基于 OpenFlow 交换机洗牌的 DDoS 攻击动态防御方法

武泽慧* 魏 强 任开磊 王清贤
(解放军信息工程大学网络空间安全学院 郑州 450001)
(数学工程与先进计算国家重点实验室 郑州 450001)

摘 要：网络资源的有限性和网络管理的分散性是传统网络难以解决分布式拒绝服务攻击问题的重要原因。当前的防御方法存在静态性、滞后性的不足，并且难以定位攻击者。针对上述问题，该文提出一种动态防御的方法。利用软件定义网络(SDN)集中控制和动态管理的特性构建 OpenFlow 交换机洗牌模型，使用贪心算法实现用户-交换机连接的动态映射，通过多轮洗牌区分出用户群中的攻击者和合法用户，对合法用户提供低延迟不间断服务。在开源 SDN 控制器 Ryu 上实现原型系统，并在 SDN 环境下进行测试。性能测试结果表明采用该方法可以通过有限次的洗牌筛选出攻击者，降低 DDoS 攻击对合法访问的影响；能力测试结果则说明了在由一个控制器组成的环形拓扑结构下该方法的防御效果与攻击流的大小无关，而是仅与攻击者的数目有关。

关键词：网络安全；软件定义网络；分布式拒绝服务攻击；动态防御
中图分类号：TP393.08 **文献标识码**：A **文章编号**：1009-5896(2017)02-0397-08
DOI: 10.11999/JEIT160449

Dynamic Defense for DDoS Attack Using OpenFlow-based Switch Shuffling Approach

WU Zehui WEI Qiang REN Kailei WANG Qingxian
(*Institute of Cyber Security, PLA Information Engineering University, Zhengzhou* 450001, *China*)
(*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou* 450001, *China*)

Abstract: The limitations of network resource and the dispersion of network management are the two major difficulties for traditional networks to address the Distributed Denial of Service (DDoS) attacks. However, current defense methods are static and hysteresis, which are unable to locate the attackers accurately. Therefore, a dynamic defense using the two pivotal features, centralized control and dynamic management, of Software Defined Networks (SDN) is proposed. An OpenFlow-based switch shuffling model is built which employs greedy algorithm to remap user-switch link dynamically. After several shuffling, attacker could be differentiated from legitimate users and provide the latter with low latency uninterrupted services. The proposed approach is implemented in Ryu, the open source SDN controller, and the prototype is tested in a real SDN. The results of performance test show that with this approach attackers in limited times of shuffling can be isolated and the effects of DDoS attacks on legal flows can be reduced. The outcomes of defense ability test demonstrate that the efficiency of the proposed dynamic approach has nothing to do with the size of attack flow, but is only related to the number of attackers in the ring topology structure which is composed of a single controller.

Key words: Cyber security; Software Defined Networks (SDN); Distributed Denial of Service (DDoS); Dynamic defense

1 引言

分布式拒绝服务(Distributed Denial of Service, DDoS)攻击是一种由拒绝服务(Denial of Service, DoS)攻击衍生出的攻击技术[1]。攻击者借助大量受控主机向目标发起攻击，将拒绝服务攻击的一对一攻击方式扩展为多对一，因此危害更大且更难防范[2]。2015 年第 1 季度，全球范围内发生的 DDoS 攻击比 2014 年同期增长了 100%。游戏、软件等行业被攻击情况严重，平均每次攻击会造成约 50 万美元的损失，攻击者甚至会要求受害者支付赎金以停止攻击[3]。

收稿日期：2016-05-03；改回日期：2016-09-27；网络出版：2016-11-14
*通信作者：武泽慧 wuzehui2010@foxmail.com
基金项目：国家 863 计划项目(2012AA012902)，国家杰出青年科学基金(61402526)
Foundation Items: The National 863 Program of China (2012AA012902), The National Science Fund for Distinguished Young Scholars (61402526)

传统网络中 DDoS 攻击难以消除的根源在其设计缺陷[1,4]。网络设计初衷是保证端到端的通信，关注的是通过发送端和接收端来实现 QoS(Quality of Service)、稳定传输、安全保证等，导致网络两端很复杂，而网络本身相对简单，仅仅负责数据转发。因此，网络两端的任何一端出现恶意行为，都会给对方造成损害，网络本身不具备流量管理的任务和能力。

