云环境中基于 SDN 的高效 DDoS 攻击检测与防御方案

何亨[1,2]，胡艳[1,2]，郑良汉[1,2]，薛正元[3]

（1. 武汉科技大学计算机科学与技术学院，湖北 武汉 430065；
2. 武汉科技大学智能信息处理与实时工业系统湖北省重点实验室，湖北 武汉 430065；
3. 华中科技大学计算机科学与技术学院，湖北 武汉 430074）

摘 要：针对云环境中 2 类典型的分布式拒绝服务（DDoS）攻击问题，提出一种基于软件定义网络架构的 DDoS 攻击检测与防御方案——SDCC。SDCC 综合使用链路带宽和数据流这 2 种检测方式，利用基于置信度过滤（CBF）的方法计算数据分组 CBF 分数，将分数低于阈值的数据分组判断为攻击分组，添加其属性信息至攻击流特征库，并通过控制器下发流表将其拦截。仿真实验表明，SDCC 能有效检测并防御不同类型 DDoS 攻击，具有较高检测效率，降低了控制器计算开销，并保持较低误判率。

关键词：云环境；DDoS 攻击；软件定义网络；基于置信度过滤

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000-436x.2018068

Efficient DDoS attack detection and prevention scheme based on SDN in cloud environment

HE Heng[1,2], HU Yan[1,2], ZHENG Lianghan[1,2], XUE Zhengyuan[3]

1. School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China
2. Hubei Province Key Laboratory of Intelligent Information Processing and Real Time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China
3. School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract: For addressing the problem of two typical types of distributed denial of service (DDoS) attacks in cloud environment, a DDoS attack detection and prevention scheme called SDCC based on software defined network (SDN) architecture was proposed. SDCC used a combination of bandwidth detection and data flow detection, utilized confidence-based filtering (CBF) method to calculate the CBF score of packets, judged the packet of CBF score below the threshold as an attacking packet, added its attribute information to the attack flow feature library, and sent the flow table to intercept it through SDN controller. Simulation results show that SDCC can detect and prevent different types of DDoS attacks effectively, and it has high detection efficiency, reduces the controller's computation overhead, and achieves a low false positive rate.

Key words: cloud environment, DDoS attack, software defined network, confidence-based filtering

收稿日期：2017-09-05；**修回日期**：2018-02-17

通信作者：何亨，heheng@wust.edu.cn

基金项目：国家自然科学基金资助项目（No.61602351, No.61502359, No.61602349）；智能信息处理与实时工业系统湖北省重点实验室开放基金资助项目（No.2016znss10B）

Foundation Items: The National Natural Science Foundation of China (No.61602351, No.61502359, No.61602349), The Open Foundation of Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System (No. 2016znss10B)

1 引言

由于云计算具有虚拟化、高可扩展性、按需服务等特点，越来越多的企业和个人将应用迁移到云计算环境中运行[1]。但是，云计算的规模大、易获取等特点，使云环境中的分布式拒绝服务（DDoS, distributed denial of service）攻击比传统网络中的攻击力度和范围都要大得多。由于传统网络中 DDoS 攻击检测算法[2~5]的检测结果受攻击强度的影响，难以应对云环境中强度更大的攻击。文献[6]提出的 PacketScore 方案利用贝叶斯公式计算数据分组的分数，分数比阈值低的是攻击分组，但由于阈值受攻击强度的影响，因此不适合在云环境中处理大流量的 DDoS 攻击。对此，研究人员提出一系列针对云环境中 DDoS 攻击的检测与防御方法。胡汉卿[7]提出基于特征检测和行为检测相结合的方法来检测 DDoS 攻击。Dou 等[8]提出基于置信度过滤（CBF, confidence-based filtering）的方法，根据数据分组中的相关特性来判断分组的合法性。文献[9]利用数据挖掘和神经网络技术训练数据分组来检测和过滤攻击分组。现有的云环境中检测和防御 DDoS 攻击的方法主要是针对大流量的洪泛式 DDoS 攻击，文献[10]通过将数据分组分类的方式检测洪泛式 DDoS 攻击，并建立黑名单存储攻击分组源 IP 进行防御。文献[11]提出了一种欺骗检测算法用于检测在云环境中对服务器发起的大流量 DDoS 攻击。但近年来云环境中逐渐流行的低速率式 DDoS 攻击[12]能很好地躲避这些针对洪泛式 DDoS 攻击的检测方法。文献[13]将基于熵的系统和异常检测系统相结合，以提供多级检测方法来检测云环境中隐蔽的小流量 DDoS 攻击，但是对云环境中大流量的 DDoS 攻击，这种方法速度较慢。尽管研究人员已经分别提出了一些检测云环境中洪泛式 DDoS 攻击和低速率式 DDoS 攻击的方法，却很少考虑到实际系统可能会遭受到不同攻击方式的情况。因此，研究在云环境中既能检测传统洪泛式 DDoS 攻击又能检测低速率式 DDoS 攻击的方法，给云用户提供安全的网络环境具有重要意义。

软件定义网络（SDN, software defined network）是一种新型的网络架构，其控制转发分离的特点简化了数据层面数据分组转发的过程；中心控制器连接了所有交换机，能实时获取网络拓扑结构及状态信息；因此，使用 SDN 架构有利于在云环境中快速检测 DDoS 攻击并做出迅速反应。研究人员利用 SDN 的特性提出了一些云环境中 DDoS 攻击的检测与防御方法[1]，结果表明，使用 SDN 架构的方法比其他未使用的检测和防御速度更快。Wang 等[14]提出基于 SDN 架构的 DaMask 系统，通过不断更新数据分组类型库来检测攻击分组，然后通过 SDN 控制器下发流表拦截攻击分组，但在新的攻击分组类型较多时，DaMask 系统需要不断更新数据分组类型库，会影响检测速度。文献[15]在 SDN 架构上，结合正常流量学习、外部黑名单和弹性容量调用来实现自动化地防御洪泛式 DDoS 攻击，但却难以检测到小流量的低速率式 DDoS 攻击。

因此，本文提出一种云环境中基于 SDN 架构，并使用改进的 CBF 方法来检测与防御 DDoS 攻击的方案——SDCC。与文献[10]解决洪泛式 DDoS 攻击和文献[12,13]解决低速率式 DDoS 攻击这些只能检测单一类型 DDoS 攻击的方案相比，SDCC 综合使用链路带宽检测和数据流检测这 2 种方式，能准确检测出不同类型的 DDoS 攻击，其中，基于链路带宽的检测方式通过 SDN 架构实时获取交换机各端口的流量信息，主要用来检测洪泛式 DDoS 攻击；基于数据流的检测方式根据系统的不同状态抽取不同比例数据分组，分析其中非正常分组比例的方式来实时检测系统中的数据流状态，主要用于检测低速率式 DDoS 攻击。在文献[8]提出的 CBF 方案中，SDCC 不仅简化了 profile 表的更新，实现了实时检测系统状态，还通过增加攻击流特征库，大幅降低了控制器的检测压力。在系统遭受 DDoS 攻击时，若不是攻击流特征库中的攻击分组类型，则使用改进的 CBF 方法计算数据分组的 CBF 分数来判断攻击分组，并将已识别的攻击分组存入攻击流特征库中，再利用 SDN 控制器下发流表拦截攻击分组以保护系统免受攻击。无论攻击源来自云外还是云内，SDCC 都能在误判率较低的情况下及时检测出云环境中不同类型的 DDoS 攻击，且在提高了 DDoS 攻击检测效率的同时有效降低了控制器的计算开销。

2 相关工作

本节首先介绍了云环境中 DDoS 攻击与传统网络环境中 DDoS 攻击的差异以及逐渐流行的低速率式 DDoS 攻击；然后介绍了 SDN 技术的发展和特点；最后介绍了 CBF 方法检测 DDoS 攻击的流程。

2.1　云环境中 DDoS 攻击

DDoS 攻击是通过控制分布式的多台服务器和 PC 机组成联合攻击平台，向一个或多个目标发送大量恶意流量，使网络带宽或系统资源被消耗殆尽，造成正常请求遭到拒绝。在传统网络中，短时间内感染足够数量的机器仍然相当复杂，但在按需服务的云环境中，却能快速创建一个强大的僵尸网络[1]，因此云环境中 DDoS 攻击的强度也会比传统网络中大得多。对于云环境中更大流量的 DDoS 攻击，传统的检测方法难以做出快速反应。所以，在云环境中检测这类大流量的洪泛式 DDoS 攻击的关键就是检测算法能快速检测出攻击分组，尽量减少攻击分组对系统资源的过多消耗。

