UDM：基于 NFV 的防止 DDoS 攻击 SDN 控制器的机制

钱红燕，薛昊，陈鸣
（南京航空航天大学计算机科学与技术学院，江苏 南京 211106）

摘 要：广泛存在的分布式拒绝服务（DDoS）攻击对于软件定义网络（SDN）的控制器形成了致命威胁，至今还没有一种安全机制能够防御。将 SDN 和网络功能虚拟化（NFV）结合，提出了一种新颖的防范 DDoS 攻击 SDN 控制器的前置检测中间盒(UDM)机制，在 SDN 交换机端口与用户主机之间分布式部署 UDM 以检测并拒止 DDoS 攻击报文。此外，还提出了一种基于 NFV 的前置中间盒的实现方法，使这种 UDM 机制更为经济和有效，实现了基于该机制的原型系统，并对其进行大量测试。实验结果表明，基于 NFV 的 UDM 机制能够实时有效地检测和防止对控制器的 DDoS 攻击。

关键词：DDoS 攻击；控制器安全；软件定义网络与网络功能虚拟化；前置检测中间盒
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2019067

UDM: NFV-based prevention mechanism against DDoS attack on SDN controller

QIAN Hongyan, XUE Hao, CHEN Ming
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Abstract: DDoS attack extensively existed have been mortal threats for the software-defined networking (SDN) controllers and there is no any security mechanism which can prevent them yet. Combining SDN and network function virtualization (NFV), a novel preventing mechanism against DDoS attacks on SDN controller called upfront detection middlebox (UDM) was proposed. The upfront detection middlebox was deployed between SDN switch interfaces and user hosts distributed, and DDoS attack packets were detected and denied. An NFV-based method of implementing the upfront middlebox was put forward, which made the UDM mechanism be economical and effective. A prototype system based on this mechanism was implemented and lots experiments were tested. The experimental results show that the UDM mechanism based on NFV can real-time and effectively detect and prevent against DDoS attacks on SDN controllers.

Key words: DDoS attack, controller security, SDN and NFV, upfront detection middlebox

1 引言

软件定义网络（SDN, software defined networking）[1]是一种将控制和转发功能相分离的网络体系结构，它的集中控制方式使网络管理与应用业务配置更加灵活和便捷，也便于部署网络新技术、新协议，促进了网络创新。然而，在 SDN 获得了空前发展的同时，SDN 的安全性也引起了越来越多的关注。特别是 SDN 控制器作为 SDN 的核心，保障其安全性成为科研人员研究的焦点之一。这是因为 SDN 的运转完全依赖控制器执行集中式的决策，如果控制器因为某种原因而无法正常工作甚至瘫痪，整个网络都将随之崩溃。

过去几十年，分布式拒绝服务（DDoS,

收稿日期：2018−06−13；**修回日期**：2019−01−03
通信作者：陈鸣，mingchennj@163.com
基金项目：国家自然科学基金资助项目（No.61772271, No.61379149）
Foundation Item: The National Natural Science Foundation of China (No.61772271, No.61379149)

distributed denial of service）攻击一直是互联网中的一种重大的安全威胁，学术界和工业界的研究人员对此进行了大量研究工作。由于 SDN 集中控制的特点，DDoS 针对 SDN 的攻击危害性也更大。根据 SDN 工作原理，当某新流到达某 SDN，进入一个 OpenFlow 交换机时，该 OpenFlow 交换机没有与该流相匹配的流表项，此时交换机就会将该报文封装成 Packet_In 报文转发至 SDN 控制器，由控制器处理，而交换机与控制器之间的链路常常会成为 DDoS 攻击的目标。控制器收到交换机传来的 Packet_In 报文后就会处理这些报文，当报文数量过多时，会大量消耗控制器的资源，降低网络性能，因此控制器资源的瓶颈也成为 DDoS 攻击的目标。控制器处理完分组请求后会下发流表到相应的 OpenFlow 交换机，而交换机所能保存的流表项数量有限，当下发的流表数量大于交换机流表容量时，流表就会溢出，这一点也成为 DDoS 攻击的目标。由于 SDN 受到攻击后会对整个网络造成巨大影响，特别是 SDN 控制器受到攻击之后，很可能会使整个网络瘫痪。因此，设计一种针对 SDN 控制器的防御机制来处理 DDoS 攻击是研究者们追求的目标。