软件定义网络(SDN)的控制与转发分离，集中控制，动态可编程，细粒度流量审查，以及动态流规则更新的特征使其在分布式拒绝服务攻击的检测和防御上具备一定的优势。现有关于 SDN 拒绝服务攻击的研究多是将传统流量审查、分析方法应用到 SDN 中，或者利用 SDN 集中管理的特性提高流量审查的效率和准确度。Braga 等人[5]于 2010 年率先在 NOX 控制器上实现检测分布式拒绝服务攻击的安全应用程序。控制器从网络流量中提取需要的信息，并从中提取 DDoS 特征。将提取到的特征输入到自组织图(Self-Organizing Map, SOM)中，使用 SOM 配合机器学习算法检测 DDoS 攻击的数据流。该方法具有较高的检测能力和较低的误报。但是受限于特征提取和机器学习算法的效率，在控制器上部署该系统后会降低控制器的处理效率。2012 年 Yeganeh 等人[6]提出降低控制器负载的 Kandoo 框架，该框架不仅可以缓解 DDoS 攻击，也有助于增强网络的鲁棒性。Kandoo 采用两层控制器结构，包括底层控制器平面和顶层控制器平面，其中底层控制器平面负责单一区域内高频度的事件处理，顶层控制器平面负责管控整个 SDN，处理底层控制器无法处理的事件。采用上述方法可以在一定程度上缓解控制器 DDoS 的攻击流，但是该方法仅能降低负载，无法识别 DDoS 攻击流，不能从根本上解决 DDoS 攻击的问题。2013 年 11 月， Shin 等人[7]提出 AVANT-GUARD 技术。AVANT-GUARD 通过包含 Connection Migration 和 Actuating Triggers 两项技术对数据平面以及数据到控制平面的安全性能进行了强化，Connection Migration 用以防止数据平面的饱和攻击；Actuating Triggers 通过给 SDN 交换机提供条件触发器提高了交换机的响应速度。AVANT-GUARD 可以防御 TCP-SYN 洪泛攻击，但是其代价是牺牲了很大的系统系能，并且无法抵御应用程序层的 DoS 攻击，也无法防御应用层使用 UDP 或者 ICMP 协议的攻击。

针对 DDoS 利用僵尸网络进行攻击的问题，Lim 等人[8]提出 DBA(DDoS Blocking Application)的僵尸网络阻断方法。DBA 作为 SDN 控制器应用软件安装在控制器之上，负责监控网络中的每个流量。当检测到 DDoS 攻击时，合法流量会被重定向到另一个备用服务器上，而攻击流则会被当前控制器阻断。由于该方法需要对每个数据流都进行监控，因此大规模应用受限；其次 SDN 中流规则是主动式安装，对控制器负载要求较高。

通过上述介绍可知，当前针利用 SDN 防御拒绝服务攻击主要集中在流量迁移中，利用 SDN 路由动态更新的策略实现攻击流的负载均衡，降低攻击对网络的影响。与当前研究不同，本文提出一种动态防御的方法，其核心是利用 SDN 的特性实现交换机洗牌，通过多轮洗牌区分出用户群中的攻击者和合法用户定位出攻击者，当攻击者被隔离后，攻击流也不会再对网络造成影响。

具体来说本文的贡献如下：

(1)提出一种 OpenFlow 交换机洗牌模型，可以实现 DDoS 攻击的防御和定位。

当前 DDoS 防御方法负载过高，灵活性较差。对此，本文通过交换机轮换的方式实现对攻击流的分向转发，通过交换机洗牌的方法实现攻击者的快速定位，灵活性高，负载增加有限。

(2)基于 SDN 集中控制和动态管理的特性提出一种服务不间断的 DDoS 防御方法。

本文方法利用 SDN 集中控制和动态管理的特性，使用控制器控制交换机轮换提供不间断服务，可以实现在 DDoS 攻击环境下对从攻击流中分离出的合法访问者继续提供服务。

(3)在开源 SDN 控制器 Ryu 上实现原型系统，并在具体 SDN 环境下进行测试。

论文组织结构：第 2 节概述本文 DDoS 攻击动态防御的原理和过程；第 3 节结合 OpenFlow 协议，提出交换机洗牌方法，并对该方法的复杂度、准确性进行分析介绍；第 4 节对方法进行仿真实验，通过多个对比结果阐述本文方法的可行性，并对性能进行测试评估；第 5 节对全文进行总结。

2 DDoS 动态防御系统

DDoS 动态防御方法使用 SDN 实现动态防御，通过控制器动态管理配置 OpenFlow[1]交换机(以下简称交换机)，使用交换机洗牌方法完成攻击流的分离与攻击者的定位，下面对动态防御方法的体系结构进行概述。

图 1 所示为 DDoS 动态防御方法的框架图，用户的网络流量通过 SDN 转发至被保护的 Web Server 中。正常状态下，用户通过 Web Server 的