但近年来黑客们为了躲避这些已有的洪泛式 DDoS 攻击检测防御方法，逐渐转为发起小流量的低速率式 DDoS[16,17]攻击。低速率式 DDoS 攻击利用 TCP 协议中超时重传机制的弱点，使攻击者通过控制多台傀儡机向攻击目标周期性发送小流量的攻击流，这些小流量的攻击流能在攻击目标处汇聚成巨大攻击流，导致攻击目标无法及时响应用户的正常请求，这种攻击方式虽然攻击流量速率低，却有很高的攻击效率[18]，且能躲避传统针对洪泛式 DDoS 攻击的检测方法。如文献[19]中提到的以 TCP 协议为目标，通过 RTO 计时器，在链接中制造运行中断，从而导致 TCP 控制机构拥塞的低速率式 DDoS 攻击。这种低速率式 DDoS 攻击相对于传统洪泛式 DDoS 攻击而言，是一个小流量逐渐汇聚的攻击过程，由于使用的攻击流量较少，其成本更低，同时难以被传统 DDoS 攻击检测方法检测到。

2.2　SDN 技术

SDN 是一种目前非常受关注的新型网络架构，它的出现主要是由于传统互联网控制逻辑和数据转发紧耦合在网络设备上，使网络设备的复杂性变高，且灵活性和扩展性降低，难以适应网络的飞速发展[20]。2008 年，由美国斯坦福大学的 Nick McKeown 教授最早提出基于逻辑控制和数据转发分离思想的 OpenFlow 技术概念，采用 OpenFlow 技术作为通信协议的 SDN 概念，随之被推广。

SDN 架构利用 OpenFlow 技术将控制功能从网络设备中分离出来。数据分组依照流表进行转发，中央控制器管理流表的生成、维护和配置。在这种控制转发分离架构下，中央控制器不仅能动态管理网络的逻辑控制功能和高层策略，还能获取全局的网络资源信息，并根据需求进行全局调配和优化。同时，这种转发控制分离的设计使上层的网络服务和应用功能程序能将底层的网络基础设施抽象化，从而能通过开放可编程的软件模式实现网络自动化控制的功能，使 SDN 具有可编程能力，便于实现网络创新和进化。

2.3　CBF 方法

Dou 等[8]提出的方案使用 CBF 方法来检测云环境中 DDoS 攻击分组，该方案能很好地解决云环境中大流量的 DDoS 攻击。由于每个用户的喜好不同，网络数据流中数据分组的 IP 头部和 TCP 头部中一些属性对值的分布也不同，DDoS 攻击者不仅很难注意这些相关特性，也难以在满足 DDoS 攻击条件的同时在数据分组中模拟这些特性。系统在正常情况下，受用户喜好的影响，流入系统中数据分组的一些属性对值出现的频率高；因此，若一个数据分组中的属性对值等于这些高频属性对值，则该数据分组的置信度高，那么通过置信度计算出来的数据分组的 CBF 分数也会高。反之，由于攻击者难以模拟出系统正常数据流中的属性对值的分布，因此攻击分组中很难出现高频属性对值，而导致攻击分组的置信度低，那么攻击分组的 CBF 分数也会比正常数据分组低很多。CBF 方案正是基于置信度，通过计算数据分组的 CBF 分数判断分组的合法性，来过滤攻击分组。

CBF 方案选取了数据分组 IP 头部的 *Total Length*、*TTL*、*Protocol Type*、*Source IP Address* 以及 TCP 头部的 *Flag*、*Destination Port Number* 这 6 个属性，两两共组成了 15 个属性对，其实现过程分为非攻击阶段和攻击阶段。在非攻击阶段，需要计算数据流中正常数据分组的单属性值和属性对值的出现频率，然后更新到 profile 表中。

Conf 表示属性值的出现频率，单属性值的 *Conf* 计算式为

$$Conf(A_i = a_{i,j}) = \frac{N(A_i = a_{i,j})}{N_n} \tag{1}$$

其中，A_i 表示数据分组中的第 i 个属性，i = 1, 2, 3, …, n，$j = 1, 2, 3, \cdots, m_i$，n 表示属性数，m_i 表示属性 A_i 拥有的值的个数，N_n 表示数据流中数据分组总数。$N(A_i = a_{i,j})$ 表示数据流中属性 A_i 的值为 $a_{i,j}$ 的分组个数。

属性对值的 *Conf* 计算式为

$$Conf(A_{i_1}=a_{i_1,j_1},A_{i_2}=a_{i_2,j_2})=\frac{N(A_{i_1}=a_{i_1,j_1},A_{i_2}=a_{i_2,j_2})}{N_n} \quad (2)$$

其中，$i_1=1,2,3,\cdots,n$，$i_2=1,2,3,\cdots,n$，$j_1=1,2,3,\cdots,m_1$，$j_2=1,2,3,\cdots,m_2$，$(A_{i_1}=a_{i_1,j_1},A_{i_2}=a_{i_2,j_2})$ 表示属性 A_{i_1} 的值为 a_{i_1,j_1}，属性 A_{i_2} 的值为 a_{i_2,j_2} 时组成的属性对值，$N(A_{i_1}=a_{i_1,j_1},A_{i_2}=a_{i_2,j_2})$ 表示数据流中属性对值为 $(A_{i_1}=a_{i_1,j_1},A_{i_2}=a_{i_2,j_2})$ 的分组个数。

profile 表是用来存储系统正常数据流的 15 个属性对不同的属性对值和对应 *Conf* 值的信息表。为了节省存储空间，profile 表生成过程中会利用最小 *Conf*（*minconf*）来筛选正常数据流中出现概率很低的属性对值，其中，*minconf* 取值根据 profile 表中属性对值的 *Conf* 分布来设定。因此，需要扫描 2 遍数据流：第一遍选出数据流中单属性值的 *Conf* 比 *minconf* 大的属性值，以此产生候选属性对值；第二遍选出候选属性对值中 *Conf* 比 *minconf* 大的属性对值，然后更新 profile 表。

在攻击阶段，对经过交换机的数据分组，首先查找 profile 表得到数据分组中属性对值的 *Conf*，若表中没有该属性对值信息，则用 *minconf* 代替计算。然后利用加权平均的方式计算数据分组的 CBF 分数。CBF 分数计算式为

$$Score(p)=\frac{\sum_{k=1}^{d}W(A_{k_1},A_{k_2})Conf(A_{k_1}=p(k_1),A_{k_2}=p(k_2))}{\sum_{k=1}^{d}W(A_{k_1},A_{k_2})} \quad (3)$$

其中，d 表示需要考虑的属性对的总数，$W(A_{k_1},A_{k_2})$ 表示属性对 (A_{k_1},A_{k_2}) 的权值，$p(k_1)$ 表示数据分组中属性 A_{k_1} 的值，$Conf(A_{k_1}=p(k_1),A_{k_2}=p(k_2))$ 表示属性对值 $(A_{k_1}=p(k_1),A_{k_2}=p(k_2))$ 的 *Conf* 值。数据分组的 CBF 分数计算过程如图 1 所示。

在计算数据分组 CBF 分数的过程中，由于数据分组之间的 *Source IP Address*、*Destination Port Number*、*TTL* 这些属性值的差异性较大，攻击者在攻击分组中很难模拟正常数据分组中这些属性值的特征，而 *TCP Flag*、*Protocol Type*、*Total Length* 这些属性值在数据分组之间的差异性较小，因此，若属性对由前 3 种属性中的两者组合而成，则权值最高；若属性对中只包含前 3 种属性中的一种，则权值其次；若属性对由后 3 种属性中的两者组合而成，则权值最低。在此基础上，属性对具体权值可以根据实际网络情况来设置。

