SDN 控制层泛洪防御机制研究：检测与缓解

周启钊[1]，于俊清[1,2]，李冬[2]
（1. 华中科技大学计算机学院，湖北 武汉 430074；2. 华中科技大学网络与计算中心，湖北 武汉 430074）

摘 要：针对 SDN 控制层中的欺骗式泛洪防御问题，提出控制器防御机制（CDM），主要包括基于关键特征多分类的泛洪检测机制和基于 SAVI 的泛洪缓解机制 2 个方面。在泛洪检测方面提出控制层泛洪关键特征解析模块，利用 Boosting 算法将各个关键特征弱分类器加权叠加形成增强型分类器，通过不断降低计算中的残差，达到更准确分类针对控制层的欺骗式泛洪攻击的效果。在泛洪缓解方面，CDM 部署基于 SAVI 的泛洪缓解机制，以绑定和验证的模式为基础执行泛洪数据包的路径过滤，同时以动态轮询的模式实现泛洪攻击安全保障和接入层交换机泛洪关键特征数据的更新，降低冗余的模型更新负载。实验结果表明，所提方法具备开销低、精度高的特点，有效地增加了控制层的安全性，减少了欺骗式泛洪攻击主机分类的时间和对应控制器 CPU 的消耗。

关键词：软件定义网络；控制层防护；泛洪检测；源地址验证
中图分类号：TP393
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021191

Research on flood defense mechanism of SDN control layer: detection and mitigation

ZHOU Qizhao[1], YU Junqing[1,2], LI Dong[2]
1. College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
2. Center of Network and Computation, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract: Aiming at the problem of spoofing flood defense in the control layer of SDN, a controller defense mechanism (CDM)was proposed, including a flood detection mechanism based on key features multi-classification and a flood mitigation mechanism based on SAVI. The flood feature analysis module of the control layer was designed for flood detection, and boosting algorithm was used to overlay each feature weak classifier to form an enhanced classifier, which can achieve more accurate classification spoofing flooding attack effect by continuously reducing the residual in the calculation. In CDM, a flood mitigation mechanism based on SAVI was deployed to realize flood mitigation, which performed flood packet path filtering based on binding-verification mode, and updated the flood features of access layer switches with dynamic polling mode to reduce redundant model update load. The experimental results show that the proposed method has the characteristics of low overhead and high precision. CDM effectively increases the security of the control layer, and reduces the time of host classification of spoofing flood attack and the CPU consumption of corresponding controller.

Keywords: software defined network, control layer protection, flood detection, source address validation

1 引言

随着互联网的资源管理需求日益多元化，软件定义网络（SDN, software defined network）[1]作为一种新兴的转控分离的架构为网络资源的管理带来了新的思路。SDN 通过将网络核心控制逻辑与底

收稿日期：2021−07−02；**修回日期**：2021−09−13
基金项目：国家重点研发计划基金资助项目（No.2018YFB1800405）
Foundation Item: The National Key Research and Development Program of China (No.2018YFB1800405)

层路由器和交换机架构进行分离，打破了传统的垂直集成网络控制和转发的模式，克服了网络基础设施部署局限性的关键问题。部署 SDN 架构的网络具备灵活性、可编程性和集中化管理等诸多优点[2]。OpenFlow 是当前最常用的 SDN 南向协议，它提供了标准化的控制层与数据层的通信指令：数据转发规则以流表项的形式进行表示，流表项则组合成为数据层的流表，SDN 架构的网络环境以流表更新及控制器反馈的形式完成通信指令交互。由于控制层为了维持与数据平面之间的高效通信引入了诸如 Packet-In 的快速信息流指令，利用现有的控制层信息指令漏洞对网络发起攻击的案例层出不穷，其中对 SDN 影响最严重的就是控制层欺骗式泛洪攻击。包括 SDN 控制层信息指令 Packet-In、通信交互 TCP/SYN 报文在内的信息流均可能被利用并以泛洪的形式发起攻击[3-4]。攻击者通过伪造源地址技术创造新的源地址或数据通道，并利用伪造控制层交互信息来与 SDN 控制器进行通信，扰乱控制层对全网的认知，进而间接对数据层的转发产生影响。根据 OpenFlow 协议的规定，控制器下发的流表项规则被交换机完全信任，若存在伪造源地址的流规则篡改行为，SDN 数据中心接入层交换机的安全保护性能将面临严重威胁。

现有基于目的地址转发的网络路由机制导致控制层泛洪中伪造源地址欺骗的现象层出不穷，其关键特点为攻击流量巨大、难以追溯和难以防御等，若缺乏高效的泛洪攻击检测机制，SDN 的控制层将存在巨大安全隐患。另一方面，由于 SDN 基于目的地址的转发模式并未涉及对源地址的检查和认证过程，控制层泛洪所造成的网络管理扰乱了接入层交换机的认证与身份识别，若缺乏合理的泛洪攻击缓解机制，攻击者可实现对 SDN 数据的窃取和网络状态的探测，对 SDN 架构的安全性产生重大影响。

2 研究背景及目的

2.1 控制层欺骗式泛洪攻击

OpenFlow 协议提供了控制器和交换机之间的安全通信指令，其规定的标准化机构使各模块网络通信间的互操作性增强，但针对控制层的欺骗式泛洪攻击仍然层出不穷。典型的控制层欺骗式泛洪攻击的类型主要包括 Packet-In 泛洪和 SYN（synchronize sequence numbers）泛洪 2 类。

每当有新的数据包进入数据层进行匹配，会在流表中寻找对应的流表项进行匹配：若直接匹配成功，则转发数据报文；否则，该数据包信息通过 OpenFlow 协议规定的控制器与数据层交互信息流 Packet-In 进行上报，由控制器进行进一步分析。当控制层和数据层建立连接后，控制器即处理来自数据层的各种 OpenFlow 协议通信指令，如图 1 所示，并分发指令给监听此通信指令的所有数据层交换机。在此过程中，由于控制器的集中特性，攻击者通过创造大量伪造源地址的 Packet-In 包触发控制器处理进程，产生 Packet-In 泛洪[4]。该 Packet-In 泛洪是一类新型的针对 SDN 控制层的攻击，将直接造成 SDN 的单点故障。在极端情况下，Packet-In 泛洪持续时间过长使得控制层的性能完全失效，进而使控制器无法处理正常消息，南向数据层网络管理混乱，SDN 的合法流量转发滞塞。

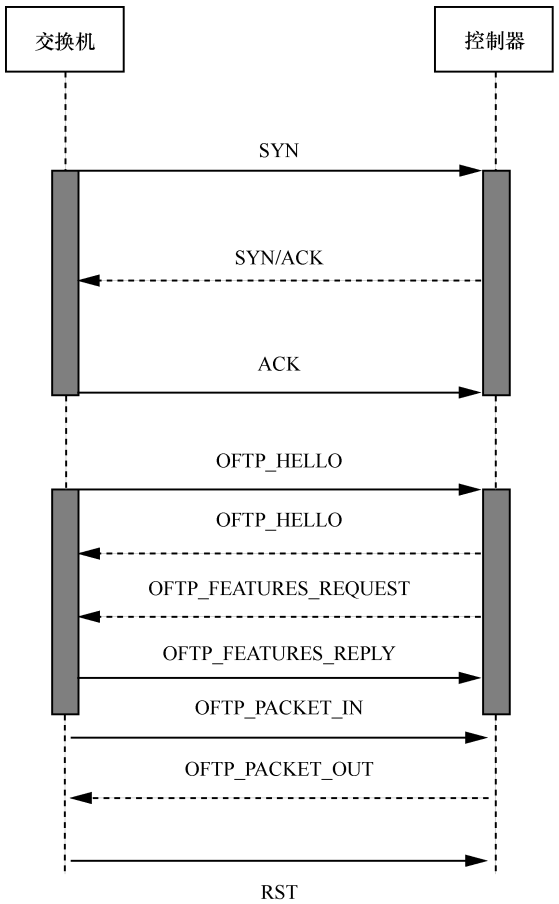

图 1 控制层 Packet-In 响应信令交互模式

由于控制层针对传统 TCP 漏洞仍然存在缺陷，依靠 TCP 建立连接时 3 次握手存在的缺陷可发起

SYN 泛洪攻击，该泛洪的形成借助 TCB（即 TCP 传输控制块）的缺失和在 SYN 包中伪装合法的源 IP 地址发起，可对 SDN 控制层造成巨大安全隐患，如图 2 所示。TCB 是一种连接所有信息的传输协议数据结构，其分配空间的大小取决于接收的 SYN 包，在控制层连接成功前或发起源被验证前该空间大小均可变[3]。攻击者可借此漏洞轻易发起欺骗式 SYN 泛洪，使到达的 SYN 包被控制层分配过多的 TCB 而导致其内核内存被耗尽。此外，攻击者通过在 SYN 包中伪装合法的源 IP 地址，使 SYN-ACK 包无法被有效响应，进而无法触发对应控制层通信指令，使主机将已分配的 TCB 从 SYN-RECEVIED 状态队列中移除，最终导致 SDN 控制层的安全通信指令管控混乱，无法继续响应数据层其他正常设备的请求。