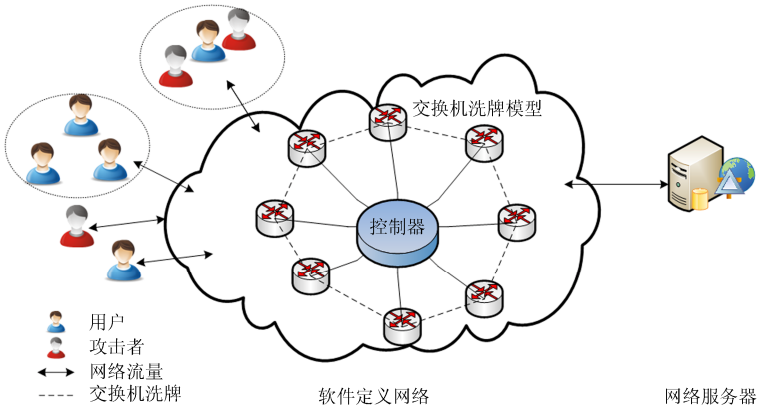


图 1　DDoS 动态防御方法框架图

DNS 域名对其进行访问，SDN 控制器通过安全通道对交换机进行配置，确定一条转发路由，由交换机负责数据包的转发。本文在交换机上部署流量吞吐率检测器，吞吐率的阈值由控制器设定。由于在用户群中潜藏攻击者，攻击者发动 DDoS 攻击会导致交换机的吞吐率剧增，交换机一旦检测到吞吐率大于阈值，即认为攻击者发动了 DDoS 攻击，SDN 进入被攻击状态。SDN 控制器启用交换机洗牌算法对交换机进行动态调度，该算法从所有用户中分离出合法用户，并调度交换机继续为其提供不间断服务，同时分离出攻击者并对其进行隔离定位。本文方法适用于使用 SDN 防御传统网络的 DDoS 攻击的场景，而针对 SDN 的 DDoS 攻击，除了传统的流量攻击，也存在如拓扑欺骗攻击等方式，本文方法如果应用在该场景中，则仅能防御基于流量的 DDoS 攻击，对于拓扑欺骗类型的攻击无效。

2.1 交换机分层

DDoS 动态防御方法使用交换机对 DDoS 流量进行分离，并通过交换机的洗牌算法实现攻击定位。为实现上述目标，并提高系统效率，首先需要对交换机进行分层处理。本文将交换机分为两层：代理层交换机(proxy switches)和隐藏层交换机(hidden switches)，如图 2 所示。

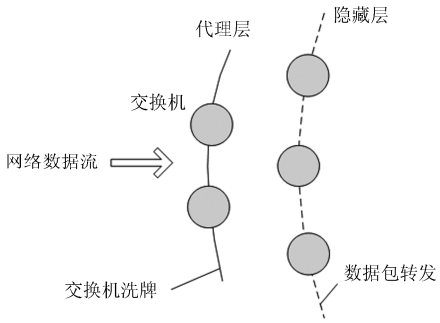


图 2 交换机分层模型图

其中代理层的交换机是直接面向用户端的，负责对来自用户的网络流量进行转发的第 1 层交换机。隐藏层交换机负责筛选后的流量转发。

2.2 交换机洗牌

本文利用 SDN 集中控制和动态管理的特性实现交换机洗牌算法。其实现原理如图 3 所示。图中 7 个用户(用户 1 至用户 7，用户 3 和用户 5 是潜藏的攻击者)分别通过 3 个代理层交换机(S1, S2, S3)完成数据转发，其中，用户 1, 2, 3 由 S1 转发，用户 4, 5 由 S2 转发，用户 6, 7 由 S3 转发。

攻击产生时交换机 S1 和 S2 处于被攻击状态，控制器启用洗牌算法调度其它代理交换机为用户 1, 2, 3, 4, 5 提供服务，进入第 1 轮洗牌，用户 1, 3, 5 由 S4 转发，用户 2, 4 由 S5 负责转发。此时由于 S5 连接用户 2, 4，并且 S5 未处于被攻击状态，因此可以判定用户 2, 4 不是攻击者，S4 仍然处于被攻击状态，因此用户 1, 3, 5 均有可能是攻击者。经过下一轮的洗牌，用户 2, 5 即可被识别处理。从中可以看出在洗牌过程中，代理层交换机存在 4 种状态：Normal(N), Attacked(A), Suspected(Su), Saved (Sa)。正常状态下，当攻击产生时，状态迁移到

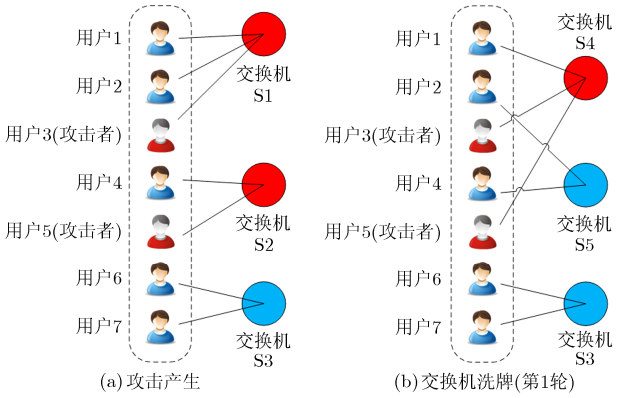


图 3 交换机洗牌示例图

Suspected，通过洗牌后，得到两种状态：Saved 和 Attacked，其中 Saved 状态经过路由重定向后进入 Normal 状态，Normal 状态的交换机负责数据包的转发直至完成所有转发任务，Attacked 状态的交换机将待转发的数据包丢弃，完成对 DDoS 攻击数据包的处理。