最后将得到的数据分组 CBF 分数与阈值比较，若比阈值小，则判断为攻击分组并将其丢弃；若比阈值大，则判断为正常分组并使其通过。

3 方案设计与实现

本节首先介绍了 SDCC 的工作流程，然后详细描述了 SDCC 中 SDCC-D 和 SDCC-P 这 2 个主要模块的具体实现和关键技术。

3.1 工作流程

SDCC 具体分为 3 层：基础设施层、控制层和应用层。基础设施层包含多台 OpenFlow 交换机，

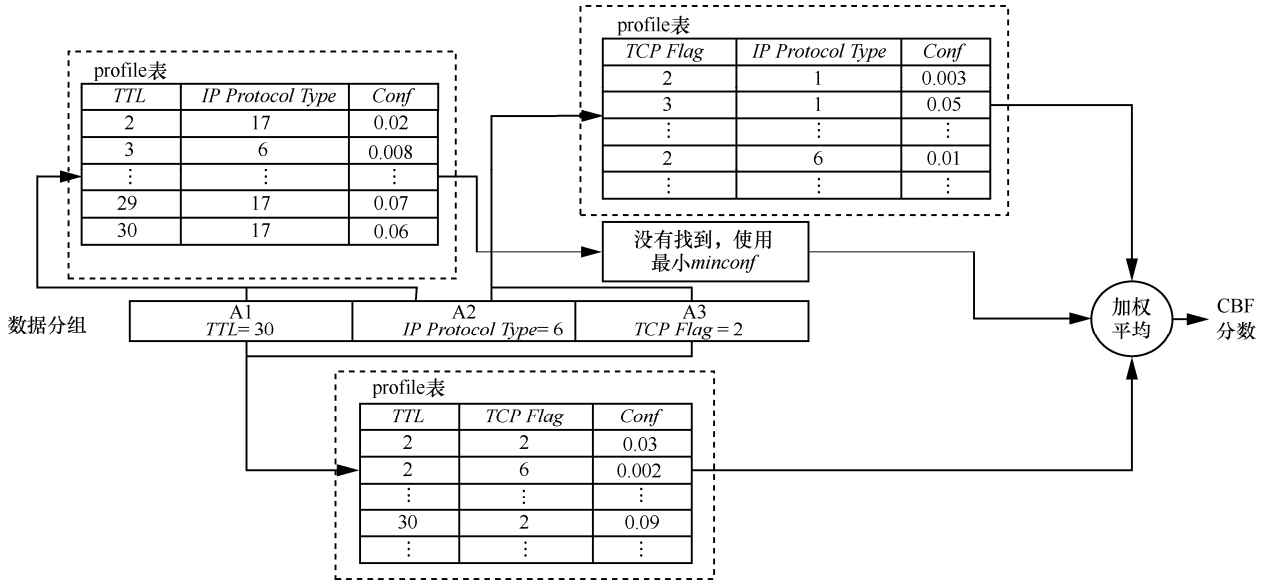


图 1 数据分组的 CBF 分数计算过程

负责系统的数据转发；控制层为 SDN 控制器，能对系统进行全局逻辑管控；应用层由 DDoS 攻击检测（SDCC-D）和 DDoS 攻击防御（SDCC-P）这 2 个核心功能模块构成。SDCC 工作流程如图 2 所示。

当数据流到达 OpenFlow 交换机时，首先会与交换机中的流表进行匹配，若交换机中没有流表与之匹配，则根据数据分组在 SDN 中的处理原理[21]，数据流中的数据分组会被转发至控制器中；SDCC-D 模块中的 2 种检测方式会分别实时检测系统的链路带宽和分析数据流中非正常分组的比例，来判断系统是否遭受到 DDoS 攻击。若未检测出系统中有 DDoS 攻击行为，系统进入正常状态，将这段正常数据流中数据分组的属性对信息存入属性对信息库，并更新 profile 表；若检测出系统中有疑似 DDoS 攻击行为，系统进入预警状态，并加大数据流的检测比例；若检测出系统中确有 DDoS 攻击行为，系统进入防御状态，此时 SDCC-P 模块会计算该攻击流中的数据分组的 CBF 分数，判断出攻击分组，查攻击流特征库中是否有相匹配的攻击分组，有就直接将该攻击分组转由交换机通过流表丢弃，没有就添加到攻击流特征库并通知控制器下发流表至 OpenFlow 交换机将其丢弃。

无论攻击源来自云外还是云内，SDCC 都能及时检测出云环境中 2 类典型的 DDoS 攻击。在攻击云中服务器时，云外攻击者通过云对外服务接入点注入攻击流量，当流量经过 OpenFlow 交换机时，OpenFlow 交换机向 Floodlight 控制器请求流表转发。Floodlight 控制器能够实时监控云中数据流情况，并分析判定是否是攻击流量，从而下发流表进行相应防御动作。而云内攻击者利用云内资源通过云内部网络对云中服务器发动攻击，使用云资源的 DDoS 攻击能够大幅加强其攻击强度，也能够避开云在接入端的流控措施。然而攻击流量经过内网的 OpenFlow 交换机时，已被控制器所知，控制器能够快速下发流表阻断来自内部攻击源的流，从而防御来自内部的 DDoS 攻击。

3.2　攻击检测模块

SDCC-D 模块功能为实时检测系统中是否有 DDoS 攻击行为，检测方式结合了链路带宽检测和数据流检测这 2 种，基于链路带宽检测通过监测交换机端口流量异常，能快速检测出洪泛式 DDoS 攻击行为，但是却难以检测出低速率式 DDoS 攻击行为，也难以区分用户自身增大访问量而引起的流量异常。基于数据流检测方式虽然能通过监测数据流中非正常分组的比例来准确地检测出 DDoS 攻击，但是它需要对每个被抽取的数据分组进行分析，速度较慢。因此，将 2 种检测方式相结合，既能准确、及时地检测出不同的 DDoS 攻击，又具有较高的检测效率，同时还能降低由用户自身增大访问量和用户非习惯性访问引起的误判率。

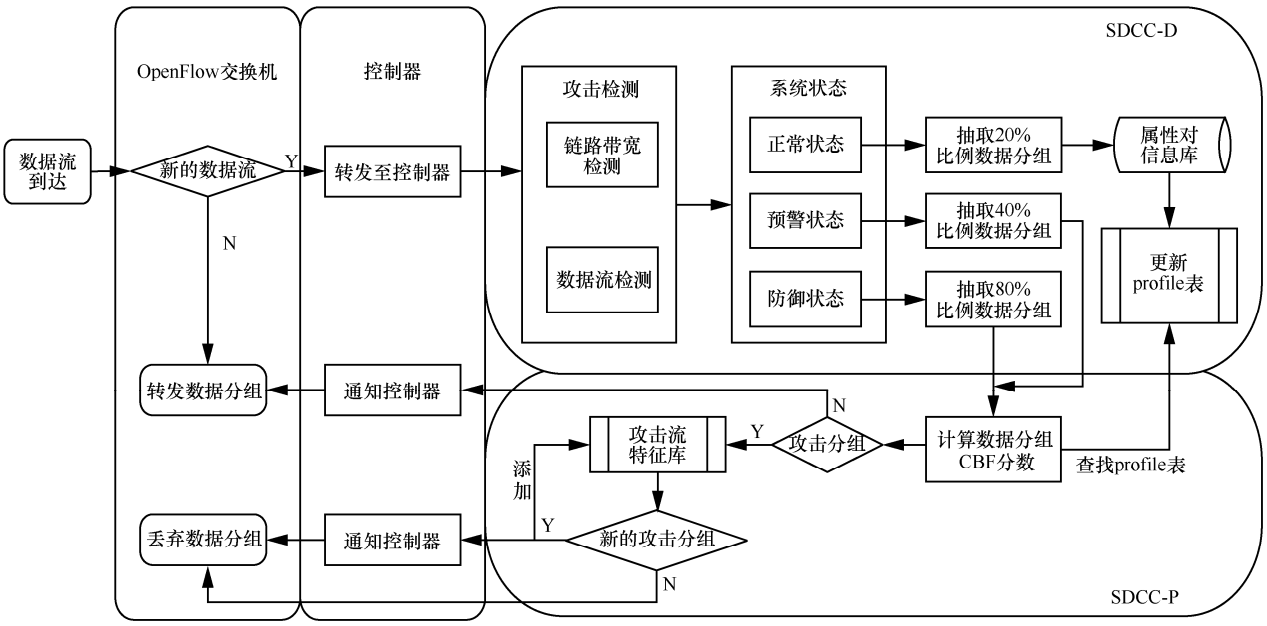


图 2　SDCC 的工作流程

当 2 种检测方式都未检测到 DDoS 攻击行为时，系统进入正常状态，为节省计算资源，系统处于正常状态时，抽取窗口内 20%比例的数据分组进行数据流检测，并提取 6 个参考属性，组成属性对存入属性对信息库中，更新 profile 表。若 2 种检测方式的其中一种方式检测到可能有 DDoS 攻击行为，系统进入预警状态，系统处于预警状态时，可能遭受到 DDoS 攻击，因此，将窗口内数据分组的抽取比例增加至 40%。若系统进入预警状态是由于数据流检测方式检测出数据流中非正常分组比例超过攻击比例，增加数据分组抽取比例后，依然能检测出非正常分组比例超过攻击比例，则判断系统遭受到低速率式 DDoS 攻击，系统进入防御状态；或链路带宽检测方式检测出交换机端口流量超过攻击阈值，紧接着，数据流检测方式检测出非正常分组比例超过攻击比例，则判断系统遭受到洪泛式 DDoS 攻击，系统进入防御状态。其中，攻击比例是指数据流状态转换时非正常数据分组比例需要达到的指标；攻击阈值是指链路带宽状态转换时交换机端口流量需要达到的指标。系统处于防御状态时，说明已经遭受到 DDoS 攻击，再次将窗口内数据分组的抽取比例提升至 80%，逐一分析抽取的数据分组，并结合 SDCC-P 模块对攻击分组进行处理。系统状态转换如图 3 所示。