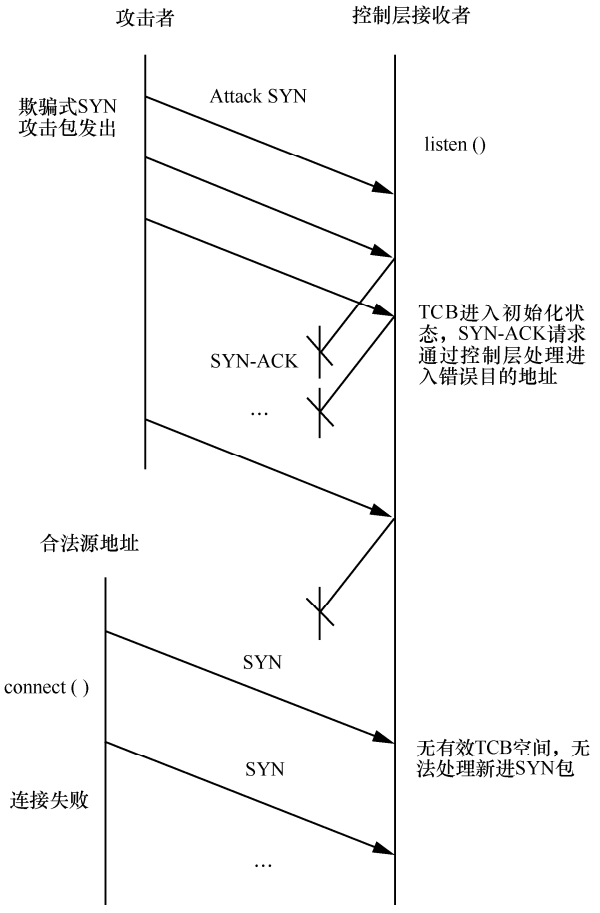

图 2　控制层 SYN 泛洪示意

2.2　现有方案存在问题

现有针对 SDN 控制层欺骗式泛洪的解决方案主要分为基于终端的 SYN 泛洪防御和基于网络的 Packet-In 泛洪防御 2 类。针对 SYN 泛洪，基于终端的对策包括 SYN Cookies 和 SYN 缓存等方法[4-6]。由于欺骗式 SYN 泛洪依赖于终端主机连接套接字的日志溢出，增加其日志队列大小可初步缓解 SYN 泛洪的攻击强度。此外，缩短一个 TCB 从进入 SYN-RECEIVED 状态到因未进入下一个状态而被回收的时间，也是一类有效的基于终端主机的解决方法[7-8]。为了缓解 ACK（acknowledge character）包丢包而产生的泛洪攻击问题，基于 SYN 缓存和 SYN Cookies 实现的方案均提供了针对控制层的防御思路：在已部署 SYN 缓存和 SYN Cookies 的控制器中，添加一个被限制大小的 Hash 表空间用于存放被分配给 TCB 的数据的指令子集，该空间能提升 SYN 泛洪发生时 SDN 控制层的容错率，在一定时间内维持控制层的全局视野和管理性能。然而，此类基于终端的 SYN 泛洪防御机制需要对其底层 TCP/IP 栈实现进行修改，而中间件如防火墙或入侵检测系统（IDS, intrusion detection system）[9]需要通过网关实现，其在 SDN 中的可扩展性相对不足。

基于网络的泛洪防御方案通常以 RFC2827 为基础实现欺骗数据包过滤，采取输入源过滤的方式执行泛洪防御。TopoGuard[10]是一种针对拓扑欺骗攻击的工具。通过验证 SDN 通信信令从出现到执行的合法性确保主机迁移的真实性。然而，该模式还缺乏对控制层泛洪尤其是 Packet-In 泛洪的测试，控制层泛洪防御模式并不全面。ISP 方案[11]直接阻断了源 IP 地址不属于源子网的包的传递，该方案提出了在网络层筛选欺骗式泛洪数据包并进行输入源过滤的模式，实验结果表明其部署能有效地过滤 SYN 泛洪攻击包，但该方法依然未考虑大量 Packet-In 消息注入的情况。为了提升控制器 Packet-In 泛洪对应的防御性能，文献[4]提出了一种基于 Packet-In 合法性检测的防御策略，通过验证 Packet-In 消息是否由伪造源地址方式生成来决定其向控制器的转发功能。然而，该方法的攻击缓解部分缺乏对控制层泛洪攻击中伪造 MAC 地址的情况的分析，其伪造源地址包过滤的精度还不够。此外，一些针对解决 SDN 控制层泛洪攻击问题提出的统计分析及阈值检测、特征检测及深度学习等方法的防御机制[12-14]弥补了现有泛洪防御模型在分类攻击主机和数据包方面精度不高的问题。其中统计分析及阈值检测方法能初步对 SDN 控制层泛洪的发生进行预警，但该类方法无法及时并有效地区

分网络转发正常突发大流和欺骗式泛洪攻击，针对控制层泛洪检测问题容易发生误报。特征检测及深度学习的方法[12]通过集合分析 SDN 中交换机、控制器及流量数据等多维特征，结合轻量级的机器学习或深度学习方法进行泛洪检测，该类型的方法由于特征维度爆炸或特征关联性问题通常需要进行大量计算，在 SDN 控制层泛洪攻击实际环境下，尤其针对 Packet-In 泛洪及 SYN 泛洪的检测效果不理想。此外，大量特征的采集和处理也增加了控制器的负载，对数据层的正常数据包转发产生了负面的影响。

针对现有基于终端的 SYN 泛洪防御和基于网络的 Packet-In 泛洪防御存在的可扩展性不足、特征维度爆炸和带来额外负载的问题，本文提出了一种低开销、高精度的 SDN 控制器防御机制（CDM，controller defense mechanism）。该方法具有以下优点。

1) 低开销的流量采集。结合 sFlow 与 OpenFlow 交互信令的共性，提出了一种轻量级、低开销的特征采集方案，使控制器以多线程的方式周期性地从接入层交换机获取针对性的特征条目，而不需要多次遍历和轮询。

2) 高精度的特征解析和攻击检测。将与 SDN 控制层泛洪攻击密切相关的特征进行组合，从多维流表项匹配域及流量特征中筛选排除低效特征，并结合梯度决策分类算法训练模型，该模型可高精度地区分正常突发流量和恶意泛洪流量。

3) 高安全性的差异化泛洪缓解。结合源地址验证绑定模式，针对欺骗式泛洪和正常突发流量的特征提出差异化的 SAVI（source address validation improvement）泛洪缓解机制，该方法不仅可有效抵御 Packet-In 泛洪和 SYN 泛洪攻击，也能有效提升缓解控制层在泛洪发生时的响应效率。

3 控制层泛洪检测机制

3.1 基于信令交互的特征采集

由于 SDN 提供了灵活的网络全局跟踪机制，同时具备交换机信息持续监控的能力，本节针对控制层泛洪攻击检测低开销的关键特征采集需求，结合 sFlow 与 OpenFlow 交互信令的共性，实现了一个轻量级的泛洪特征采集模块。sFlow（RFC3176）[15]是一种网络导出协议，通常用于网络性能和数据的测量，协议提供了对数据包进行检测的各类信息采集协议，并嵌入专用集成芯片（ASIC）中对数据包进行转发和收集。如图 3 所示，基于 sFlow 的数据采集系统主要由 ASIC 中的 sFlow 服务端（sFlow agent）和远程的 sFlow 采集端（sFlow collector）2 部分组成。其中，sFlow 服务端用于获取交换机或路由器的网络数据测量结果，当特定时间窗口结束或者缓冲区满后，将数据测量结果打包封装为 sFlow 的报文发送到 sFlow 采集端。随后，sFlow 采集端对 sFlow 的报文进行解析，分析并输出对应的网络统计数据[16]。