3 交换机洗牌模型

3.1 问题定义

为保证每轮洗牌中更快速地分离出攻击者和合法用户，需要实现一个高效的洗牌算法，本节对算法描述中需要的符号和问题进行定义，如表 1 所示。

表 1 符号描述

符号	说明
N_{sum}	当前 SDN 中的所有用户数
N_A	攻击者总数
N_{su}	攻击发生时处于可疑状态的用户数
N_{sa}	一轮洗牌后，被筛选为合法状态用户的用户数
N_{us}	一轮洗牌后仍为可疑状态的用户数
$E(N_{sa})$	目标函数，表示每轮洗牌中被识别为合法用户的数目的期望值
S	代理层交换机总数
S_j	交换机 j 负责转发的用户数
p_j	交换机 j 不被攻击的概率值
$\boldsymbol{A}$	表示 User-to-Switch 的分配情况

通过上述符号描述，可得如下两个等式：$N_{\text{sum}} = N_A + N_{su}$ 和 $N_{su} = N_{sa} + N_{us}$。对于目标函数为 $E(N_{sa})$，能使 $E(N_{sa})$ 值最大，并且时间复杂度较低的算法即为所求。同时，可得等式(1)：

$$p_j = C_{N_{\text{sum}}}^{N_A - S_j} \Big/ C_{N_{\text{sum}}}^{N_A} \tag{1}$$

易得

$$E(N_{sa}) = \sum_j^S p_j S_j = \sum_j^S \frac{C_{N_{\text{sum}}}^{N_A - S_j}}{C_{N_{\text{sum}}}^{N_A}} S_j \tag{2}$$

根据式(2)可知，增大 S 的值会使得 $E(N_{sa})$ 值也增大，但是如果 $S \geq N_{\text{sum}}$，则每个交换机仅负责转发一个用户的数据包，交换机与用户一一对应，此时 DDoS 攻击者的定位是显而易见的。但是上述情况在当前 IP 网络和 SDN 中都是不现实的，并且通常 $S << N_{\text{sum}}$。因此上述求 $E(N_{sa})$ 的问题可以归纳为如下最优化算法问题：

给定 S，向量 $\boldsymbol{A}$ 表示 User-to-Switch 的分配情况，如 $A_1 = 2$ 表示交换机 S_1 负责转发两个用户的数据包，简称 S_1 连接两个用户。公式(2)可以转化为求解满足等式 $\sum_{j=1}^S A_j = N$ 的 User-to-Switch 分配最优化问题。

3.2 攻击者数目似然估计

由上述描述可知，攻击者的数目是个变量，会在计算中会影响 $E(N_{sa})$，因此在实现贪心算法之前需要对攻击者的数目进行预估计。本文使用最大似然估计对攻击者的数目进行预估计。假设所有代理层交换机中未被攻击的交换机数目为 X，在攻击产生时，$X = m$，在一次具体的攻击中，X 的值是可知的(通过交换机中部署的流量检测器可以得知该交换机是否被攻击)。根据文献[9]中的模型可知：

$$\begin{aligned} p(X = m) = {} & p(X \geq m) - C_{m+1}^m p(X \geq (m+1)) \\ & + C_{m+2}^m p(X \geq (m+2)) - \cdots \\ & + (-1)^{S-m} C_S^m p(X \geq S) \end{aligned} \tag{3}$$

使用集合 $U = \{u_1, u_2, \cdots, u_m\}$ 表示未被攻击的代理层交换机，则 $\sum_{j=1}^m S_j$ 表示所有未被攻击的代理层交换机的总和，$N_{\text{sum}} - \sum_{j=1}^m S_j$ 表示攻击者的数目，可以得到等式(4)：

$$p(X \geq m) = \sum_U^m C_{N_{\text{sum}} - \sum_{j=1}^m S_j}^{N_A} \Big/ C_{N_{\text{sum}}}^{N_A} \tag{4}$$

根据式(3)和式(4)即可推导出 N_A 的值。根据推导出的 N_A 即可使用贪心算法进行交换机洗牌。

3.3 交换机贪心洗牌算法

当 $N_A \leq S$ (攻击者的数目小于等于 proxy switch)时，平均分配的方法是最优的，但是在 DDoS 攻击下，这种情况几乎不存在，因此本文仅研究 $N_A >> S$ (攻击者数目远大于代理层交换机数目)的情况，此时平均分配的方法便不再是最优的，但是可以通过贪心算法得到近似最优值。

显然，当代理层每个交换机连接较少的用户时，代理层交换机被攻击的概率会降低，但是此时每轮洗牌后，被 Saved 的用户数目也会降低，因此需要选择一种提高每轮被 Saved 用户的算法，即满足

$$E(S_j) = \left(C_{N_{\text{sum}} - S_j}^{N_A} \Big/ C_{N_{\text{sum}}}^{N_A} \right) S_j \tag{5}$$