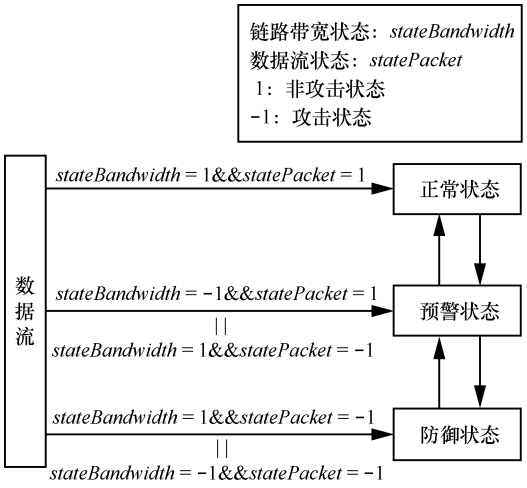


图 3 系统状态转换

SDCC-D 模块通过结合系统实时的链路带宽状态和数据流状态实现系统状态的动态转换。用户突然增大访问量，或用户的非正常访问可能造成链路带宽和数据流异常，为防止误判，SDCC 增加了预警状态，在预警状态加大检测范围，依然能检测出 DDoS 攻击行为，系统才进入防御状态。

3.2.1 基于链路带宽检测

链路带宽检测利用 SDN 控制器能获取底层 OpenFlow 交换机端口的实时流量信息的功能。每隔 T 时间段检测一次当前交换机端口流量信息并与攻击阈值比较，若某个端口的流量大于攻击阈值时，链路带宽状态变为攻击状态；若所有端口的流量都低于攻击阈值时，链路带宽状态变为非攻击状态。基于链路带宽检测方式能对洪泛式 DDoS 攻击有迅速的反应，如 SYN/ACK Flood 攻击。

3.2.2 基于数据流检测

数据流检测是以数据分组的个数 N 为窗口大小，根据系统的不同状态，从每个窗口中随机抽取不同比例的数据分组。由于 SDN 架构转发控制分离的特点，当数据分组进入 OpenFlow 交换机后，若交换机中没有匹配流表，数据分组会被打包在 Packet_in 消息中发送至控制器，因此，能通过控制器提取到 Packet_in 消息中数据分组的属性信息[22,23]。然后结合 CBF 方法，根据提取到的数据分组属性信息查看 profile 表，计算数据分组的 CBF 分数，将 CBF 分数小于丢弃阈值的数据分组判断为非正常分组，其中，丢弃阈值是指判断数据分组是否为非正常分组时，其 CBF 分数需要达到的指标。接着，计算非正常分组的比例，若数据流在非攻击状态下，连续 5 个窗口的非正常分组比例大于攻击比例，表明该数据流中极大可能含有攻击分组，数据流状态变为攻击状态；若数据流在攻击状态下，连续 5 个窗口的非正常分组比例小于攻击比例，表明该数据流含有攻击分组的可能性较低，数据流状态变为非攻击状态；其中窗口数目的选取参考文献[24]，其目的在于降低由于用户非正常访问而造成的误判率。基于数据流检测根据系统不同状态选取不同比例数据分组进行分析的方式，既能缓解原 CBF 方案中检查所有数据分组导致资源消耗较大的问题，又能有效检测出隐蔽在数据流中的低速率式 DDoS 攻击，如 TCP 协议攻击。

3.2.3 更新 profile 表

在 CBF 方法中，为节省 profile 表的存储空间，在非攻击阶段需要对数据流进行 2 次扫描，因此难以达到实时检测系统中数据流的目的。为此，本方案基于 SDN 架构，结合属性对信息库存储数据分组的属性对值信息和出现次数，在更新 profile 表时，直接从属性对信息库中选取比 *minconf* 大的属性对值，与 profile 表中的属性对值比较即可。若属性对

信息库中的属性对值的 *Conf* 比 profile 表中的该属性对值的 *Conf* 大，则用大的 *Conf* 值更新 profile 表；若 profile 表中没有该属性对值信息，则添加到 profile 表中。通过这种方法只需扫描一遍数据流，不需要分阶段地扫描数据流，便于实现数据流的实时检测。为确保 profile 表中存储的是正常数据分组的属性对信息，当数据流进入 SDCC-D 模块，通过 2 种 DDoS 攻击检测方式后，未检测到数据流中有 DDoS 攻击行为，系统进入正常状态后，才会将数据流中的数据分组属性对信息存入属性对信息库，然后更新 profile 表。以 *TTL*、*Flag* 这 2 个属性为例，*minconf* 取值 0.04，改进的 profile 表更新流程如图 4 所示。其中，*minconf* 用于剔除正常数据流中出现次数极少的属性值以提高 profile 表的利用效率，其值根据表中属性对值的 *Conf* 分布来设定，对于不同的 *minconf* 取值，profile 表的更新流程不变。

3.3　攻击防御模块

SDCC-P 模块利用 SDN 控制器提供的北向 REST API 接口向 OpenFlow 交换机下发流表的功能。当系统处于防御状态时，SDCC-P 模块首先会对抽取的数据分组计算 CBF 分数，若 CBF 分数高于丢弃阈值，则判断该数据分组为正常分组，并通知控制器下发流表至交换机转发该数据分组；若 CBF 分数低于丢弃阈值，则判断该数据分组为攻击分组，控制器会调用北向 REST API 接口下发流表至 OpenFlow 交换机丢弃该数据分组，此后该类攻击分组都会在交换机处被拦截，从而达到在攻击源 OpenFlow 交换机处拦截攻击分组的目的。由于 SDN 架构转发控制分离的机制，当有同类攻击分组到达 OpenFlow 交换机时，不需要再经过控制器判断，就可以在交换机处直接通过流表匹配丢弃该分组，加快了系统对攻击分组的反应速度。

攻击流特征库用来存储已被识别出且被系统拦截的攻击分组信息。由于 DDoS 攻击，特别是洪泛式 DDoS 攻击，会迅速产生大量的攻击分组，在攻击者开始发起攻击时，交换机中没有匹配流表，大量的攻击分组会同时涌向控制器，且这些攻击分组中有很多类型相同，而控制器无法迅速处理突然涌入的大量数据分组。为减小控制器压力，避免控制器对每个攻击分组都下发流表而导致数据处理速度减慢的问题，SDCC-P 模块在每判断出一个攻击分组后，就将该攻击分组的属性信息存到攻击流特征库中，然后控制器针对该分组下发一个流表至 OpenFlow 交换机；对之后经过控制器的攻击分组，首先，判断是否已存在于攻击流特征库，存在则表明控制器已下发过匹配流表，不需要再分析该攻击分组，直接将其发给交换机通过流表匹配来丢弃该攻击分组；不存在则表明这是一个新的攻击分组，就先将这个新攻击分组属性信息添加到攻击流特征库中，然后控制器下发一个新的流表至交换机来丢弃该新攻击分组。通过这种方式不仅能加快控制