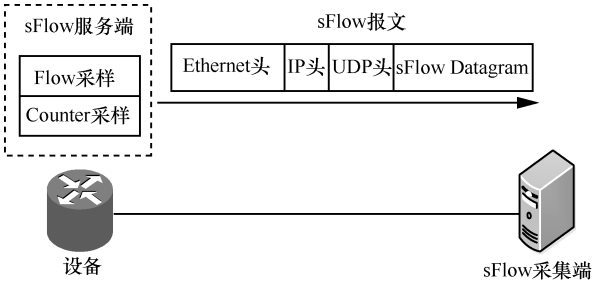

图 3 基于 sFlow 的数据采集系统示意

而在 OpenFlow 协议中，SDN 控制器通过 OFPT_STATS_REQUEST 对交换机进行请求。交换机将所请求的统计数据信息通过 OFPT_ MULTI-PART_ REPLY 消息回复给 SDN 控制器。结合 sFlow 服务端的采集特性，该模块将现有 OFPT_MULTIPART_REPLY 消息可获取的部分进行保留，利用控制器定期通过交互信息 OFPT_STATS_REQUEST/REPLY 与 OpenFlow 交换机进行通信的特性，将 sFlow 服务端 IP/UDP 头与 OFPT_STATS_REQUEST 消息返回流表匹配域信息重合的部分进行整合。sFlow 服务端整合流表匹配域数据采集示例如图 4 所示，IP/UDP 头包括数据包长、数据包缓存字节、流表空间大小等，为每个接入层交换机单独创建一个数据收集线程并开始执行，这缩减了现有 SDN 接入层交换机遍历采集模式的冗余数据特征处理流程[16]。采用 Java/Python 实现基于信令交互的多线程特征采集，其中 OFPT_STATS_REQUEST 主要负责查询流表的最大存储量和活跃流表项的数目，另一线程的 sFlow 服务端主要用于获取流表对应数据流量基本信息。基于 sFlow 的多线程信息采集与整合模型流程如图 5 所示，通过动态的参数控制获取交换机流表和流量 2 方面的信息，为了确保两者的兼容性，sFlow 服务端仅接收 SDN 数据中心接入层交换机流表对应的数据流信息采集和存储。

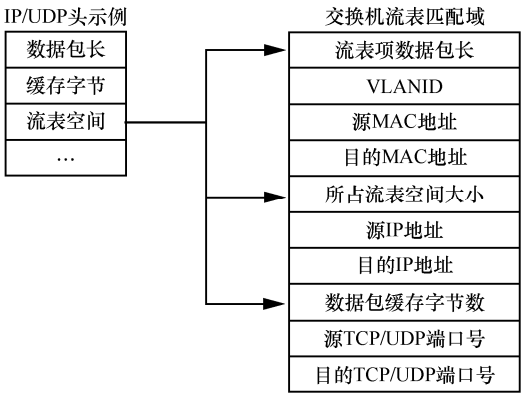

图 4　sFlow 服务端整合流表匹配域数据采集示例

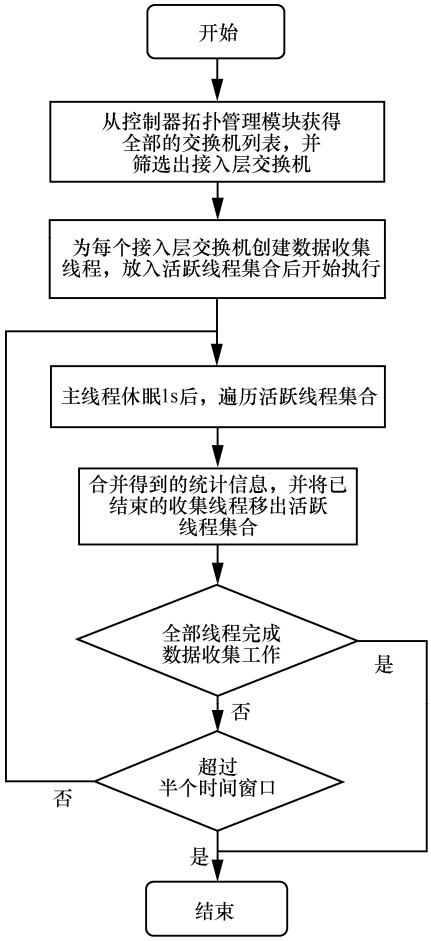

图 5　基于 sFlow 的多线程信息采集与整合模型流程

3.2　控制层泛洪特征解析

3.2.1　控制层负载单位阶跃函数

针对 SDN 控制层的 SYN 泛洪攻击可利用因拥塞而丢包的 ACK-SYN 或者握手完成的 ACK 包单独发起，此时由于 SYN-RECEIVED 的响应时间减少，合法连接的 TCB 空间也会因控制器繁忙无法重传失配包，最终导致数据包丢弃。为了解决控制层 SYN 泛洪攻击问题，现有常见的防御机制通过优化系统设置实现[8]，典型的如降低对应的 SYN 时延[17]，使主机尽快释放半连接的占用。该模式存在的主要问题是无法根据数据流状态衡量控制层负载，因此为了量化 SDN 突发大流或泛洪攻击发生时控制层的负载状态，结合现有 OpenFlow 协议中顺序的匹配模式，本节基于数据流对应流表项时延 idle_timeout（T_{init}）对控制层负载进行建模[18]。SDN 流表时延与控制器状态模型如图 6 所示，其中 $\{p_1,p_2,\cdots,p_i,\cdots,p_n\}$ 表示 SDN 数据流传输的数据包长度，此处认为数据包是独立分布并服从 Pareto 分布的[19]，分布公式如式(1)所示，其中，k,τ,α 是 Pareto 分布的参数。以此数据流分布为基础，可从理论角度分析数据流分布与控制层负载之间的关系。假设 SDN 数据流数目为无穷大，若用 $\{t_1,t_2,\cdots,t_i,\cdots,t_n\}$ 分割数据包之间的传输间隔，则该间隔服从负指数分布，如式(2)所示。

$$p(p_i<\tau)=\left(\frac{k}{\tau}\right)^{\alpha},\overline{p_i}=\frac{k\alpha}{\alpha-1} \tag{1}$$

$$t_i\sim\exp(\lambda),\overline{t_i}=\lambda^{-1}(\lambda\geqslant 0) \tag{2}$$

若数据包的空闲时间间隔较大，则 OpenFlow 交换机此段间隔时间内必然会被控制器重新调度，此数据流所对应配置的流表项也会随之到期删除。当后续的数据包需要进行流表匹配时，由于找不到对应的流表项，交换机会立即向控制器发送 Packet-In 消息请求流表项的下发。在此期间，控制层泛洪造成的安全问题主要可表征为 2 个关键的特征：控制层异常发生时的负载高低与实际连接的端口数目有关，如式(3)所示；由于匹配失败产生的 Packet-In 消息会耗费控制器大量的计算资源，以控制器处理 Packet-In 消息的性能为依据，可引入单位阶跃函数 $H(\cdot)$对控制器的负载状态即处理 Packet-In 消息的数量进行量化，如式(4)所示。

$$\text{GPP}=\frac{\text{Num}_{\text{ports}}}{T_{\text{init}}} \tag{3}$$

$$\text{Packet\_In}_{\text{num}}(t,T_{\text{init}})=\lim_{n\to\infty}\sum_{i=1}^{n}H(t_i-T_{\text{init}}) \tag{4}$$

单位阶跃函数代表控制器的负载与 Packet-In 消息的数量呈正相关，假设 SDN 控制器处理一条 Packet-In 消息的负载为 cost，为了同时降低流表的失配率，总负载值可通过多个数据包匹配流表初始时延值累加进行估算。根据数据流切割传输模式的