算法 1 描述了使用贪心算法实现的交换机洗牌过程。算法在以下 3 种情况下退出：(1)代理层交换机比用户多，每个用户可以分配一个代理层的交换机，二者之间可以实现一一对应，无需洗牌即可实现定位；(2)仅剩一个代理层交换机，剩下所有的用户均被分到该代理层交换机；(3)攻击者的值为 0，此时无需进行洗牌。

表 2 的 GreedyShuffle 算法是一个递归算法，调用 MS()函数得到满足式(5)的分配方案，k 表示一个代理层交换机连接的用户数目。pA 表示按照 k 值进

表 2　GreedyShuffle 算法

算法 1　GreedyShuffle() //计算 user-to-proxy switch 的分配策略
输入　user, attacker, pS(代理交换机数目)
输出　numShuffle
def GreedyShuffle(user, attacker, pS)
if user<=pS **then**
　　Assign 1 exclusive proxy switch to end user;
else if proxySwitch == 1 **then**
　　Assign all users to the proxy switch;
else if attacker == 0 **then**
　　Evenly distribute user over proxy switch;
else
　　k = MS(user, 0, user −1, attacker);
　　pA = ⌊user / k⌋;
　　if pA >= pS **then**
　　pAssign = pS − 1;
　　pRem = pSwitch − pA;
　　uR = user − pA*k;
　　aR = Shuffle(attacker*uR/user);
　　GreedyShuffle(uR, aR, pR);
def MS(user, left, right, attacker)
max=0;
mS=0;//
for i = left → right **do**
　　saved= $C_{\text{user}-i}^{\text{attacker}} / C_{\text{user}}^{\text{attacker}}$;
　　if saved > max **then**
　　max=saved;
　　mS=i;
return mS;

行分配时需要的代理层交换机的数目。pR, uR, aR 均表示剩余的代理层交换机数目。算法 1 是一个递归算法，并且容易得到其时间复杂度 $\Theta(N_{\text{sum}} \cdot N_A)$，当 N_{sum} 和 N_A 较大时，复杂度太高，计算量太大。对此，本文使用 Eger[10]提出的 Stirling 近似等式 $n! \approx (n/e)^n \sqrt{2\pi n}$，化简后可得 $E(S_j) = N_{\text{sum}}/(N_A \cdot e)$。因此算法 1 的 MS 函数中的循环可以省去，该函数复杂度将为 1，算法 1 的时间复杂度降低为 $\Theta(N_A)$。

4　仿真测试

本节对上述基于 OpenFlow 交换机洗牌算法的性能以及使用该方法进行 DDoS 防御的能力进行测试。前者使用 MATLAB 对算法进行模拟，在算法中模拟了一定数目的用户、攻击者、代理层交换机，为方便比较，在模拟中将上述变量设置为固定值；后者在开源 SDN 控制器 Ryu 和开源 OpenFlow 交换机 Open vSwitch 上实现 DDoS 动态防御模块，作为 Ryu 控制器的一个模块加载启动。

4.1 算法性能测试

在模拟中用户和攻击者的选取根据 Matsumoto[11]提出的 Mersenne-Twiste 模型随机产生。攻击者攻击所有与之相连接的代理层交换机，并且假设攻击者具有足够的带宽和资源来进行攻击。

图 4 描述了洗牌轮数、攻击者数目、代理交换机数目三者之间的关系。其中实线代表的是使用算法 1 隔离 80%和 95%的攻击者时，洗牌轮数、攻击者数目，以及代理交换机数目三者之间的变化关系；虚线代表的是隔离 80%和 95%的攻击者时的理论上界(根据式(5)计算得到)。与预期结果相似，当隔离攻击者的百分比恒定时，攻击者数目越多，需要洗牌的轮数也越多，代理交换机的数目也越多。图 4(a)和图 4(b)中，用户数目和代理交换机的值是固定的，攻击者的数目从 10 递增至 500。图 4(c)和图 4(d)中用户数目与图 4(a)和图 4(b)相同，攻击者的数目固定为 100，代理交换机的数据从 40 递增至 500。每轮模拟执行 10 次，取其均值。根据图 4 可知，算法 1 的执行结果接近理论上界，可以认为是近似最优的。从图 4(a)和图 4(b)也可以看出当隔离攻击者的比例确定时，需要洗牌的轮数与攻击者数目之间呈现近似线性增长的关系。图 4(c)和图 4(d)则说明当代理层交换机数目增加时，需要洗牌的轮数会减少，并且当代理交换机数目小于攻击者数目时，变化极为明显，而当代理交换机数目大于攻击者数目时，变化趋向平缓。

4.2 DDoS 动态防御能力测试

本文对基于交换机洗牌的 DDoS 动态防御方法进行了原型系统实现，使用 Python 语言开发，作为开源控制器 Ryu v3.6 的应用软件随 Ryu 控制器同时启动。使用的 OpenFlow 协议版本为 1.1.0，代理交换机使用虚拟交换机 Open vSwitch v2.3.1(拓扑图中简称 OVS)完成。原型系统实现共计约 3000 行 Python 代码。

