量数据交换的交换机也往往有攻击流量经过. 将此类交换机集群迁移到其他控制器域，虽然会导致迁出域控制器的负载快速下降，但是同时迁入域控制器的负载也将快速上升，从而导致控制器之间发生控制器乒乓问题，迁入域与迁出域身份互换，将部分交换机再次迁移回原控制器，这也导致了图8中该策略具有较高的保持度和首选率，但单次迁移率并不理想. 在图7中可以看出，Balcon策略虽然使得控制器2的饱和次数得到了大幅度的改善，但控制器之间交叉饱和，使得在比较长的时间内，控制器集群总是有一个控制器处于饱和情况，集群饱和次数更接近控制器饱和次数之和.

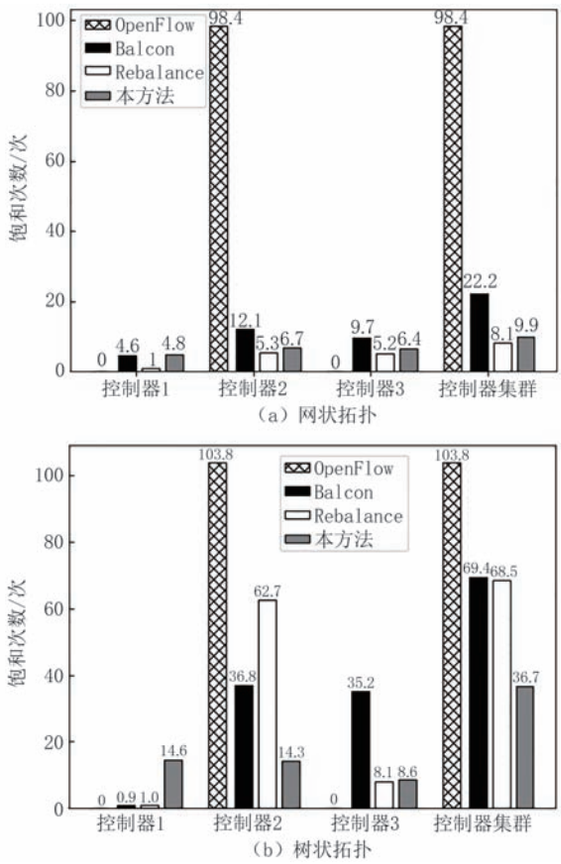


图7 不同方法下的控制器饱和次数

表5 不同策略下控制器集群的饱和率

方法	Openflow	Balcon	Rebalance	本方法
网状网络下系统饱和率	0.65	0.15	0.05	0.07
树状网络下系统饱和率	0.69	0.46	0.46	0.24

对于Rebalance策略，迁移触发条件是以控制器间的负载标准差为标准，若当前标准差与迁移能够达到的最小标准差之差大于阈值，则执行标准差最小方案的迁移. 另一方面，该策略迁入目标的选择，是由控制器与交换机之间的跳数来决定的，方法以最小跳数来决定交换机的目标迁入域. 正是由于Rebalance策略的这种迁入域选择方法，该策略往往会陷入局部最优的情况，即采取的迁移方案只能保证负载标准差区域最小，并非是全局最小的. 因此，呈现出图7所示的结果，控制器1与控制器2和3的饱和情况差距较大. 在图8中可以看到Rebalance策略具有较低的保持度和首选率，这是由于该策略追求的迁移效果是控制器之间的负载标准差，而攻击路径上的交换机都是具有较高负载贡献的交换机，若以其为迁移目标，反而可能会使交换机之间的负载标准差变大，所以该策略需要使用部分低负载贡

献的交换机来达到平衡控制器之间的负载标准差的目的，从而造成了这种现象.