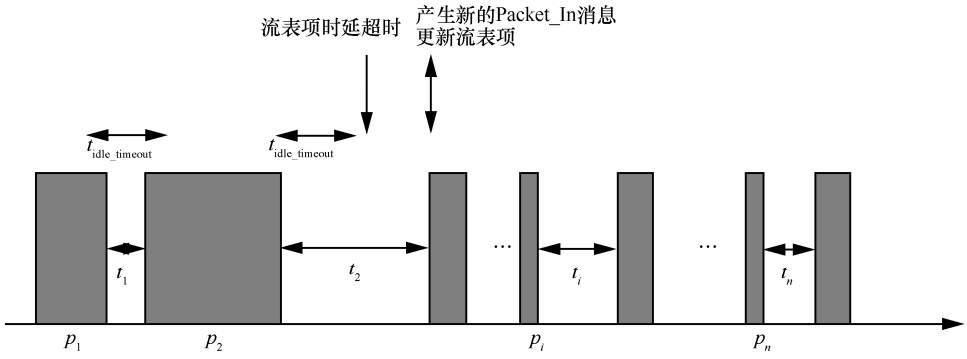

图 6 SDN 流表时延与控制器状态模型

特性，设多个数据包对应数据流切割分批处理过程中被分成了 n 个数据包，每转发一个数据包，其对应流表项均需要与数据包进行一次完整匹配，故数据流切割传输模式下东西向数据包传输平均造成的 SDN 控制器负载关键特征可由式(5)计算。

$$\mathrm{BPF}=\lim_{n\to\infty}\frac{\sum_{i=1}^{n}H(t_i-T_{\text{init}})\cos t}{n}=\cos t\cdot \mathrm{e}^{-\lambda T_{\text{init}}} \tag{5}$$

3.2.2 数据层端口流量差及信息熵

现有研究表明，针对控制层的欺骗式泛洪具有 2 个重要特征：突发性流量大和流量速率倾斜[3]。因此通过对 SDN 接入层交换机端口流量信息的分析，可初步鉴别 SDN 大流量的异常状态。端口出入流量差的绝对值常常被用于表征影响 SDN 控制层流量的关键因子，其计算方法如式(6)所示。

$$\mathrm{DTB}=\frac{\sum_{n}\mathrm{Traffic\_in}-\sum_{n}\mathrm{Traffic\_out}}{T_{\text{init}}} \tag{6}$$

现有方法以接入层交换机端口的出入流量差的绝对值为标准进行数据流鉴别，其模型判定依赖流量阈值的选择，而该流量阈值通常随着网络应用需求、底层拓扑和设备性能的变化而差异极大，在不同的拓扑规模和设备性能状态下对异常流量的判定和端口的定位均会存在严重偏差，其可扩展性不足。为了更好地确定异常流警报的阈值，通过流量差和熵值和转化模型可定性流量的随机程度，而结合 φ-熵对流量差进行表示也有助于量化阈值，提升其稳定性。假设 SDN 接入层交换机端口流量差为 x，其取值集合可用 $X=\{x_1,x_2,\cdots,x_n\}$ 表示，针对每个流量差取值的概率分布可用 $P=\{p_1,p_2,\cdots,p_n\}$ 表示，且每个取值之间互不影响。其中，$\sum_{i=1}^{n}p_i=1, 0\leqslant p_i\leqslant 1$，变量 x 的信息熵为

$$H=-\sum_{i=1}^{n}p_i\log(p_i) \tag{7}$$

对于 SDN 接入层交换机端口流量差集合 X，其 φ-熵可表示为

$$\mathrm{EPF}=-\frac{1}{\sinh(\varphi)}\left(\sum_{i=1}^{n}p_i\sinh(\varphi\mathrm{lb}p_i)\right) \tag{8}$$

φ-熵可用于更加精确量化主机各维度的出入流量差的绝对值与 Packet-In 泛洪攻击之间的关系。本节模型多维度 φ-熵特征包括源 IP 地址信息（sIP）、源端口信息（sPort）、目的 IP 地址信息（dIP）和目的端口信息（dPort）。为了对异常流量进行判定，需要保存前几个连续时间窗口的多维度 φ-熵。如图 7 所示，若主机 h_1 对应的端口突然产生了异常大流量，此时该模型应该对 s_1 对应 h_1 的端口 p_1 实时的流量差熵值进行计算，通过出入流量标准差值转化 φ-熵评估，判定 h_1 发出的流量是正常通信大流量或泛洪攻击流量。当该流量为欺骗式泛洪异常大流量时，即出入流量标准差值转化 φ-熵超过阈值，对应出入端口流量失衡，则判定为疑似泛洪攻击流量；而若 h_1 主机属于正常的流量密集型应用，即出入流量标准差值转化 φ-熵未超过阈值，对应出入端口流量平衡，则不会被判定为疑似泛洪攻击流量。

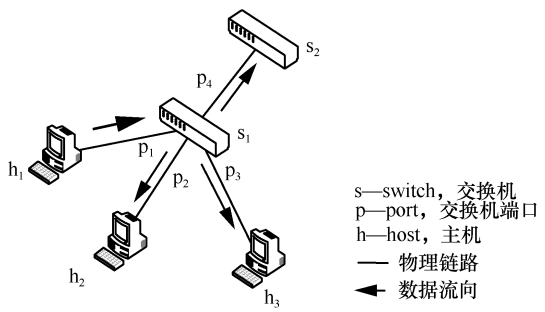

图 7 SDN 接入层交换机端口数据流量变化拓扑示意

3.2.3　欺骗式泛洪指令流状态

分析控制层欺骗式 SYN 泛洪攻击的特征，通常在一定的时间间隔内，攻击者在 SYN 包中伪装合法的源 IP 地址，而这个 IP 地址将不能响应 SYN-ACK 包，此时就无法触发控制层通信指令使主机将已分配的 TCB 从 SYN-RECEVIED 状态队列中移除。分析该 SYN-RECEVIED 状态队列对应流表项与数据流中包间隔大小的关系，发现数据包间隔较大的数据流往往需要占用更多的转发时间和更大的网络带宽，实际的突发大流占总流量的比例较低[17]。由于 SDN 数据流的切割传输模式，数据包间隔较大的数据流分割得到的数据包对应流表项包含的转发目的地址通常极为相似。此时，数据包间隔较大的数据流占有的流表资源虽十分有限，但对应流表项的地位却十分关键。该攻击的一个主要特征是源 IP 地址欺骗，若关键流表项被欺骗式泛洪数据包影响导致频繁被更新或删除，大间隔数据包正常的传输会持续被间断，对应交换机向 SDN 控制器频繁发送 Packet-In 消息查询，最终导致控制器不堪重负，降低控制器对其他网络数据包的响应能力[20]。在此过程中，生成泛洪包的过程与普通数据流的显著差别在于单指令流包含较少的数据包个数，因此，通过对指令流包含数据包中位数特征的计算，如式(9)所示，能反映出控制层欺骗式泛洪的存在。

$$\mathrm{MPF}=\begin{cases}\mathrm{PktinFl}_{(n+1)/2}, n\text{为奇}\\ \dfrac{1}{2}(\mathrm{PktinFl}_{n/2}+\mathrm{PktinFl}_{(n+1)/2}), n\text{为偶}\end{cases} \tag{9}$$

在突发大流和欺骗式攻击 2 类网络状态下，SDN 数据层东西向均存在大规模的数据包和流量传输情况。然而，以接入层交换机端口为基准，2 类大规模数据包和流量传输状态存在明显的差别：伪造源地址的入流量无法得到交换机流表和控制器的有效响应，故对应的端口出入流量比例会产生严重的失衡；而流量密集型应用的大流量传输均为合法操作，接入层交换机端口流量并不会产生明显的失衡。在此 2 类状态下，仅仅通过检测数据流包含数据包中位数特征判定泛洪攻击在极端多数据包状态下误报率较高。因此为了实现可靠的检测，控制层欺骗式泛洪关键特征还包括成对流状态，如式(10)所示。正常通信的数据流所产生的成对通信指令通常标记着相同的通信协议，同时其源 IP 和目的 IP 也有着对应的源地址与目的地址。由于欺骗式泛洪攻击大量增加了外部流量，例如在 SYN 泛洪状态下，通过伪造源地址发起的外部大流量将产生大量无效 SYN-ACK 指令，这导致欺骗式泛洪发生时也存在大量包含失配流信息的 SYN-ACK 指令。以此特征为基础，通过对 SYN-ACK 指令包含流信息中成对流的数量的检测也能反映出欺骗式泛洪的存在，如式(11)所示。

$$\mathrm{PPF}=\frac{2\mathrm{Num}_{\mathrm{pair\_flows}}}{\mathrm{Num}_{\mathrm{flows}}} \tag{10}$$

$$\mathrm{GFF}=\frac{\mathrm{Num}_{\mathrm{flows}}-2\mathrm{Num}_{\mathrm{pair\_flows}}}{T_{\mathrm{init}}} \tag{11}$$