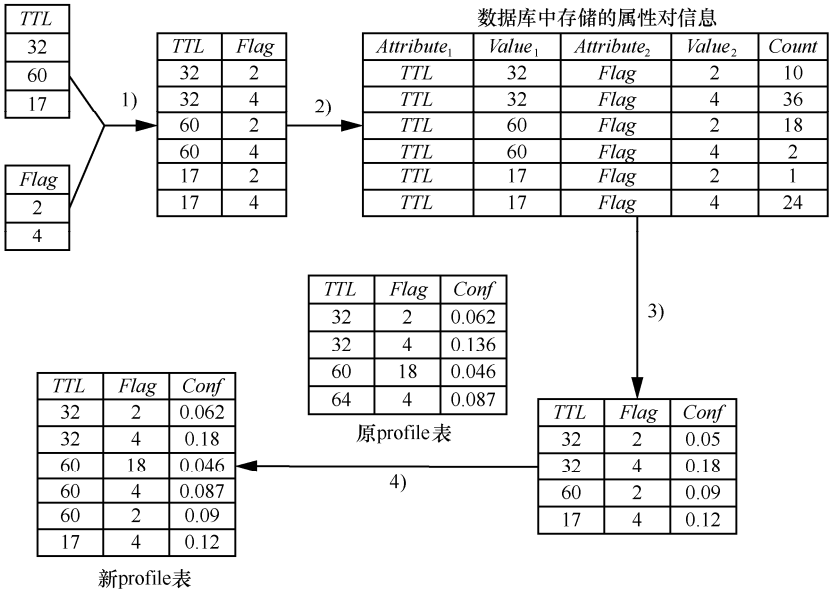


图 4　profile 表更新流程

器的数据处理速度，也能减少交换机中功能重复的流表，降低交换机的内存消耗。

4 方案分析与评估

本节首先介绍了实验环境以及在实验中的参数设置，然后分别模拟了洪泛式 DDoS 攻击和低速率式 DDoS 攻击这 2 组实验，并对实验结果进行了分析，最后将 SDCC 与其他相关方案进行性能比较与评估。

4.1 实验环境

实验中使用阿里云作为云环境，使用网络仿真软件 mininet[25]来模拟主机、服务器和 OpenFlow 交换机，并搭建网络拓扑，其中，mininet 的运行内存为 1 GB。使用开源的 SDN 控制器 Floodlight 与 OpenFlow 交换机通信，在阿里云服务器中内存为 8 GB，CPU 为 2.5 GHz 的 Windows 系统上运行控制器。实验中 Floodlight 控制器（C0）和 mininet 虚拟机都运行在阿里云服务器上。其中，mininet 模拟了 3 台网络主机（H1、H2、H3）、2 台 OpenFlow 交换机（SW1、SW2）和一台服务器（S0），实验拓扑如图 5 所示，其中实线表示底层网络数据通信链路，虚线表示 Floodlight 控制器与 OpenFlow 交换机之间传输控制信息的专用通道。

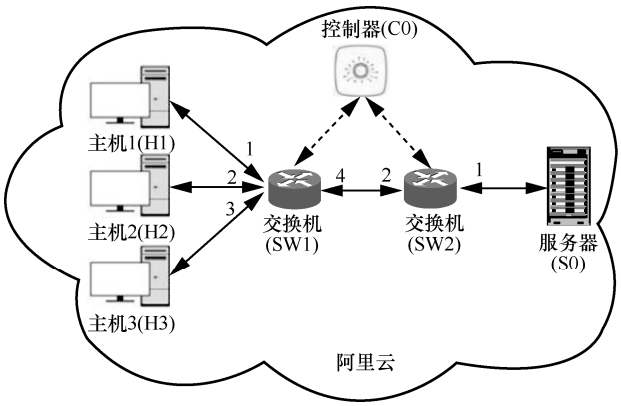


图 5 实验拓扑

本文分别模拟了洪泛式 DDoS 攻击和低速率式 DDoS 攻击这 2 组实验，都是通过云环境中的主机向云环境中的服务器发起攻击。由于 SDCC 通过数据分组的属性信息检测 DDoS 攻击，对于无论是云内还是云外的攻击者，其检测和防御方式均相同，因此，可以通过模拟云内的主机向服务器发起攻击的实验来说明 SDCC 能有效检测和防御云环境中的 DDoS 攻击。

在实验中，H1、H2、H3 通过 hping3 工具发送 TCP 数据分组以模拟主机对服务器的正常访问、低速率式 DDoS 攻击和洪泛式 DDoS 攻击。其中，攻击密度是指主机每秒发送攻击分组的个数，低速率式 DDoS 攻击实验的攻击密度分别为 100 个/秒和 1 000 个/秒，洪泛式 DDoS 攻击实验的攻击密度分别为 10^4 个/秒和 10^6 个/秒。在本实验中使用主机 H1 模拟正常用户访问，H2、H3 这 2 台主机模拟攻击者向 S0 发起 DDoS 攻击，对于更复杂的网络情况，可做类似的实验分析，并得出相似的实验结果。

实验参数取值如表 1 所示。*Interval* 用于系统链路带宽监测，每隔 1 s 系统检测一次链路当前带宽，并判断当前链路状态。*AttackThreshold* 用于链路状态判断，若链路带宽大于 *AttackThreshold*，则表示系统可能遭受了 DDoS 攻击，链路状态变为攻击状态。*WindowSize* 用于系统数据流监测，在控制器每获取 50 个数据分组时分析一次数据流状态。*DiscardThreshold* 用于数据分组判断，若数据分组的 CBF 分数低于 *DiscardThreshold*，则表示该数据分组为非正常分组。*AttackRatio* 用于数据流状态判断，在非攻击状态下，数据流中非正常分组比例高于 *AttackRatio* 时，表示数据流可能遭受到 DDoS 攻击，数据流状态变为攻击状态；在攻击状态下，数据流中非正常分组比例低于 *AttackRatio* 时，表示数据流已是正常数据流，数据流状态变为非攻击状态。表 1 给出了 SDCC 中采取的一组典型参数设定，在实际环境中，可以以这组参数值作为标准，然后根据网络运行情况对参数取值进行适当调整。

表 1 实验参数取值

参数	参数描述	参数取值
Interval	时间间隔/s	1
WindowSize	窗口大小	50
AttackThreshold	攻击阈值	80
DiscardThreshold	丢弃阈值	0.3
AttackRatio	攻击比例	50%（非攻击状态下） 25%（攻击状态下）

4.2 实验结果与分析

在洪泛式 DDoS 攻击和低速率式 DDoS 攻击这 2 组实验中，通过结合 SW1 的端口 1、端口 2、端口 3 和 SW2 的端口 1 的实时流量以及系统的链路状态和数据流状态来分析 SDCC 的 DDoS 攻击防御效果。

在攻击密度为 10^4 个/秒的洪泛式 DDoS 攻击模

拟实验中，从第 4 s 开始，H2 向 S0 发送大量攻击分组，如图 6 所示。在第 4 s 时 SW1 的端口 2 吞吐量超过 *AttackThreshold*，此时，系统的链路带宽状态变为攻击状态，如图 7 所示，系统进入预警状态，紧接着，数据流状态也变为攻击状态，由于该攻击密度下，一个窗口内数据流的检测时间低于 100 ms，时间间隔很短，因此，在图 7 中第 4 s 时，数据流的状态也是攻击状态，由此系统进入防御状态，会对 H2 发送的攻击分组进行拦截。由图 6 可知，从第 7 s 开始 H3 也对 S0 发送大量的攻击分组，此时，已处于防御状态下的系统会直接对 H3 发送的攻击分组进行拦截，因此，从第 4 s 开始，SW2 的端口 1 和 SW1 的端口 1 的实时吞吐量一致，表示 S0 只接收到了 H1 发送的正常数据分组，而 H2、H3 发送的攻击分组被 SDCC-P 模块成功拦截。此时，交换机中已存在拦截 H2、H3 发送的攻击分组的流表，无论系统处于什么状态，都会对这些攻击分组进行拦截，由图 7 可知，从第 23 s 开始，系统已处于正常状态，在图 6 中 SW2 的端口 1 吞吐量始终与 SW1 的端口 1 保持一致。实验表明 SDCC 能高效检测并防御洪泛式 DDoS 攻击。