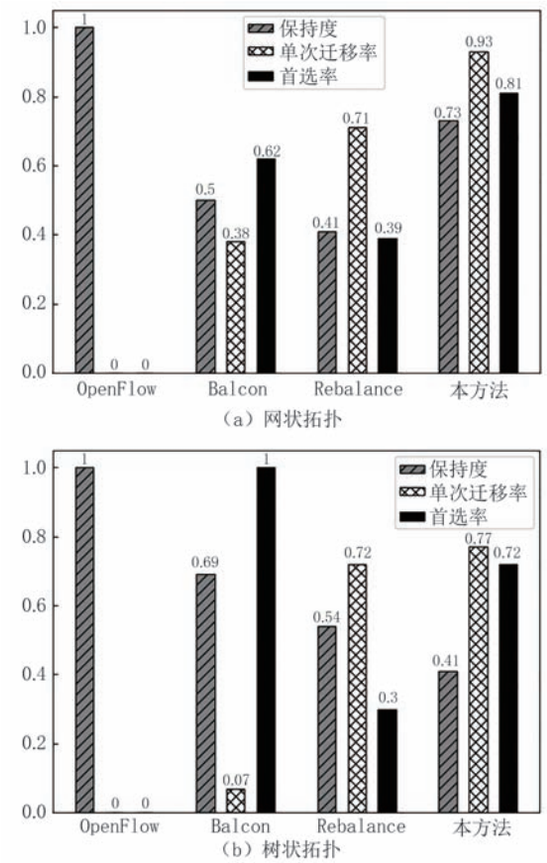


图8　不同方法下的迁移效果

本方法以控制器负载检测饱和，将攻击路径上的交换机迁移到能够容纳其负载的其他子域. 首先，本方法能够有效地缓解攻击对控制器的影响. 相较于无防御策略，有效地降低了控制器集群的饱和率，在两种拓扑结构下分别由65%降低到7%，由69%降低至24%，下降幅度分别为90%和65%. 本方法采用单目标迁移策略，在一个单位时间内，仅允许一个交换机执行迁移动作，因此从控制器出现饱和情况的时刻起，此后的多个单位时间内，控制器可能都不会由饱和状态改变为未饱和状态，使得控制器的负载相比于集群迁移不能快速下降，正如图7(a)所示，所以本方法的饱和次数会略高于Rebalance策略，但差距较小，仍处于可以接受的范围. 但在面对树状网络时，由于交换机负载贡献情况类似，Rebalance容易陷入局部最优，导致无法通过迁移缓解饱和现象，所以呈现出7(b)的结果.

其次，本方法能够有效地均衡控制器之间的负载水平. 在6.2.2中，我们提到控制器之间的负载应该均衡且稳定，从而降低控制器乒乓问题的发生概率. 图9和图10展示了不同拓扑下采取不同防御策略时，控制器集群的负载情况，以折线表示控制器集群的平均负载，以阴影表示控制器集群中控制器的负载最大值和最小值，即控制器集群的负载范围. 通过图9(a)和图10(a)可以看到，OpenFlow策略具有最大的阴影面积，即控制器之间的负载差距最大. 图9(d)和图10(d)为本方法所呈现的负载，在时间维度上，平均负载折线最趋向于直线，虽然本方法在负载的调节的速度上略低于其他两种方法，但是在控制器负载的调节质量上远远高于其他两种方法，控制器集群所具有的负载更加稳定. 除此之外，本方法曲线的阴影面积最狭小，在攻击持续期间，负载标准差最低时长占64.6%和63.5%，Balcon方法占25.4%和32.9%，而Rebalance方法仅占10.0%和3.6%. 从这些数据能够看出，本方法有着优秀的负载均衡能力.

此外，本方法能够有效地保证控制器乒乓问题发生概率最低化. 在本方法中，我们能够准确地锁定攻击路径上的交换机，只需要将这些高负载贡献度的交换机迁移出饱和域，就可以达到降低控制器负载的目的. 然后，依靠剩余负载进行的迁入域目标选择，我们能够提前预估交换机迁入目标子域后是否会引发目标控制器饱和，尽量保证不牺牲其他子域的服务质量来处理饱和问题. 实验结果如图8所示，迁移交换机选择方法保证了本方法在首选率的高表现，辅助控制器选择方法确保了本方法具有最高的单次迁移率，即控制器乒乓问题的发生率最低. 在面对树状拓扑时，由于一半的交换机都处于攻击路径上，交换机负载贡献类似，当控制器饱和时，本方法为了保证迁移操作不会引发控制器乒乓问题，仅能对负载贡献较低的一些交换机进行迁移，以尽可能缓解控制器的饱和情况，这也导致了本方法在树状拓扑中具有较低的保持率. 但是，将三个迁移效果按照相同的权值进行加和，形成了对每种方法的总体评分后，本方法在总体上的评分表现明显高于其他两种方法，所以本方法具有最好的迁移效果.