3.3　控制层攻击检测模型

为了检测与分类针对 SDN 控制层的泛洪攻击，本节以各控制层泛洪解析关键特征为基础，提出了基于 XGBoost（eXtreme gradient boosting）算法的控制层攻击检测模型。符号说明如表 1 所示。

表 1　XGBoost 增强型分类器符号说明

符号	说明
$\zeta(\Phi)$	由多维特征构成的控制层泛洪检测目标数据包
y_i	累加模型的输出
f_k	控制层攻击检测特征
$\sum_i l(\hat{y}_i, y_i)$	误差函数，使攻击检测模型越来越贴合实验训练数据
$\sum_k \Omega(f_k)$	分类检测树的复杂度函数，其值越小复杂度越低，泛化能力越强
γ, λ	正则化函数分裂参数
T	泛洪检测特征叶子节点的个数
w	节点的数值

XGBoost 是一种基于梯度 Boosting 的集成学习算法[21]，具有高准确度和可扩展性的特点。Boosting 算法将各个弱分类器加权叠加形成增强型分类器，通过不断降低计算中的残差，使之前的模型残差向梯度方向进一步降低，从而有效降低分类误差，达到更准确的分类的效果。XGBoost 对分类模型的目标函数的损失函数生成二阶泰勒展开，并在损失函数之外对正则项求导。其在优化分类的目标函数的同时，对用于分类的决策树模型进行了预剪枝，从而得到分类的最优参数，使分类结果更准确。基于 XGBoost 算法实现 Boosting 的步骤可表述如下。

1) 目标函数

$$\zeta(\Phi)=\sum_i l(\hat{y}_i, y_i)+\sum_k \Omega(f_k) \tag{12}$$

$$\Omega(f)=\gamma T+\frac{1}{2}\lambda\|w\|^2 \tag{13}$$

2) 训练目标函数

$$\zeta^{(t)}=\sum_{i=1}^{n} l(y_i,\hat{y}^{(t-1)}+f_t(x_i))+\Omega(f_t) \tag{14}$$

3) 目标函数二阶泰勒展开近似

$$\zeta^{(t)}\cong\sum_{i=1}^{n}\left[l(y_i,\hat{y}^{(t-1)})+g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i))\right]+\Omega(f_t) \tag{15}$$

4) 去掉常数项

$$\tilde{\zeta}^{(t)}=\sum_{i=1}^{n}\left[g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i))\right]+\Omega(f_t) \tag{16}$$

5) 求出目标函数最优解

$$\tilde{\zeta}^{(t)}(q)=-\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in g_i} g_i)^2}{\sum_{i\in l_j} h_i+\lambda}+\gamma T \tag{17}$$

其中，目标函数 $\zeta(\Phi)$ 代表由多维特征构成的控制层泛洪检测目标数据包，这个目标函数分为误差函数和正则化项 2 部分。时间窗口内特征集合的正则化定义复杂度，其值越小，复杂度越低，泛化能力越强。以此为基础训练输出泛洪检测目标累加函数，T 表示泛洪检测特征叶子节点的个数，w 表示节点的数值。接下来训练目标函数及求解最优解的过程则是 XGBoost 梯度下降分裂节点的标准训练流程。本文模型采用基于 SDN 的入侵检测数据集 InSDN[22]进行线下模型训练，而线上部分数据由轻量级流量采集模块实现。针对控制层泛洪攻击为 SDN 带来的安全问题，在时间与轮数不断递增的基础上进行交叉验证生成欺骗式泛洪检测模型，模型性能及参数测试结果将展示于 5.2 节。

4 基于 SAVI 的泛洪缓解机制

4.1 控制层泛洪源地址验证

若泛洪攻击仅通过重复伪装单一的源地址进行，该地址将立即被检测出并被过滤。由于 OpenFlow 协议提供了控制器和交换机之间的安全通信指令，其规定的标准化机构使各模块网络通信间的互操作性增强，为了达到欺骗通信指令攻击控制层的效果，现有控制层泛洪的发起往往运用许多不同源地址伪装，这将使 SDN 控制层欺骗式泛洪的防御更加困难[3,23]，此时最好的泛洪缓解方法则是尽可能过滤与源地址相近的数据包。

以 RFC7513[24]协议为例，SAVI 以绑定−验证模式为基础执行路径过滤，过滤的粒度则取决于 IP 前缀的粒度。绑定规则的构建是将源 IP 地址信息和 MAC 地址信息等一些难以冒用的属性进行关联，再进行统一的验证和过滤。典型的绑定关联表项包括主机端口、主机源地址及 MAC 地址，可记为 <SwitchPort, IP, MAC>[25]。由于 IP 地址只是逻辑上可被随意修改的一种标识，无法在源地址验证技术中被单独用于标识源地址对应用户；而 MAC 地址是物理上的一种标识，是一种固化在网卡内难以被冒用的属性。因此，SAVI 的源地址验证模式将源 IP 地址信息和 MAC 地址信息等一些难以冒用的属性绑定起来，统一数据结构进行验证。文献[26-27]在 SDN 中实现的 SAVI 通过 SDN 控制器事先获得底层主机的源 IP 地址、MAC 地址和交换机接口信息，将无状态的 IP 地址信息和底层的 MAC 地址信息、交换机 Port 信息绑定起来，形成三元组的过滤表项。该方案在控制器中生成和维护源地址关联绑定表，以源地址绑定表为依据执行伪造源地址的验证及路径过滤，保证数据层来自指定源地址的数据包只能通过已有的绑定关联信息执行操作。由于其 Port 作为交换机的端口，在基于伪造源地址的异常状态下，即使底层 IP 地址信息和 MAC 地址信息被伪造，攻击者也不可能同时拥有真实的交换机或路由连接端口。因此，以现有的 SAVI 绑定验证模式为基础，可根据绑定过滤表实现欺骗式泛洪攻击包过滤。

4.2 动态轮询及泛洪缓解

在执行 SDN 控制层泛洪攻击检测模块后，整合模型分类结果，可基于 SDN-SAVI（静态源地址验证）[28]与 D-SAVI（动态源地址验证）[23]实现差异化的安全管理。从网络安全管理的时间维度上来说，控制层泛洪缓解措施执行于正常网络流量和恶意网络行为的行为建模后，并根据网络流量分析、异常行为分析等结果进行动态分类处理。SDN 控制层差异化泛洪缓解模型如图 8 所示。在检测到 SDN 正常大流数据传输状态下，将持续进行交换机数据采集和 sFlow 流量数据采集。当主机初次请求接入 SDN 时，控制器会向其对应的交换机下发侦听网络地址分配报文状态的请求。针对不同的交换机地址分配机制（无状态自动配置机制和动态变化配置机制），控制器为了获取主机对应的交换机接入网络的目的和实时状态，下发不同的 AAM 报文对其进行侦听和请求查询泛洪检测模块持续运行。

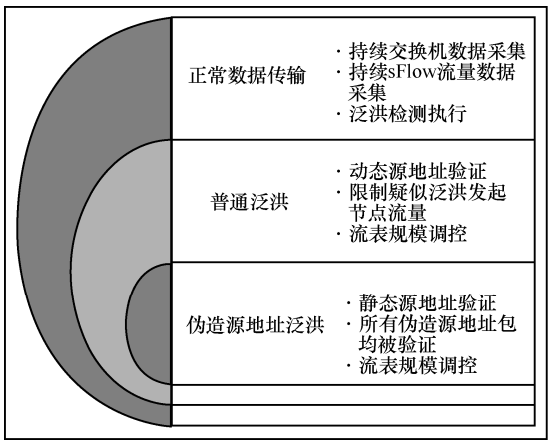

图 8　SDN 控制层差异化泛洪缓解模型

在普通泛洪发生时，执行动态源地址验证（D-SAVI）方案，交换机将接收的 AAM 报文进行预处理，并和部署在 OpenFlow 交换机中的侦听规则进行匹配。根据匹配结果，主机对应的交换机将 AAM 报文所包含的主机状态信息进行封装，以 Packet-In 包的形式与控制器进行通信。控制器成功接收 AAM 报文后，会从 Packet-In 包中解析出主机相关的地址信息和交换机状态信息，构建并更新绑定关系，限制疑似泛洪发起节点的流量，使其无法影响控制层信令发布及有效连接建立；在伪造源地址泛洪发生时，所有伪造源地址数据包均经过静态源地址验证（SDN-SAVI）方案筛选，通过将绑定表的构建和更新完全部署于控制器，利用 SDN 的全局视野维持 SAVI 的安全性标准，保证控制层通信的安全性。在现有的 SAVI 绑定机制下，为了维持 SAVI 绝对安全的标准，该绑定关系不会随着网络状态的变化而产生任何变化。根据三元组绑定信息的维护，控制器随后将持续执行源地址验证和路径过滤。