使用图 5 所示的拓扑结构进行测试，其中 OVS-1 至 OVS-5 表示代理层交换机，其余表示转发层交换机；使用 Ryu 对交换机进行管理和配置，Ryu 安装在 Intel Xeon CPU、主频 2.13 GHz 和 64 GB 内存的物理主机上，并在 Ryu 上部署流量监测程序定时获取网络中的流量分布情况；使用 5 个 Kali 虚拟机模拟 3 种类型的流量，其中 Agg1 至 Agg4 表示合法流量，Agg5 表示攻击流量，攻击流使用 TCP SYN 方式实现。

(1)抗 DDoS 攻击能力：　图 6 描述了不使用本文方法(图 6(a))和使用本文方法(图 6(b))时流量的

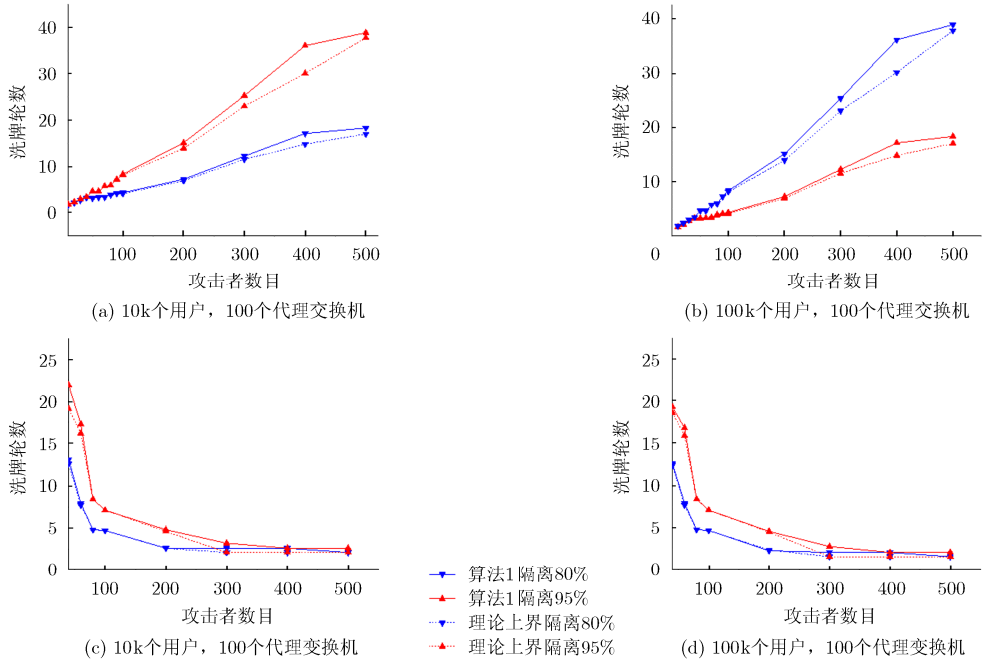


图 4 攻击者、洗牌轮数和代理交换机之间的变化关系图

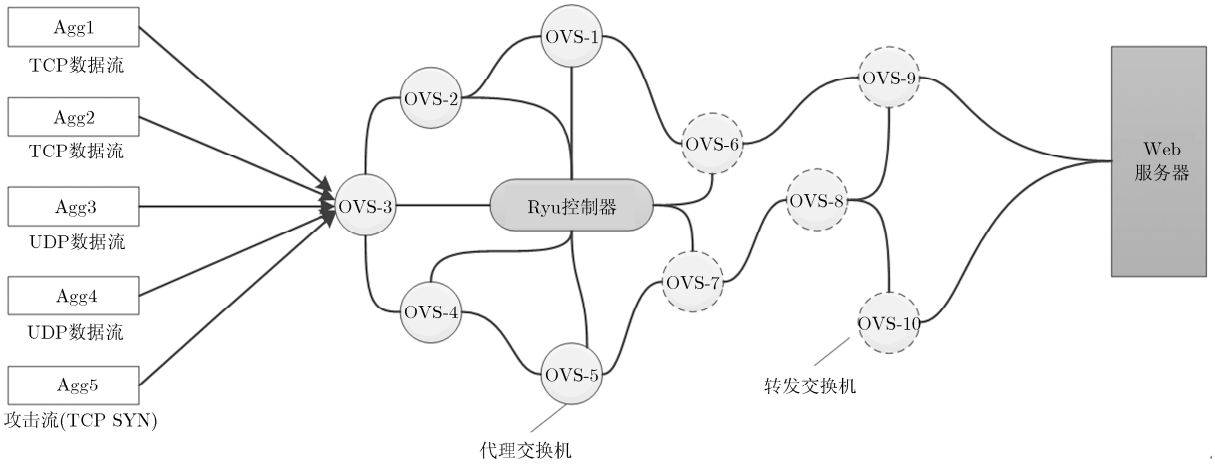


图 5 能力测试拓扑结构图

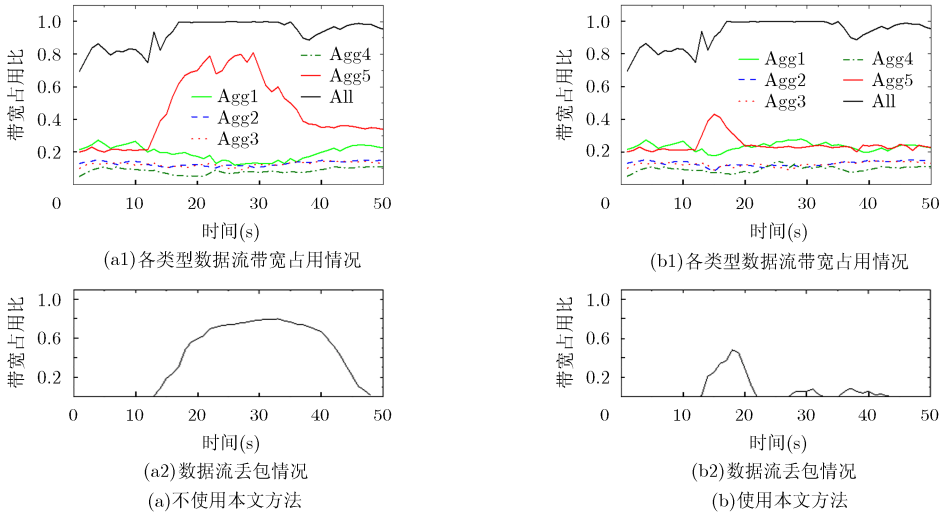


图 6 动态防御方法使用前、后的网络流量分布图

带宽占用率和数据包丢失率的分布情况。数据流发送速率从 $t=13$ 时开始增加，到 $t=30$ 时开始下降。从图 6(a1)可以看出，在不使用本文方法时，高带宽的数据流会耗尽几乎所有的网络带宽。图 6(a2)代表了正常通信的数据包的丢失率，当 Agg5 的速率增大时，正常网络数据包的丢失率也开始增大。图 6(b1)和图 6(b2)展示了在使用本文方法后的模拟结果。当 Agg5 的速率开始增大时，丢包率也开始增大，但是当丢包率增大到设定的阈值 10%时，本文方法检测到 Agg5 的异常，启动交换机洗牌方法对 Agg5 进行流量控制重定向，筛选攻击者，转发正常网络数据包。从图 6(b1)和(b2)可以看出，部署本文方法后，攻击流量的持续时间显著变短，对带宽的占用率和丢包率均减小。