对交换机进行迁移的最终目标是保护控制器的服务功能，而控制器的服务功能最直接的体现就在于流量转发的往返时延上. 所以，我们测量了应用不同方法下合法流量的往返时延，结果如图11所示. 由于流表的存在，只有第一个数据包的信息被发往控制器，所以我们在相同Mac地址下仅发送4个数据包，既模拟正常流量转发，又避免速率过

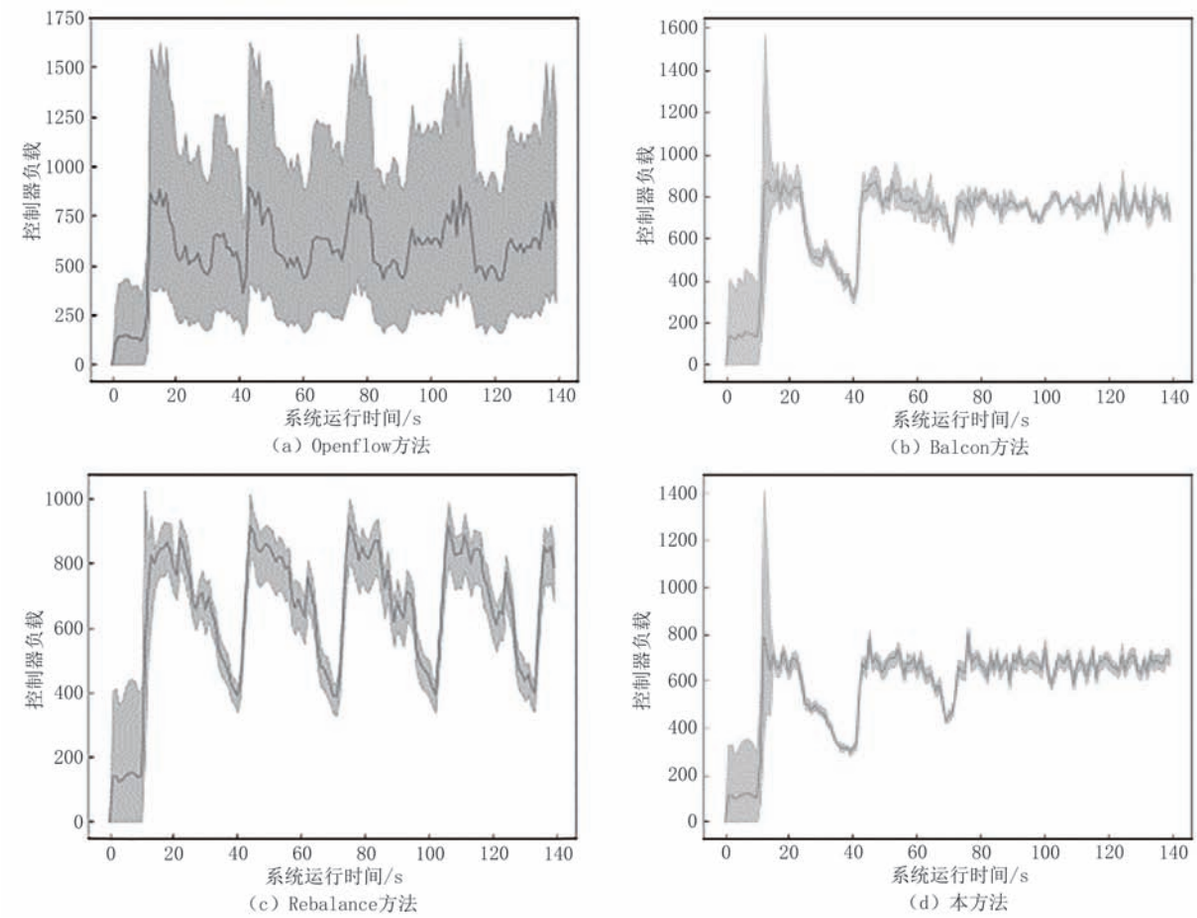


图9 网状网络下不同方法下控制器集群的负载情况

高,形成新的攻击.从图中可以看到,虽然本方法在网状网络具有最大的单次RTT,但是最大RTT仅高0.4秒,并且本方法的平均RTT总是低于其他三种方法,说明出现最大RTT仅是小概率事件,在总体上还是有着较低的RTT,保证了服务质量.

综上所述,本方法能够有效地缓解控制器的饱和情况,相较于其他两种方法在迁移效果和负载均衡方面有着更好的表现,并且能够有效降低攻击带来的服务质量的下降,保证服务器较快地转发数据包,对控制平面饱和攻击产生有效的防御.

6.2.4 交换机迁移代价

在迁移过程中,控制器会将处于任务队列中的Packet_In消息请求暂时搁置,优先处理迁移请求,这一过程会产生额外的时延开销,延后流量转发时间,影响服务质量.与6.2.3中的实验环境一致,我们同样进行了10次重复实验,监控了不同方法下执行的迁移动作,求取了监控期间内不同方法执行迁移操作的平均次数和执行迁移的时间代价,实验结果如图12和图13所示.

在不同的网络拓扑下,Balcon策略的执行次数都略大于饱和次数,这是由于Balcon策略是一种集群迁移策略,饱和后可能执行多于一次的迁移操作.在面对树状网络时,转发路径重合度高,各类重要交换机间负载差距较小.Rebalance策略在面对树状网络时,容易陷入局部最优,经过几次迁移之后,系统无法找到负载标准差更小的交换机迁移方式,系统维持原样,从而呈现出迁移次数远远小于控制器饱和次数.作为单目标迁移的方法,本文所提出方法的迁移次数都是小于饱和次数的,这与方法设定的最坏适应匹配有关,当迁移交换机与辅助控制器集群匹配失败时,将不执行迁移,以避免级联故障,所以会呈现图12中所示的结果.不同方法在不同网络类型中的表现各不相同,但从两种拓扑的迁移总次数上来看,本文所提出的方法相较于其他两种分别低81.1%和11.4%,表明了该方法能够通过更少的迁移操作,完成控制器的饱和缓解.

迁移时间开销是从发生饱和开始,到所有控制器收到角色更改回复消息,交换机重新恢复流量转发功能的时间开销.相较于网状网络,树状网络拓扑简单,迁移交换机的选择过程速度较快,所以迁移