在攻击密度为 10^6 个/秒的洪泛式 DDoS 攻击模拟实验中，从第 4 s 开始，H2、H3 同时向 S0 发送大量攻击分组，如图 8 所示。在第 4 s 时 SW1 的端口 2、端口 3 吞吐量都超过 *AttackThreshold*，此时，系统进入预警状态，由于该攻击密度下，一个窗口内数据流的检测时间在 100 ms 左右，时间间隔很短，因此，在第 4 s，数据流状态也变为攻击状态，如图 9 所示，系统进入防御状态，对 H2、H3 发送的攻击分组进行拦截。因此，从第 4 s 开始，SW2 的端口 1 处的吞吐量始终与 SW1 的端口 1 保持一致。实验表明 SDCC 能有效防御不同攻击密度的洪泛式 DDoS 攻击。

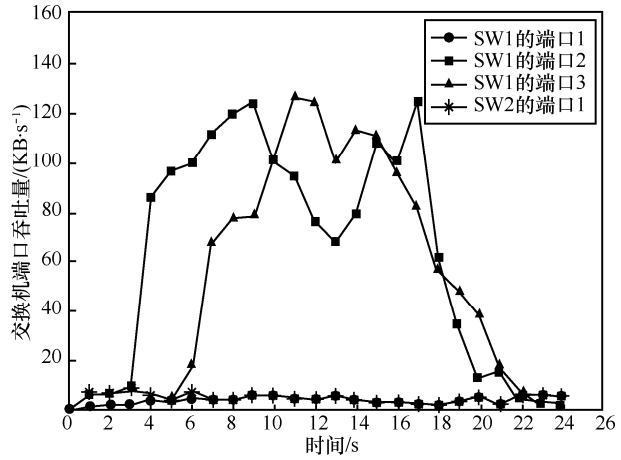


图 6　洪泛式 DDoS 攻击下交换机端口实时吞吐量变化
（攻击密度为 10^4 个/秒）

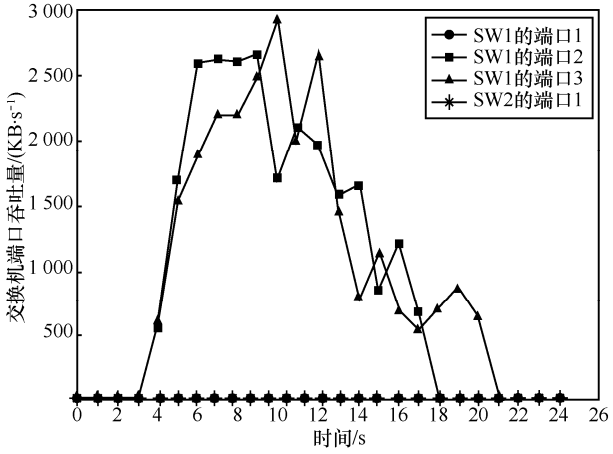


图 8　洪泛式 DDoS 攻击下交换机端口实时吞吐量变化
（攻击密度为 10^6 个/秒）

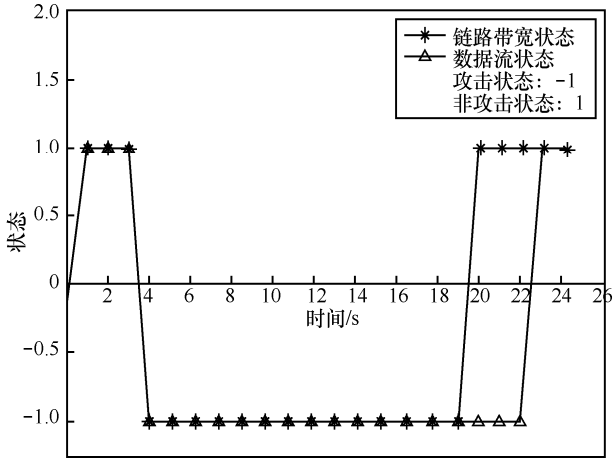


图 7　洪泛式 DDoS 攻击下链路状态和数据流状态变化
（攻击密度为 10^4 个/秒）

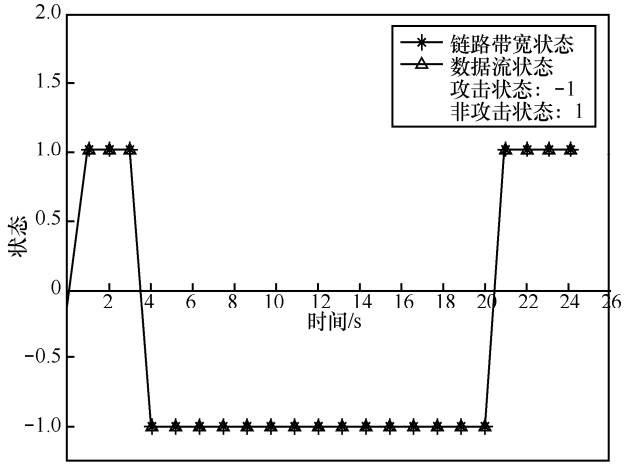


图 9　洪泛式 DDoS 攻击下链路状态和数据流状态变化
（攻击密度为 10^6 个/秒）

在攻击密度为 100 个/秒的低速率式 DDoS 攻击模拟实验中，H1 向 S0 发送的一直都是正常分组，从第 4 s 开始，H2 向 S0 不间断地发送小流量的攻击分组，如图 10 所示，从第 4 s 到第 5 s，SW2 的端口 1 实时吞吐量高于 SW1 的端口 1，表示 S0 能接收到 H2 发送的少量攻击分组，但是由于此时攻击流量非常小，对系统造成的影响很小。在第 6 s 时，非正常分组的比例超过 *AttackRatio*，SDCC-D 模块会检测出数据流异常，系统进入预警状态，之后，依然检测出数据流异常，系统进入防御状态，对 H2 发送的攻击分组进行拦截，如图 11 所示，在第 6 s 时，链路带宽为正常状态，数据流为攻击状态，此时，图 10 中 SW2 的端口 1 与 SW1 的端口 1 的实时吞吐量一致。从第 7 s 开始，H3 开始向 S0 发送攻击分组，此时，系统已处于防御状态，能直接对 H3 发送的攻击分组进行拦截，如图 10 所示，从第 6 s 开始，SW2 的端口 1 始终与 SW1 的端口 1 吞吐量一致，表示 SDCC-P 模块成功地拦截了 H2、H3 发出的攻击分组。实验表明 SDCC 能高效检测并防御低速率式 DDoS 攻击。

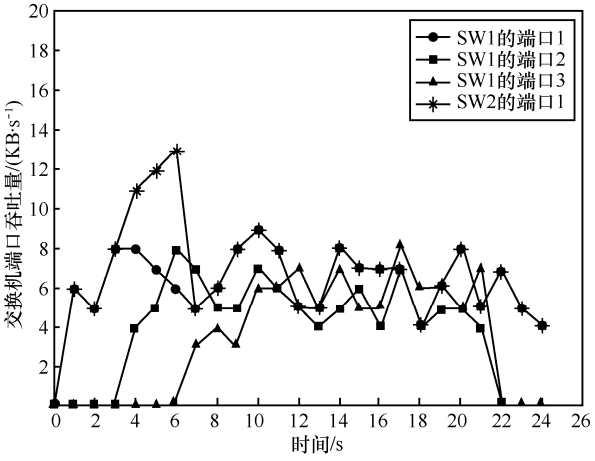


图 10 低速率式 DDoS 攻击下交换机端口实时吞吐量变化
（攻击密度为 100 个/秒）

在攻击密度为 1 000 个/秒的低速率式 DDoS 攻击模拟实验中，从第 4 s 开始，H2、H3 同时向 S0 不间断地发送小流量的攻击分组，如图 12 所示，虽然 H2、H3 发出的攻击流的链路带宽都小于 *AttackThreshold*，但由于在 SW1 处汇聚后，使 SW1 处的链路带宽大于 *AttackThreshold*，链路带宽变为攻击状态，紧接着，数据流状态也变为攻击状态，系统进入防御状态，如图 13 所示，在第 4 s 时，链路带宽和数据流都为攻击状态。从图 12 可以看出，从第 4 s 开始，S0 只接收到了 H1 发送的正常数据分组，表示 SDCC-P 模块成功地拦截了 H2、H3 发出的攻击分组。实验表明 SDCC 能有效防御不同攻击密度的低速率式 DDoS 攻击。