图 7(a)所示是在上述测试拓扑结构下，攻击流大小与动态防御方法恢复网络正常所需时间的关系。测试中使用多个攻击者构造不同大小的攻击流对拓扑结构中的代理交换机进行攻击。从中可以看出二者近似呈现对数关系，当攻击者被隔离后，攻击者实施攻击时产生攻击流的大小对防御的影响不大，根据图 7(a)可推断在 1200 s 时，攻击者几乎被全部隔离出，后续尽管攻击流继续增大，但对网络几乎不造成影响。

(2)与已有工具的对比测试：　与传统网络中 DDoS 防御工具 Ice-Shield 防火墙(试用版)、开源防御工具 DDoS Deflate 进行对比测试。其中防火墙和开源防御工具部署在 Server 中，采用图 5 所示拓扑进行测试。测试结果如图 7(b)所示，横坐标是时间轴，纵坐标表示网络中总流量的变化，分别在 20 s 和 200 s 时启动攻击脚本。从图 7(b)中可得到：(1)在 80 s 时，黑线即开始下降，表示本文方法的攻击流筛选能力相较其他两种工具较强；(2)在 200~250 s 之间，其余两条曲线的最低点比黑线低，并且低于 0 s 时刻的值，说明另外两种工具存在一定的误报，且比本文方法要高；(3)在 200 s 后，黑线基本保持不变，说明攻击者隔离后，再次启动攻击，对网络不会造成影响，而其它两个工具由于无法隔离攻击者，在攻击再次发生时，无法产生该防御效果。

(3)对网络负载的影响：　在控制器和交换机上实现该动态防御方法会增加数据包转发的时延，下面对其时延开销进行测试。通过测试数据包由用户发送到 Web 服务器的往返时间测试系统的负载开销。选择 10 个地理位置相距较远的用户节点组成 5 对测试对象。RTT 根据 5 对节点的 TCP 消息往返时间确定，取 100 次结果的均值。吞吐率测试中的数据由 10 个 Iperf 会话组成。从表 3 和表 4 可以看出，引入采用本文方法后，节点之间的通信时延通常低于 30%，而对吞吐率的影响较大。吞吐率的变化主要是由于控制器动态更新交换机流表所致，而不是算法本身的问题，建立控制器与代理层交换机之间的专用通道可提高吞吐率。