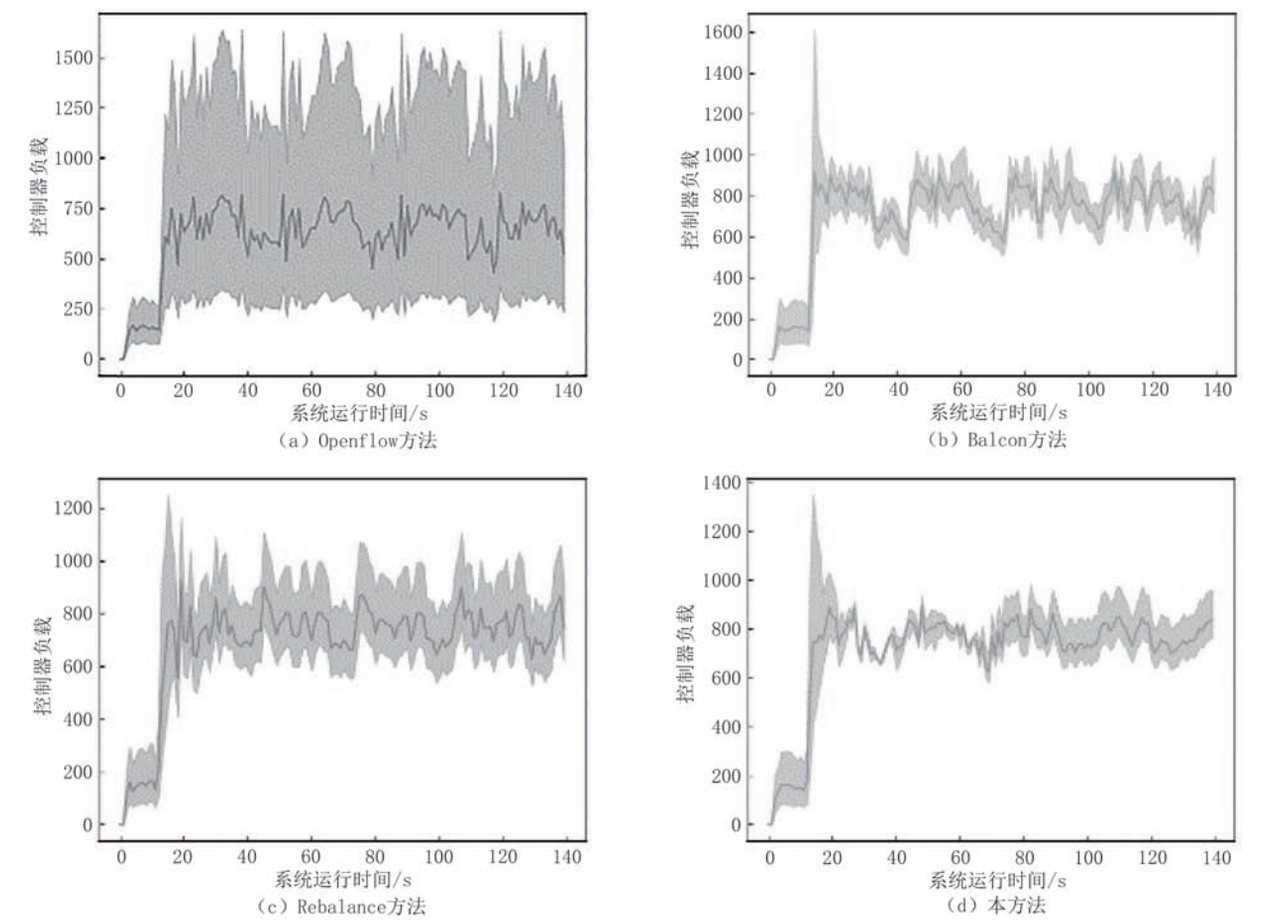


图10　树状网络下不同方法下控制器集群的负载情况

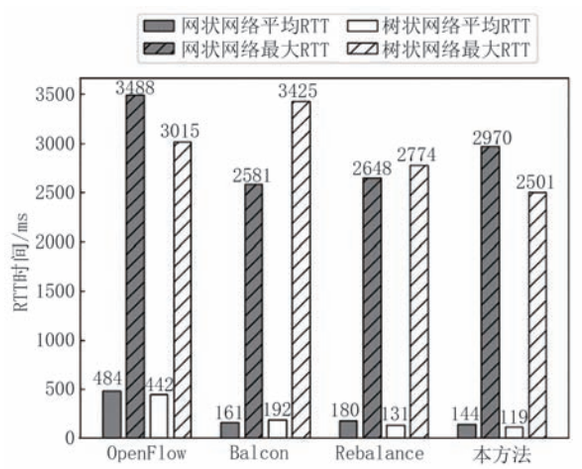


图11　合法流量往返时延

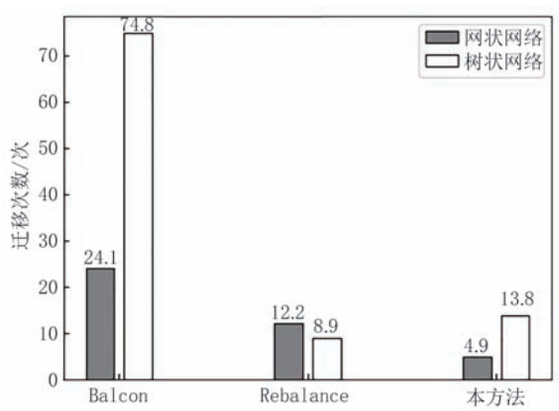


图12　不同方法的迁移次数

时间开销都略低. 对于不同的方法而言，Balcon策略由最大速率交换机开始构建迁移集群，往往在搜索1~3个交换机后，就可以完成迁移交换机的选择. Rebalance策略基于贪心算法，需要找到使得控制器负载标准差最小的迁移方法，所以该策略的时间开销最大. 本文所提出的方法，需要遍历饱和域内的所有交换机，并将其与辅助控制器队列进行最坏适应匹配，所以时间开销会略大于Balcon策略. 如图13所示，本方法所具有的单次时间开销最多比Balcon策略高出0.28秒，但从综合迁移次数与时间开销的总开销情况来看，本方法分别比Balcon策略少61.3%和73.9%. 通过迁移次数和迁移时间开销可以看出，本方法相较于其他两种方法具有较低的迁移代价，能够保证对流量转发过程造成较低的时延影响，维护网络的鲁棒性.

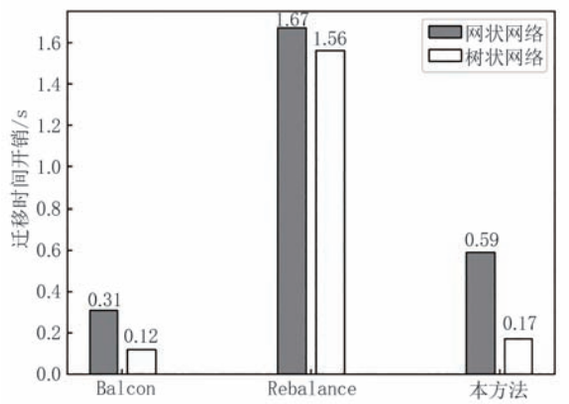


图13 不同方法的迁移时间开销

7 总 结

控制平面饱和攻击通过向控制器发送大量经过伪造的数据包，占据请求队列，迫使控制器处理无用的数据包，大量消耗计算资源，导致合法请求无法获得响应，交换机无法及时收到流规则命令，最终丢弃合法流量．作为DDoS攻击的特殊实践，控制平面饱和攻击具有隐蔽性高、破坏力强等特点，已经成为SDN网络中最大的安全威胁．

本文针对控制平面饱和攻击，提出了一种基于交换机的迁移防御方法．基于控制器负载阈值检测饱和，并通过流表统计特征和空间特征图，准确选择攻击路径上的交换机，根据交换机的负载贡献度选择合适的辅助控制器进行迁移，从而达到降低饱和控制器负载的目的．实验结果表明，本方法在面对控制平面饱和攻击时，在控制器防御效果和交换机迁移上都有较高的表现．目前，本方法仅采用了主动防御的方法，在之后的研究中可将交换机迁移向专用的流量清洗控制器，实现主被动防御结合的方法，进一步提高网络对控制平面饱和攻击的防御力．

参 考 文 献

[1] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74

[2] Zhu L, Karim M M, Sharif K, et al. SDN controllers: A comprehensive analysis and performance evaluation study. ACM Computing Surveys , 2020, 53(6): 1-40

[3] Yao G, Bi J, Guo L. On the cascading failures of multi-controllers in software defined networks//Proceedings of the 21st IEEE International Conference on Network Protocols. Gttingen, Germany, 2013: 1-2

[4] Yue M, Yan Q, Zheng H, et al. Cross-plane DDoS attack defense architecture based on flow table features in SDN. Security and Communication Networks, 2022, 2022 (1) : 7409083

[5] Li Z, Meng W. Mind the amplification: Cracking content delivery networks via DDoS attacks// Proceedings of the Wireless Algorithms, Systems, and Applications: 16th International Conference, WASA 2021. Nanjing, China. 2021, Part II 16: 186-197