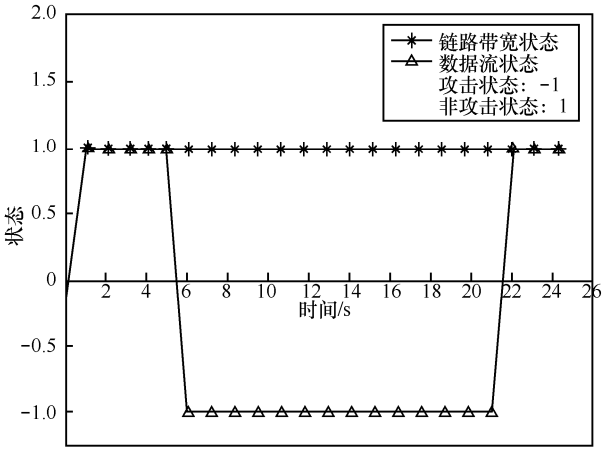


图 11 低速率式 DDoS 攻击下链路状态和数据流状态变化
（攻击密度为 100 个/秒）

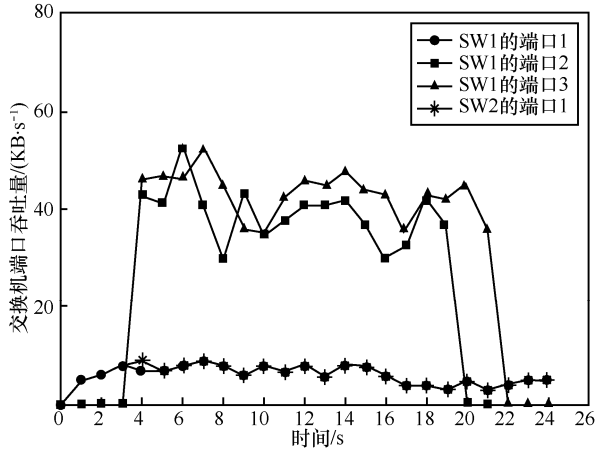


图 12 低速率式 DDoS 攻击下交换机端口实时吞吐量变化
（攻击密度为 1 000 个/秒）

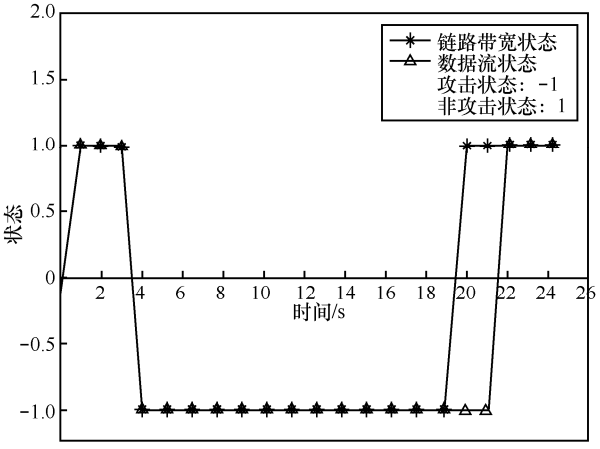


图 13 低速率式 DDoS 攻击下链路状态和数据流状态变化
（攻击密度为 1 000 个/秒）

4.3　性能评估

图 14 显示了在攻击时期、不同攻击密度下，PacketScore[6]、CBF[8]、SDCC 这 3 种方案对一个窗口内数据分组的处理时间的对比情况。为了便于比较，在 3 种方案中，统一设置为一个窗口内包含 50 个数据分组，攻击密度范围为 10^3~10^4 个/秒。从图 14 中可以看出，相对于 PacketScore 方案，CBF 方案对每个窗口内的数据处理速度更快，这是由于 PacketScore 方案采取的是动态阈值的方法，在攻击检测时既需要逐一计算数据分组是攻击分组的可能性，又需要计算近期的阈值，因此，对一个窗口的数据处理时间花费最多。CBF 方法在这一方面做了改进，为加快攻击时期数据处理速度，使用的是静态阈值的方法，而且计算数据分组的 CBF 分数时只需要查看 profile 表，其算法复杂度为 $O(1)$。SDCC 在一个窗口内的数据处理时间最少，这是由于 SDCC 在 CBF 方案的基础上做了进一步的优化，在攻击数据流中，有很多相同的攻击分组，而 SDCC 并没有像前 2 种方案一样对数据分组进行逐一处理，而是随机抽取窗口内一定比例的数据分组逐一计算 CBF 分数再进行判断；另外，通过攻击流特征库存储已识别的攻击分组，当再出现库中已有的攻击分组需要判断时，不做处理并直接转发至交换机，通过流表拦截该攻击分组。这种方法有效提升了对一个窗口内数据分组的处理速度，相比于 CBF 方法，其数据处理速率提升了约 19.4%；此外，对于 CBF 方法，当发送分组密度越大时，对于攻击流中相同的攻击分组，控制器下发相同的流表数量越多，其窗口数据处理时间越长，而 SDCC 的攻击流特征库能有效缓解重复流表的问题，因此，由图 14 可以看出，随着攻击密度增大，CBF 和 PacketScore 方法的数据处理时间上升的趋势比 SDCC 更明显。

系统在正常状态下，相对于 CBF 方案，SDCC 简化了 profile 表的更新，因此 CPU 的占用率更低。由表 2 可以看出，当进入系统的是正常数据流时，SDCC 的 CPU 占用率比 CBF 低 2.6%；在系统遭受 DDoS 攻击状态下，由于 SDCC 采取的是比例抽取数据分组进行检测，且使用了攻击流特征库避免了对相同攻击分组的重复检测，相对于 CBF 方案的逐一检测数据分组而言，对 CPU 的使用率要低得多，特别是在洪泛式这种大流量的 DDoS 攻击下，CBF 方法需要迅速对大量数据分组计算分数并产生流表，SDCC 由于比例抽取和攻击流特征库有效缓解了控制器的压力。由表 2 可知，当进入系统的是低速率式 DDoS 攻击数据流时，SDCC 的 CPU 占用率比 CBF 的低 11.1%；而当进入系统的是洪泛式 DDoS 攻击数据流时，SDCC 的 CPU 占用率比 CBF 的低 19.7%。

表 2　不同类型数据流下 CPU 占用率比较

数据流类型	CPU 占用率	
	SDCC	CBF
正常数据流	5.1%	7.7%
低速率式 DDoS 攻击数据流	7.2%	18.3%
洪泛式 DDoS 攻击数据流	15.8%	35.5%

图 15 显示了在不同发送分组密度下 SDCC 和 CBF 方案的误判率比较，发送分组密度的范围是 10~10^6 个/秒。误判率是指将正常数据分组判断成攻击分组的概率。

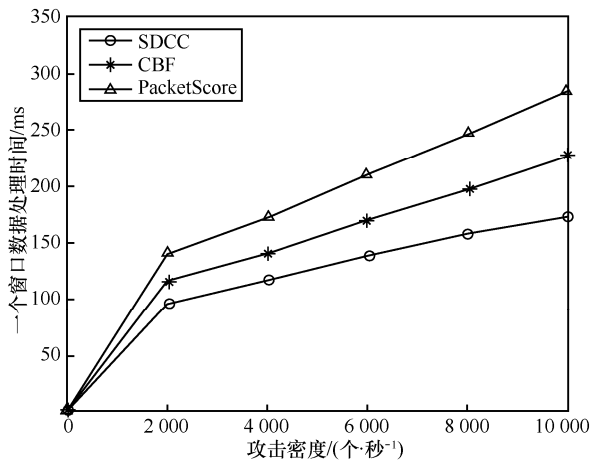


图 14　对一个窗口大小的数据分组处理时间比较

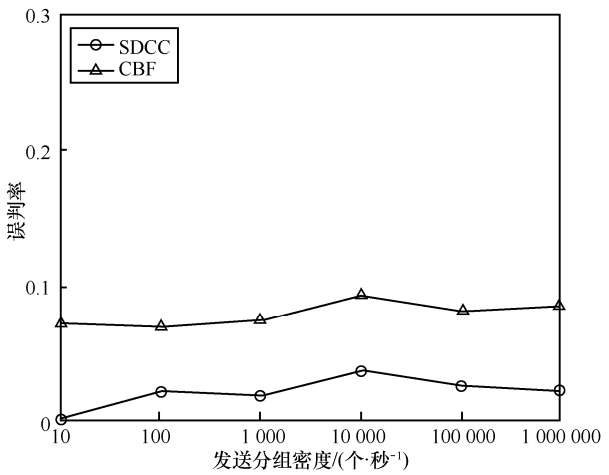


图 15　误判率比较

由图 15 可以看出，随着攻击密度增大，2 种方案的误判率波动不大，表明这 2 种方案的误判率不受攻击强度影响，都适用于云环境。与 CBF 方案中只要检测到 CBF 分数比阈值低就判断为攻击分组所不同的是，SDCC 通过检测数据流中 CBF 分数小于丢弃阈值的数据分组比例来判断系统中是否有 DDoS 攻击行为，只有确定系统遭受到 DDoS 攻击，进入防御状态，才会对 CBF 分数比丢弃阈值低的数据分组进行拦截，这种检测方式能降低由于用户自身非习惯性访问引起的误判率。通过图 15 可以看出，SDCC 的误判率比 CBF 方案低至少 4.63%。