目前，解决 SDN 中 DDoS 攻击的方法大体包括以下 2 种：一种是为 SDN 中的交换机和控制器增加更多的缓存队列，使遇到 DDoS 时可以有更多的资源来应对，然而这种方法只是在遇到 DDoS 攻击时起缓解作用，并不能从根本上解决 DDoS 攻击，当 DDoS 攻击流量过大时，依然会耗尽所有的缓存资源；另一种是在 SDN 的控制平面建立完整的 DDoS 检测和防御机制，通过 DDoS 检测方法来判断攻击的类型，然后报警或采用相应的方式进行 DDoS 防御，但是 DDoS 攻击通常是在很短时间内进行的很大流量的攻击，这种方法需要定时收集流量数据，并对其进行分析之后再进行防御，实时性不高，往往分析完攻击类型进行防御时，整个网络已经受到影响甚至瘫痪。

近年来，随着网络功能虚拟化（NFV, network function virtualization）[2]的发展，其具有的优点如不依赖底层硬件、极高的可扩展性、部署和迁移的灵活性以及设备的低成本性都为研究人员所关注。面对 DDoS 对 SDN 的攻击，本文希望可以借助 NFV 多方面的优势来解决 2 个问题：1) 如何创新设计一种实时性较高的 DDoS 检测防御控制机制；2) 如何提高这种机制的灵活性、实用性。

本文的主要贡献如下：通过分析 SDN 控制器的工作过程，提出了一种新型分布式防范 SDN 中 DDoS 攻击的机制；根据 NFV 和 SDN 的特征，提出了一种基于 NFV 实现上述机制的技术，并通过 SDN 和 NFV 结合的技术给出了防范 DDoS 攻击的方案；原型系统的实验表明了该机制和技术的有效性。

2　相关工作

在 SDN 中，控制器集中式控制决策方式需要分析所有新流并为其安装流表。当受到 DDoS 攻击时，控制器可能会由于负载过大而处于低效或者瘫痪状态。文献[3]提出，当网络中的流量大于 10 Gbit/s 时，SDN 控制器就很难对网络中产生的大量新流进行合理和及时的处理。文献[4]提出，针对控制器资源的各种攻击行为需要提高可扩展性，避免遇到攻击时对 SDN 的危害更为严重。

为了减少 DDoS 攻击对控制器性能的影响，文献[5]提出了多层公平队列（MLFQ, multi-layer fair queueing），在控制器端为每个交换机创建多层次的缓冲队列。MLFQ 实施 SDN 控制器资源与多层队列的公平共享，可根据控制器负载动态扩展和聚合，是一种简单而有效的 DDoS 缓解方法。文献[6-7]针对控制器资源饱和威胁提出了解决方案，但是需要对交换机进行修改。文献[8]提出了一种主动防御 DDoS 攻击的体系结构安全覆盖服务，该机制能够有效缓解攻击的影响。

近年来，NFV 和相关技术在很多新兴应用领域都展现出独特的魅力和良好的应用前景[2]。NFV 将网络功能的上层软件与底层硬件解耦，并支持灵活动态的配置、启动网络功能及高效合理的分配基础设施的资源，降低了网络中的设备投资成本和运营成本，提高了网络的灵活性和可扩展性，加快了网络的创新和新技术的部署应用。应用 NFV 可以更好地运用分布式协调方法来缓解 DDoS 攻击。文献[9]提出了一种基于 NFV 和 SDN 的 DDoS 缓解框架，利用中心控制和全局网络视图的 SDN 特性在控制平面对网络流量进行监控分析，检测出异常流量后将触发相应对策计算的动作，防范资源虚拟化、实例化、部署和互联。文献[10-12]提出了通过应用 NFV 来考虑功能灵活性的 DDoS 攻击检测和预防机制。文献[13]提出了一种在 OpenFlow 边缘交换机中运行的基于熵的轻量级 DDoS 洪泛攻击检测模型，