5　结束语

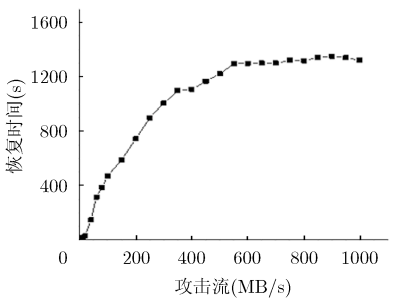
DDoS 攻击始终是网络安全中悬而未决的难

表 3 代理交换机的引入对网络负载的影响

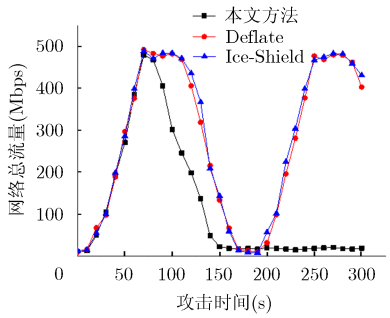
测试对象	直接连接	间接连接			
	往返时延(ms)	平均往返时延(ms)	负载增加(%)	最大往返时延(ms)	负载增加(%)
1	65	105	61.54	143	121.30
2	88	102	15.91	129	46.59
3	86	105	22.09	132	53.49
4	92	113	22.83	131	42.39
5	85	108	27.06	119	42.50

表 4 代理交换机的引入对吞吐率的影响(MB/s)

	测试对象				
	1	2	3	4	5
直接连接	89.57	91.61	86.39	124.52	122.36
间接连接	17.36	16.90	13.82	16.11	14.87



(a)攻击流与网络恢复时间的关系



(b)攻击时间与网络总流量的关系

图 7 压力测试及对比测试的结果图

题，许多网站运行商选择内容分发网络(CDN)，或者部署在云环境中来缓解 DDoS 攻击。但是这种防御方法是攻防双方资源的对抗，不能有效解决 DDoS 攻击问题。本文提出的 DDoS 动态防御方法利用 SDN 集中控制和动态管理的特性，通过交换机洗牌的方法，不仅可以有效防御 DDoS 攻击，同时能够对攻击者进行筛选定位。但是随着 DDoS 攻击技术的发展，动态防御的方法也要随之改进，未来可以考虑将利用 SDN 实现的动态防御方法与云环境配合使用，实现多维立体的攻击防护。

参 考 文 献

[1] PRAS A, SANTANNA J, and STEINBERGER J. DDoS 3.0-How Terrorists Bring Down the Internet[M]. New York: Springer, 2016: 1–4. doi: 10.1007/978-3-319-31559-1_1.

[2] YADAV V K, TRIVEDI C, and MEHTRE M. DDA: an approach to handle DDoS (Ping Flood) attack[C]. International Conference on ICT for Sustainable Development, Singapore, 2016: 11–23. doi: rg/10.1007/978-981-10-0129-1_2.

[3] NAGPAL B, SHARMA P, and CHAUHAN N. DDoS tools: classification, analysis and comparison[C]. IEEE International Conference on Computing for Sustainable Global Development, New Delhi, India, 2015: 342–346.

[4] LIU Xia, YANG Xin, and XIA Yu. Netfence: preventing internet denial of service from inside out[C]. ACM Sigcomm Computer Communication Review, New York, NY, USA, 2010: 255–266. doi: 10.1145/1851182.1851214.

[5] BRAGA R, MOTA E, and PASSITO A. Lightweight DDoS flooding attack detection using NOX/OpenFlow[C]. International Conference on Local Computer Networks, Washington, DC, USA, 2010: 408–415. doi: 10.1109/lcn.2010.5735752.

[6] YEGANEH S and CANJALI Y. Kandoo: a framework for efficient and scalable offloading of control applications[C]. ACM Workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, 2012: 19–24. doi: 10.1145/ 2342441.2342446.

[7] SHIN S and PORRAS P. AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks[C]. International Conference on Computer and Communications Security. Berlin, Germany, 2013: 413–424. doi: 10.1145 /2508859.2516684.

[8] LIM S, HA J, KIM H, *et al*. A SDN-oriented DDoS blocking scheme for botnet-based attacks[C]. International Conference on Ubiquitous and Future Networks, Shanghai, China, 2014: 63–68. doi: 10.1109/icufn.2014.6876752.

[9] JOHNSON N and KOTZ S. Urn models and their applications: an approach to modern discrete probability theory[J]. *Journal of International Statistical Review*, 1978, 20(4): 104–119. doi: 10.2307/3617688.

[10] EGER S. Stirling's approximation for central extended binomial coefficients[J]. *Journal of American Mathematica*, 2014, 121(4): 344–349. doi: 10.4169/amer.math.monthly.121.04.344.

[11] MATSUMOTO M and NISHIMURA T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator[J], *Journal of Model*, 1998, 8(1): 3–30. doi: 10.1145/272991.272995.

武泽慧： 男，1988 年生，博士，研究方向为网络安全.

魏 强： 男，1979 年生，副教授，硕士生导师，研究方向为工控网络安全.

任开磊： 男，1992 年生，硕士，研究方向为 SDN 安全.

王清贤： 男，1960 年生，教授，博士生导师，研究方向为网络与信息安全.