5 结束语

DDoS 攻击已成为云环境中主要的安全威胁之一。本文针对云环境中的 2 类典型的 DDoS 攻击问题，即洪泛式 DDoS 攻击与低速率式 DDoS 攻击，提出 SDCC，它使用 SDN 架构，简化了数据分组转发过程，控制器能实时获取全局的网络状态；综合使用链路带宽检测和数据流检测这 2 种方式，能准确地检测出 2 种类型的 DDoS 攻击；通过改进 CBF 方案的 profile 更新过程，有利于实时检测，而且根据系统的不同状态按比例抽取数据分组检测，并使用攻击流特征库存储已识别的攻击分组，降低了控制器的压力，加快了检测速度。与其他相关方案相比，SDCC 处理数据的速度更快，消耗的计算资源更少，且误判率更低，从而有效地保护云系统免受 DDoS 攻击。

与传统网络相比，SDN 架构转发控制分离的特点简化了数据转发过程，提高了数据分组的检测效率，更有利于实时检测系统状态。但随着云计算被更广泛的应用，业务流量规模变得更大，使单控制器的 SDN 架构中控制器的压力越来越大；同时，由于网络管理都集中于控制器一点，一旦控制器发生故障，将存在整个网络不可用的风险。因此，为适应流量规模更大的网络环境，克服 SDN 架构管理过于集中带来的弱点，在未来工作中将研究使用多控制器分区管理的策略，既能缓解大规模流量环境中控制器的压力，且在部分控制器发生故障时，网络的安全服务仍然能够正常运行。

参考文献：

[1] YAN Q, YU F R, GONG Q, et al. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges[J]. IEEE Communications Surveys & Tutorials, 2016, 18(1): 602-622.

[2] CHEN L C, LONGSTAFF T A, CARLEY K M. Characterization of defense mechanisms against distributed denial of service attacks[J]. Computers & Security, 2004, 23(8): 665-678.

[3] PENG T, LECKIE C, RAMAMOHANARAO K. Survey of network-based defense mechanisms countering the DoS and DDoS problems[J]. ACM Computing Surveys, 2007, 39(1): 3.

[4] TARIQ U, HONG M P, LHEE K. A comprehensive categorization of DDoS attack and DDoS defense techniques[C]//International Conference on Advanced Data Mining and Applications. 2006: 1025-1036.

[5] SPECHT S M, LEE R B. Distributed denial of service: taxonomies of attacks, tools, and countermeasures[C]// The 17th International Conference on Parallel and Distributed Computing Systems. 2004: 543-550.

[6] KIM Y, LAU W C, CHUAH M C, et al. PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks[J]. IEEE Transactions on Dependable & Secure Computing, 2006, 3(2): 141-155.

[7] 胡汉卿. 基于云计算DDoS攻击防御研究[D]. 南京: 南京邮电大学, 2015.
HU H Q. Research on DDoS attack defense based on cloud computing[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2015.

[8] DOU W, CHEN Q, CHEN J. A confidence-based filtering method for DDoS attack defense in cloud environment[J]. Future Generation Computer Systems, 2013, 29(7): 1838-1850.

[9] SHAMSOLMOALI P, ALAM M A, BISWAS R. C2DF: high rate DDOS filtering method in cloud computing[J]. International Journal of Computer Network & Information Security, 2014, 6(9): 43-50.

[10] SAHI A, LAI D, LI Y, et al. An efficient DDoS TCP flood attack detection and prevention system in a cloud environment[J]. IEEE Access, 2017, 5: 6036-6048.

[11] JEYANTHI N, BARDE U, SRAVANI M, et al. Detection of distributed denial of service attacks in cloud computing by identifying spoofed IP[J]. International Journal of Communication Networks & Distributed Systems, 2013, 11(3): 262-279.

[12] 吴志军, 张东. 低速率DDoS攻击的仿真和特征提取[J]. 通信学报, 2008, 29(1): 71-76.
WU Z J, ZHANG D. Simulation and feature extraction of low rate DDoS attacks[J]. Journal on Communications, 2008, 29(1): 71-76.

[13] NAVAZ A S S, SANGEETHA V, PRABHADEVI C. Entropy based anomaly detection system to prevent DDoS attacks in cloud[J]. International Journal of Computer Applications, 2013, 62(15): 42-47.

[14] WANG B, ZHENG Y, LOU W, et al. DDoS attack protection in the era of cloud computing and software-defined networking[J]. Computer Networks, 2015, 81(C): 308-319.

[15] KALLIOLA A, LEE K, LEE H, et al. Flooding DDoS mitigation and traffic management with software defined networking[C]//International Conference on Cloud Networking. 2015: 248-254.

[16] ZHANG C, CAI Z, CHEN W, et al. Flow level detection and filtering of low-rate DDoS[J]. Computer Networks the International Journal of Computer & Telecommunications Networking, 2012, 56(15): 3417-3431.

[17] HOQUE N, BHATTACHARYYA D K, KALITA J K. A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis[C]//International Conference on Communication Systems and Networks. 2016: 1-2.

[18] 孙义明, 杨丽萍. 信息化战争中的战术数据链[M]. 北京: 北京邮电大学出版社, 2005.
SUN Y M, YANG L P. Tactical data chain in information warfare[M]. Beijing: Beijing University of Posts and Telecommunications Press, 2005.

[19] 田开琳, 李明. 一种可靠检测低速率 DDoS 攻击的异常检测系统[J]. 现代电子技术, 2009, 32(7): 68-71.
TIAN K L, LI M. An anomaly detection system for reliable detection of low rate DDoS attacks[J]. Modern Electronic Technology, 2009, 32(7): 68-71.

[20] 左青云, 陈鸣, 赵广松, 等. 基于 OpenFlow 的 SDN 技术研究[J]. 软件学报, 2013(5): 1078-1097.
ZUO Q Y, CHEN M, ZHAO G S, et al. Research of SDN technology based on OpenFlow[J]. Journal of Software, 2013(5): 1078-1097.

[21] LIU T C, YANG B H, ZHANG Y, et al. Data packet processing in SDN[P]. US20150281127, 2015.

[22] FOUNDATION O N. Software-defined networking: the new norm for networks[R]. ONF White Paper, 2012.

[23] NADEAU T D, GRAY K. 软件定义网络: SDN 与 OpenFlow 解析[M]. 毕军, 单业, 张绍宇, 等译. 北京: 人民邮电出版社, 2014.
NADEAU T D, GRAY K. Software defined network: SDN and OpenFlow parsing[M]. Translated by BI J, SHAN Y, ZHANG S Y, et al. Beijing: Posts & Telecom Press, 2014.

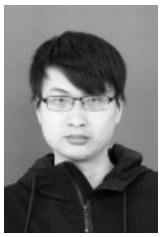
[24] MOUSAVI S M, STHILAIRE M. Early detection of DDoS attacks against SDN controllers[C]//International Conference on Computing, NETWORKING and Communications. 2015:77-81.

[25] LANTZ B, HELLER B, MCKEOWN N. A network in a laptop:rapid prototyping for software-defined networks[C]//ACM Workshop on Hot Topics in Networks. 2010:1-6.

[作者简介]

何亨（1981-），男，湖北武汉人，博士，武汉科技大学副教授，主要研究方向为云计算、软件定义网络、网络安全等。

胡艳（1993-），女，湖北黄冈人，武汉科技大学硕士生，主要研究方向为云计算、软件定义网络、网络安全等。



郑良汉（1995-），男，湖北武汉人，武汉科技大学硕士生，主要研究方向为云计算、网络安全等。

薛正元（1989-），男，河南社旗人，华中科技大学博士生，主要研究方向为云计算、大数据技术等。