[6] Zhou Y, Cheng G, Yu S. An SDN-enabled proactive defense framework for DDoS mitigation in IoT networks. IEEE Transactions on Information Forensics and Security, 2021, 16: 5366-5380

[7] Zheng J, Li Q, Gu G, et al. Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis. IEEE Transactions on Information Forensics and Security, 2018, 13 (7): 1838-1853

[8] Cao Y, Jiang H, Deng Y, et al. Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network. IEEE Transactions on Dependable and Secure Computing, 2021, 19(6): 3855-3872

[9] Xu J, Wang L, Xu Z. An enhanced saturation attack and its mitigation mechanism in software-defined networking. Computer Networks, 2020, 169: 107092

[10] Kaur S, Kumar K, Aggarwal N, et al. A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions. Computers & Security, 2021, 110: 102423

[11] Li Z, Xing W, Khamaiseh S, et al. Detecting saturation attacks based on self-similarity of OpenFlow traffic. IEEE Transactions on Network and Service Management, 2019, 17(1): 607-621

[12] Mousavi S M, St-Hilaire M. Early detection of DDoS attacks against SDN controllers// Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC). Garden Grove, USA, 2015: 77-81

[13] Huang X, Xue K, Xing Y, et al. FSDM: Fast recovery saturation attack detection and mitigation framework in SDN// Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). Delhi, India, 2020: 329-337

[14] Wang R, Jia Z, Ju L. An entropy-based distributed DDoS detection mechanism in software-defined networking// Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA. Helsinki, Finland, 2015, 1: 310-317

[15] Kumar P, Tripathi M, Nehra A, et al. SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN. IEEE Transactions on Network and Service Management, 2018, 15(4): 1545-1559

[16] Wang W, Ke X, Wang L. A HMM-R approach to detect L-DDoS attack adaptively on SDN controller. Future Internet, 2018, 10(9): 83

[17] Mishra A, Gupta N, Gupta B B. Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX

controller. Telecommunication Systems, 2021, 77: 47-62
[18] Ye J, Cheng X, Zhu J, et al. A DDoS attack detection method based on SVM in software defined network. Security and Communication Networks, 2018, 2018(1): 9804061
[19] Xu Y, Sun H, Xiang F, et al. Efficient DDoS detection based on K-FKNN in software defined networks. IEEE Access, 2019, 7: 160536-160545
[20] Dong S, Sarem M. DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. IEEE Access, 2019, 8: 5039-5048
[21] Gao D, Liu Z, Liu Y, et al. Defending against Packet-In messages flooding attack under SDN context. Soft Computing, 2018, 22: 6797-6809
[22] Yuan B, Zhang F, Wan J, et al. Resource investment for DDoS attack resistant SDN: A practical assessment. Science China Information Sciences, 2023, 66(7): 172103
[23] Dixit A A, Hao F, Mukherjee S, et al. Elasticon: An elastic distributed SDN controller//Proceedings of the 10th ACM/IEEE Symposium on Architectures for Networking and Communications Systems. Marina del Rey, USA, 2014: 17-28
[24] Dai Y, Wang A, Guo Y, et al. Elastically augmenting the control-path throughput in SDN to deal with Internet DDoS attacks. ACM Transactions on Internet Technology, 2023, 23(1): 1-25
[25] Wang Y, Hu T, Tang G, et al. SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking. IEEE Access, 2019, 7: 34699-34710
[26] Macedo R, de Castro R, Santos A, et al. Self-organized SDN controller cluster conformations against DDoS attacks effects// Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM). Washington, USA, 2016: 1-6
[27] Yuan B, Zhao H, Lin C, et al. Minimizing financial cost of DDoS attack defense in clouds with fine-grained resource management. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2541-2554
[28] Lai W K, Wang Y C, Chen Y C, et al. TSSM: Time-sharing switch migration to balance loads of distributed SDN controllers. IEEE Transactions on Network and Service Management, 2022, 19(2): 1585-1597
[29] Tootoonchian A, Gorbunov S, Ganjali Y, et al. On Controller Performance in Software-Defined Networks// Proceedings of the 2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. San Jose, USA, 2012
[30] Min Z, Hua Q, Jihong Z. Dynamic switch migration algorithm with Q-learning towards scalable SDN control plane// Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP). Nanjing, China, 2017: 1-4
[31] Sahoo K S, Puthal D, Tiwary M, et al. ESMLB: Efficient switch migration-based load balancing for multicontroller SDN in IoT. IEEE Internet of Things Journal, 2019, 7(7): 5852-5860
[32] Xu Y, Cello M, Wang I C, et al. Dynamic switch migration in distributed software-defined networks to achieve controller load balance. IEEE Journal on Selected Areas in Communications, 2019, 37(3): 515-529
[33] Liu Y, Gu H, Yan F, et al. Highly-efficient switch migration for controller load balancing in elastic optical inter-datacenter networks. IEEE Journal on Selected Areas in Communications, 2021, 39(9): 2748-2761
[34] Deng S, Gao X, Lu Z, et al. DoS vulnerabilities and mitigation strategies in software-defined networks. Journal of Network and Computer Applications, 2019, 125: 209-219
[35] Yuan B, Zou D, Jin H, et al. HostWatcher: Protecting hosts in cloud data centers through software-defined networking. Future Generation Computer Systems, 2020, 105: 964-972
[36] Liu Q, Zhang JH, Hu T, Zhao W. Controller load balancing mechanism based on biological implications in SDN. Journal of Software, 2017,28(Suppl.(2)):50－60 (in Chinese)
(刘强,张建辉,胡涛,赵伟. SDN中基于生物启示的控制器负载均衡机制.软件学报,2017,28(s2):50-60.)
[37] Zhang B, Wang X, Huang M. Multi-objective optimization controller placement problem in internet-oriented software defined network. Computer Communications, 2018, 123: 24-35