为了维持基于 SAVI 的泛洪缓解机制的安全性，控制层泛洪缓解模块主要以窗口轮询的模式进行。控制器会为所有 SDN 数据层主机均维护一个历史状态列表，以控制层泛洪特征构建的泛洪检测模型为基础，可推导每个主机随时间推移的时序状态列表。为了使 D-SAVI 进行安全保障的随机轮询更具针对性，本节提出了控制层泛洪安全保障的随机轮询算法，其流程如图 9 所示。该算法对所有正常主机遍历完毕后，根据主机时序状态列表的异常状态比例排序进行逆序轮询和检测，对其进行临时数据采集与分析，并部署源地址验证规则。因此，每次挑选的主机个数直接影响安全保障的随机轮询模块发现有控制层泛洪攻击安全隐患的主机的及时性，间接影响基于 SAVI 的泛洪缓解机制为控制器带来的负载：若每次挑选主机个数越少，则控制器和交换机资源消耗也越少，其安全性指标则会相应降低；反之则能更快发现控制层泛洪攻击异常主机，降低网络的安全威胁，但对 SDN 资源消耗也更多。因此，在 SDN 的不同拓扑结构中，选择合适数量实现安全保障的随机轮询模块是重要的平衡性指标。安全保障的随机轮询模块基于 3.2.2 节的流量差和信息熵实现。

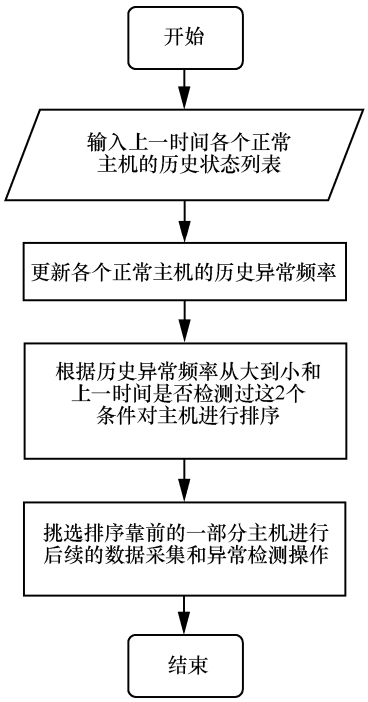

图 9　动态轮询及泛洪缓解流程

5　性能测试与分析

5.1　实验平台及关键参数

本节在多台服务器和基于 Vmware 安装的 Ubuntu 16.04.2 虚拟机系统上搭建了模拟的 SDN 环境。图 10 是经典的数据中心实验拓扑，本实验拓扑以 Fat-Tree 为基础搭建。安装 Floodlight 控制器作为 SDN 的控制平面，采用 Mininet 仿真对底层 SDN 拓扑进行差异化的仿真。SDN 拓扑中的 SDN 交换机由开源的虚拟交换机（OVS, OpenvSwitch）实现。在 Mininet 中实现软件定义网络的交换机可选择多种模式，其中最常见的就是 OVS 实现。该实验网络主要包含 14 台 OVS，其中，S_1 和 S_2 为核心层交换机，S_3～S_6 为汇聚层交换机，S_7～S_{14} 为接

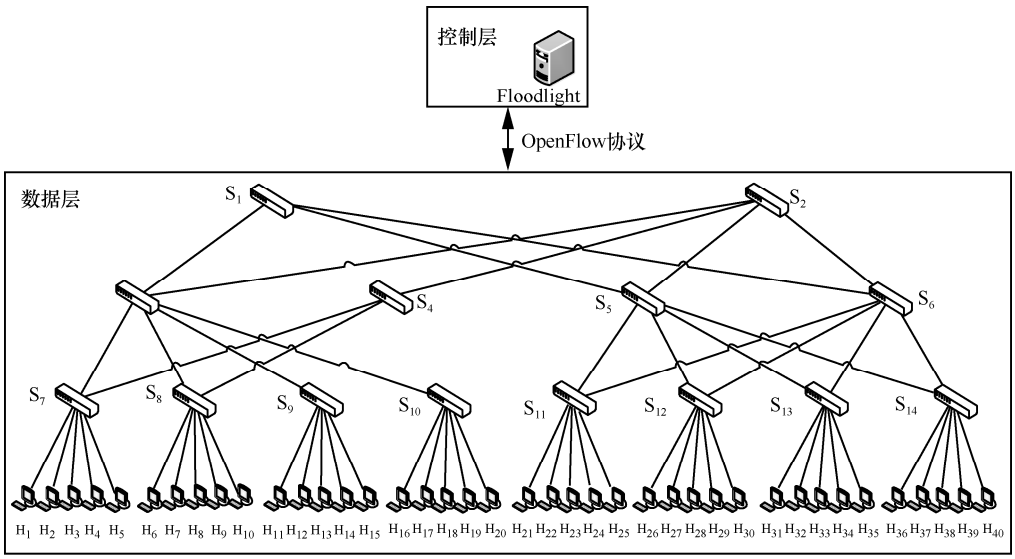

图 10　实验拓扑

入层交换机。实验网络共包含 40 台主机接入网络，本实验在模拟 SDN 数据中心网络拓扑采用分布式流量生成器（D-ITG, distributed Internet traffic generator）模拟发送正常背景流量和流量密集型应用“大象流”，同时通过脚本控制 Fat-Tree 拓扑的数据层主机随机组成多对进行相互通信，产生正常的数据中心东西向周期性网络通信流量。

SDN 控制层泛洪攻击流量由 Python 的 Scapy 工具模拟发出，通过模拟不同速率和比例的 Packet-In 泛洪和 SYN 泛洪攻击，致使 SDN 数据中心数据层目标主机的正常通信受到影响。控制器持续收集 30 min 流量数据。将上述攻击流量和正常 SDN 数据中心背景流量进行混合发包。在网络拓扑内分别启动静态与动态的源地址动态验证系统，在网络稳定后选定 H_1、H_6、H_{11}、H_{16}、H_{21}、H_{26}、H_{31} 和 H_{36} 为伪造源地址攻击主机，其中，H_1 和 H_2 连接于同一接入层交换机 S_1 的不同端口，而其余主机分别连接于其他交换机。

5.2　泛洪主机检测实验

本节对泛洪攻击检测算法的性能进行分析和比较，将本文的控制层攻击检测算法与其他基于特征的决策分类算法包括决策树（DT）[29]、随机森林（RF）[30]、k 近邻（KNN）[31]、朴素贝叶斯（NB）[32] 和支持向量机（SVM）[33]进行性能比较。由表 2 的结果得出，基于 XGBoost 的控制层攻击检测与分类算法具备在检测准确率和召回率方面最强的综合性能，其各个弱分类器加权叠加形成增强型分类器，通过不断降低计算中的残差，使之前的模型残差向梯度方向进一步降低，从而有效降低分类误差，达到更准确的分类效果。

表 2　　**异常检测算法比较**

算法	准确率	召回率	F1-Score
XGBOOST	96.03%	92.08%	94.01%
DT	95.28%	92.52%	93.88%
SVM	94.79%	90.91%	92.81%
KNN	93.17%	60.11%	73.08%
RF	68.53%	90.03%	77.82%
NB	71.25%	96.56%	81.99%

为了降低伪造源地址主机异常分类模型的在特征方面的复杂度，增强异常流警报时发现异常主机的效率，并减少对 SDN 系统资源的占用，对多维符合特征进行解析与评估是必要的流程。通过对每个特征目标函数最优解的遍历计算，选择 XGBoost 算法中损失函数的量化计算值作为分裂点，并用 GetScore( )函数计算增益损失。在遍历所有特征后，增益损失的最值通常可有效衡量一个特征是否具有最大的信息增益和卡方统计量，该统计量可直接反映出特征对分类模型的贡献程度。以控制层泛洪多维解析特征为基础，分别评估其检测泛洪攻击的准确率与召回率，结果如图 11 和图 12 所示。从图 11 和图 12 可以看出，在 Packet-In 泛洪攻击环境下，EPF、DTB 和 MPF 具备更好的分类性能，这是由于 Packet-In 泛洪攻击场景下突发性流量

大和流量速率倾斜。通过对 SDN 接入层交换机端口流量信息的分析，可初步鉴别 SDN 大流量的异常状态。而在 SYN 泛洪攻击的状态下，PPF、GFF 和 EPF 具备相对更好的分类性能。由于欺骗式泛洪攻击大量增加了外部流量，通过伪造源地址发起的外部大流量将产生大量无效 SYN-ACK 指令，即控制层欺骗式泛洪发生时存在大量包含失配流信息的 SYN-ACK 指令，这使成对性特征 PPF 和 GFF 具备更好的分类性能。

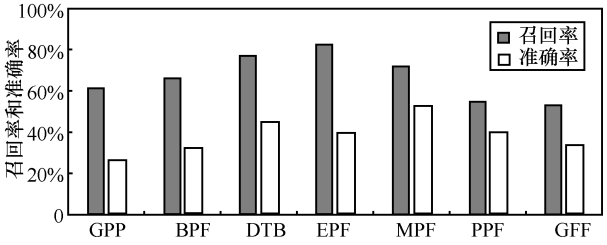

图 11　控制层 Packet-In 泛洪特征分类效率测试

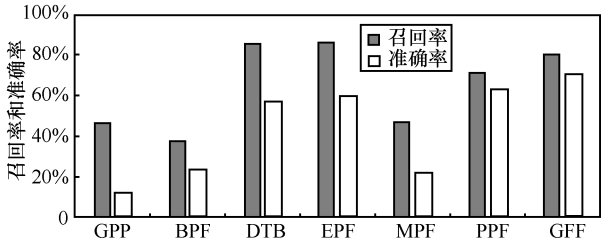

图 12　控制层 SYN 泛洪特征分类效率测试

准确率与召回率存在差异的原因在于准确率和召回率在控制层异常检测样本定义方面的差别：召回率是所有分类异常样本结果中包含的伪造样本数量占总伪造数的比例，而准确率则是对任意样本分类并定位伪造样本的准确比例。由于控制器泛洪场景涉及的表项特征通常都是高维的、稀疏的，并且样本量巨大，以 XGBoost 算法为基础的控制层攻击检测模型的本质是多个表项特征弱分类器的训练和组合，通过熵、信息增益、基尼指数等方法，各个特征弱分类器在每次分裂时选取最优的分裂节点，然后配置多维简单的弱分类器，可以迅速提高识别速度。该组合分类器能够广泛地检索识别疑似攻击样本，召回率相对可观，但各个弱分类器由于样本结果输出多，其实际分类准确率较低。

综合来看，EPF 在两类控制层泛洪场景下有着稳定的性能，这主要是由于 φ-熵可用于更加精确的量化主机当前端口多维度的出入流量差的绝对值与 Packet-In 泛洪攻击之间的关系。因此，本节对数据层端口信息 φ-熵用于建立数据层泛洪攻击分类模型的参数进行测试，结果如表 3 和表 4 所示。结果表明，基于熵的泛洪攻击流分类模型需要计算出合理的阈值以达到更好的分类效果。根据 SDN 拓扑的状态，本文实验以φ=0.3 为基础[34]进行阈值测试，并且基于 Type A 攻击类型进行了 50 次阈值测试，分别计算了基于泛洪攻击流量和普通背景流量状态下的熵值，如表 5 和表 6 所示。结果表明，最大攻击熵明显大于最小背景熵（max$A$ > min$N$），最大背景熵明显大于最小攻击熵（max$N$ > min$A$），这符合在控制层泛洪攻击场景下熵值变化的特点。

表 3　背景流量下异常流警报模型 φ-熵值

背景流量	平均熵	标准差	最小熵	最大熵
sIP	2.612	0.005 1	2.606 9	2.617 1
sPort	2.134	0.027 2	2.106 8	2.161 2
dIP	2.406	0.018 6	2.387 4	2.424 6
dPort	2.126	0.025 3	2.100 7	2.151 3

表 4　泛洪攻击下异常流警报模型 φ-熵值

泛洪攻击	平均熵	标准差	最小熵	最大熵
sIP	2.879	0.019 7	2.859 3	2.898 7
sPort	2.539	0.028 7	2.510 3	2.567 7
dIP	2.205	0.012 7	2.192 3	2.217 7
dPort	1.904	0.009 2	1.894 8	1.913 2

表 5　φ-熵值异常流警报模型最小熵阈值测定

特征	背景流量的最小熵（min$N$）	攻击流量 A 的最大熵（max$A$）	$\frac{\min N+\max A}{2}$
sIP	2.606 9	2.898 7	2.752 8
sPort	2.106 8	2.567 7	2.3372 5

表 6　φ-熵值异常流警报模型最大熵阈值测定

特征	背景流量的最大熵（max$N$）	攻击流量 A 的最小熵（min$A$）	$\frac{\max N+\min A}{2}$
dIP	2.424 6	2.192 3	2.308 45
dPort	2.151 3	1.894 8	2.023 05

5.3　泛洪攻击缓解实验

本节 SDN 控制层泛洪缓解模块基于 SAVI 实现，将对该模块在泛洪缓解及安全性方面的性能进行测试与分析。图 13 是部署了基于 SAVI 的泛洪攻击缓解模块的 SDN 中泛洪攻击数据包的实时数目测试值。在差异化的状态分组验证体系之下，基于 SAVI 的泛洪攻击缓解模块能显著降低欺骗式泛洪

攻击数据包的数目，差异化的动态轮询模式相比于 D-SAVI 具备更快的响应速度。这符合控制层泛洪攻击检测与缓解在时延方面的需求，符合 SAVI 源地址绑定−验证体系的基本的安全性能标准[26]。

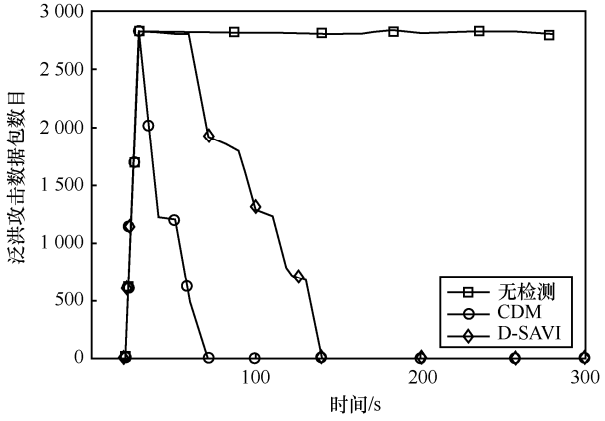

图 13 泛洪攻击数据包数目测试

响应时间是控制器对一个 Packet-In 请求作出反馈的时延，图 14 展示了在泛洪攻击状态下控制器响应时间测试结果，结果表明当 SDN 中激活泛洪攻击主机的数目增多时，控制器的响应时间也将相应增加。在无泛洪攻击检测与防御部署的状态下，控制器的性能将受到严重的影响，数据层的基础通信和与控制层的交互指令的完整性受到极大的破坏。通过控制层泛洪攻击检测与防御机制 CDM 的部署，控制器的响应效率能恢复近似于无攻击状态下的性能。这表明 CDM 能有效检测与缓解控制层欺骗式泛洪，维护 SDN 控制层的安全运行。

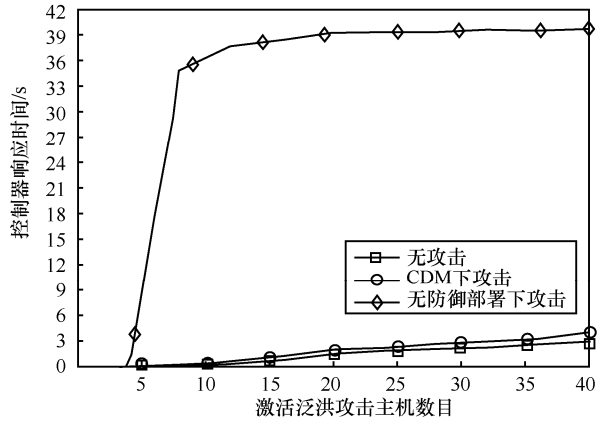

图 14 控制器响应时间测试

6 结束语

本文提出了一种轻量级、高精度的控制层泛洪检测与缓解机制，该方法将泛洪防御问题分为检测和缓解 2 个步骤，分别解决了泛洪攻击主机分类、泛洪攻击数据包路径过滤和负载优化问题。在泛洪检测方面，提出了轻量级的控制层泛洪关键特征解析模块，利用 Boosting 算法将各个特征弱分类器加权叠加形成增强型分类器，通过不断降低计算中的残差，达到了高精度的欺骗式泛洪攻击检测效果。在泛洪缓解方面，CDM 部署了基于 SAVI 的泛洪数据包过滤机制，以绑定−验证的模式为基础执行控制层泛洪数据包的路径过滤，同时以动态轮询的模式实现安全保障和泛洪关键特征实时更新。所提方法要求泛洪特征更新的实时性，暂未考虑 SDN 链路动态变化的情况，若面向更复杂的拓扑变化状态实施差异化的泛洪主机源地址验证模式，其动态轮询安全保障模块负载将会较大，如何在 CDM 中解决这个问题将是下一步工作的重点。

参考文献：

[1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.