GUAN Ji-Zhe, M. S. candidate. His main interests are software defined network, DDoS mitigation and proactive defense.

CHENG Guang, Ph. D., professor. His main research areas include network security, network measurement and traffic behavior analysis.

ZHOU Yu-Yang, Ph. D., research assistant. His main research areas include moving target defense, DDoS mitigation, intrusion detection and Android malware detection.

Background

Software Defined Network (SDN) is a new network architecture. The distributed deployment method improves the availability and scalability of SDN. However, in the face of saturation attack on the control plane, there are still risks of single point of failure and cascading failures. At present, the defense methods against this attack have achieved good results, but they are insufficient in resource utilization and load balancing. Therefore, in this paper, we propose a control plane saturation attack defense method based on switch migration. Firstly, based on the load threshold of the controller, attacks are detected. Then, by using flow table statistics and spatial feature maps, switches on the attack path are accurately selected, and appropriate auxiliary controllers are selected for migration based on the load contribution of the switches, thereby achieving the goal of reducing saturated controller load. Experimental results show that this method can effectively defend against control plane saturation attack, and has excellent performance in resource utilization and load balancing.

This topic belongs to the research of moving target defense technology for LDDoS attacks in edge computing environment. This project is oriented to the requirements of active, proactive and efficient defense of LDDoS attacks in large-scale edge computing environment, with the core goal of improving the dynamic defense effectiveness of complex heterogeneous, resource constrained, delay sensitive edge scenarios and ensuring the maintenance of application quality of service under LDDoS attacks. The research results of the project can be applied to the Internet of Things, the Vehicle-to-everything, intelligent manufacturing, smart cities and other fields in the later stage, which can provide important support and capability guarantee for the security protection of large-scale edge computing scenarios, and has good application prospects.

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62172093, Grant No. 62202097, and Grant No. U22B2025, in part by the China Postdoctoral Science Foundation under Grant No. 2024T170143 and Grant No. 2022M710677, and in part by the Jiangsu Funding Program for Excellent Postdoctoral Talent under Grant No. 2022ZB137.

The research team has focused on DDoS mitigation for years, and some papers in this field have published in highly-ranked journals.