[2] 黄韬, 刘江, 魏亮, 等. 软件定义网络核心原理与应用实践[J]. 通信学报, 2015, 36(3): 288.
HUANG T, LIU J, WEI L, et al. SDN core principles and application practice[J]. Journal on Communications, 2015, 36(3): 288.

[3] KUMAR P, TRIPATHI M, NEHRA A, et al. SAFETY: early detection and mitigation of TCP SYN flood utilizing entropy in SDN[J]. IEEE Transactions on Network and Service Management, 2018, 15(4): 1545-1559.

[4] GAO D Y, LIU Z H, LIU Y, et al. Defending against Packet-In messages flooding attack under SDN context[J]. Soft Computing, 2018, 22(20): 6797-6809.

[5] RAVI N, SHALINIE S M, LAL C, et al. AEGIS: detection and mitigation of TCP SYN flood on SDN controller[J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 745-759.

[6] DANG V T, HUONG T T, THANH N H, et al. SDN-based SYN proxy—a solution to enhance performance of attack mitigation under TCP SYN flood[J]. The Computer Journal, 2019, 62(4): 518-534.

[7] AL MHDAWI A K, AL-RAWESHIDY H S. iPRDR: intelligent power reduction decision routing protocol for big traffic flood in hybrid-SDN architecture[J]. IEEE Access, 2018, 6: 10944-10955.

[8] MOHAMMADI R, CONTI M, LAL C, et al. SYN-Guard: an effective counter for SYN flooding attack in software-defined networking[J]. International Journal of Communication Systems, 2019, 32(17): e4061.

[9] DERHAB A, GUERROUMI M, GUMAEI A, et al. Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security[J]. Sensors (Basel, Switzerland), 2019, 19(14): 3119.

[10] XIANG S Q, ZHU H B, XIAO L L, et al. Modeling and verifying TopoGuard in OpenFlow-based software defined networks[C]// Proceedings of 2018 International Symposium on Theoretical Aspects of Software Engineering (TASE). Piscataway: IEEE Press, 2018: 84-91.

[11] KAZEMANIAN P, CHANG M, ZENG H Y, et al. Real time network

policy checking using header space analysis[C]//Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13). Berkeley: USENIX Association, 2013: 99-111.

[12] TUAN N N, HUNG P H, NGHIA N D, et al. A robust TCP-SYN flood mitigation scheme using machine learning based on SDN[C]// Proceedings of 2019 International Conference on Information and Communication Technology Convergence (ICTC). Piscataway: IEEE Press, 2019: 363-368.

[13] SEMERCI M, CEMGIL A T, SANKUR B. An intelligent cyber security system against DDoS attacks in SIP networks[J]. Computer Networks, 2018, 136: 137-154.

[14] GARG S, KAUR K, KUMAR N, et al. Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: a social multimedia perspective[J]. IEEE Transactions on Multimedia, 2019, 21(3): 566-578.

[15] PHAAL P, PANCHEN S, MCKEE N. InMon corporation's flow: a method for monitoring traffic in switched and routed networks[R]. 2001.

[16] CICIOĞLU M, ÇALHAN A. HUBsFLOW: a novel interface protocol for SDN-enabled WBANs[J]. Computer Networks, 2019, 160: 105-117.

[17] PANDA A, SAMAL S S, TURUK A K, et al. Dynamic hard timeout based flow table management in openflow enabled SDN[C]// Proceedings of 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). Piscataway: IEEE Press, 2019: 1-6.

[18] SHIRALI-SHAHREZA S, GANJALI Y. Delayed installation and expedited eviction: an alternative approach to reduce flow table occupancy in SDN switches[J]. IEEE/ACM Transactions on Networking, 2018, 26(4): 1547-1561.

[19] BASTA A, BLENK A, HOFFMANN K, et al. Towards a cost optimal design for a 5G mobile core network based on SDN and NFV[J]. IEEE Transactions on Network and Service Management, 2017, 14(4): 1061-1075.

[20] SCHNEPF N, BADONNEL R, LAHMADI A, et al. Synaptic: a formal checker for SDN-based security policies[C]//Proceedings of NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. Piscataway: IEEE Press, 2018: 1-2.

[21] CHEN T, TONG H, BENESTY M. Xgboost: extreme gradient boosting [C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. New York: ACM Press, 2016: 1615-1624.

[22] ELSAYED M S, LE-KHAC N A, JURCUT A D. InSDN: a novel SDN intrusion dataset[J]. IEEE Access, 2020, 8: 165263-165284.

[23] ZHOU Q Z, YU J Q, LI D. A dynamic and lightweight framework to secure source addresses in the SDN-based networks[J]. Computer Networks, 2021, 193: 108075.

[24] BI J, WU J, YAO G, et al. Source address validation improvement (SAVI) solution for DHCP[R]. RFC Editor, 2015.

[25] WU J, BI J, BAGNULO M, et al. Source address validation improvement (SAVI) framework[R]. RFC Editor, 2013.

[26] LIU B Y, BI J, ZHOU Y. Source address validation in software defined networks[C]//Proceedings of Proceedings of the 2016 ACM SIGCOMM Conference. New York: ACM Press, 2016: 595-596.

[27] CHEN G L, HU G W, JIANG Y, et al. SAVSH: IP source address validation for SDN hybrid networks[C]//Proceedings of 2016 IEEE Symposium on Computers and Communication (ISCC). Piscataway: IEEE Press, 2016: 409-414.

[28] LI C L, WU Q, LI H W, et al. SDN-Ti: a general solution based on SDN to attacker traceback and identification in IPv6 networks[C]// Proceedings of ICC 2019 - 2019 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2019: 1-7.

[29] WU Y C, TSENG H R, YANG W, et al. DDoS detection and traceback with decision tree and grey relational analysis[C]//Proceedings of 2009 3rd International Conference on Multimedia and Ubiquitous Engineering. Piscataway: IEEE Press, 2009: 306-314.

[30] BELGIU M, DRĂGUȚ L. Random forest in remote sensing: a review of applications and future directions[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2016, 114: 24-31.

[31] ZHANG S C, LI X L, ZONG M, et al. Efficient kNN classification with different numbers of nearest neighbors[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(5): 1774-1785.

[32] CHU S C, DAO T K, PAN J S, et al. Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification[J]. EURASIP Journal on Wireless Communications and Networking, 2020, 2020(1): 52.

[33] WANG H W, GU J, WANG S S. An effective intrusion detection framework based on SVM with feature augmentation[J]. Knowledge-Based Systems, 2017, 136: 130-139.

[34] WANG J X, QI H, HE Y, et al. FlowTracer: an effective flow trajectory detection solution based on probabilistic packet tagging in SDN-enabled networks[J]. IEEE Transactions on Network and Service Management, 2019, 16(4): 1884-1898.

[作者简介]

周启钊（1991− ），男，湖南长沙人，华中科技大学博士生，主要研究方向为机器学习、软件定义网络、网络安全等。

于俊清（1975− ），男，内蒙古赤峰人，博士，华中科技大学教授、博士生导师，主要研究方向为数字媒体处理与检索、网络安全、多核计算与流编译等。

李冬（1979− ），男，湖北武汉人，博士，华中科技大学讲师，主要研究方向为网络安全、入侵检测、僵尸网络检测、网络流数据挖掘与分析、无线网络跨层优化等。