计算目的 IP 地址的熵，如果低于阈值，则检测到攻击，该方案实现了 SDN 中的分布式异常检测，并减少了控制器上的流量采集过载。

综上所述，目前的所有防御机制都仅能缓解 DDoS 攻击的危害，而无法防止 DDoS 对 SDN 控制器的攻击。此外，有些防御机制的实现开销较大，难以实际部署。

3 防范攻击的分布式机制

3.1 防范攻击的难点

为了便于分析，图 1 给出一个典型的 OpenFlow 网络实例。其中，OpenFlow 交换机（S_1、S_2、S_3）相互连通，并与周围的台式主机（pc_1、pc_2、pc_3）、手提电脑（$user_1$、$user_2$）以及服务器相连，它们构成了 SDN 的数据平面；而控制平面的 SDN 控制器与 3 个交换机相连，其中的 SDN 控制器负责处理来自交换机的报文并下发路径流表，使 SDN 中的各个主机之间可以相互通信。

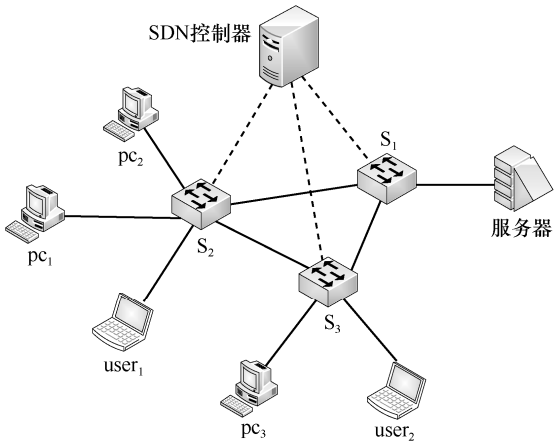


图 1 一个典型的 OpenFlow 网络实例

图 2 给出了在上述 SDN 中处理数据流时流表下发工作过程的主要步骤：1) 当用户发送的报文进入 OpenFlow 交换机后，首先会与交换机中的流表进行匹配，若无相应的流表项则会触发 table_miss 事件，交换机将报文封装成 Packet_In 报文后发送给 SDN 控制器，请求控制器处理；2) SDN 控制器根据已知的网络拓扑情况为新的数据流请求计算路径，将流表以 Packet_Out 报文发送至路径上所有的 OpenFlow 交换机上；3) OpenFlow 交换机收到 Packet_Out 报文后为报文安装相应的流表项，流表安装完毕之后，报文即可按照流表在 OpenFlow 交换机之间转发。在步骤 1)中，如果交换机收到发送的 Packet_In 报文数量过大，超过链路的容量，会导致正常的分组报文无法顺利到达 SDN 控制器，OpenFlow 交换机无法正常工作，与该交换机相连的主机、服务器等也都会受到影响。在步骤 2)中，如果控制器受到攻击，接收到过多的请求，导致控制器资源消耗过度，控制器瘫痪以至无法处理来自交换机的正常请求，与该控制器相连的所有交换机以及与交换机相连的主机、服务器等都会受到影响，整个 SDN 都无法正常工作。在步骤 3)中，如果控制器下发的流表项过多，超出 OpenFlow 交换机中流表的容量，会导致一些流表项被丢弃，致使一些数据流需要重新传入 SDN 控制器处理，增加控制器的负担，影响 SDN 控制器以及与交换机相连的主机、服务器等的正常工作。

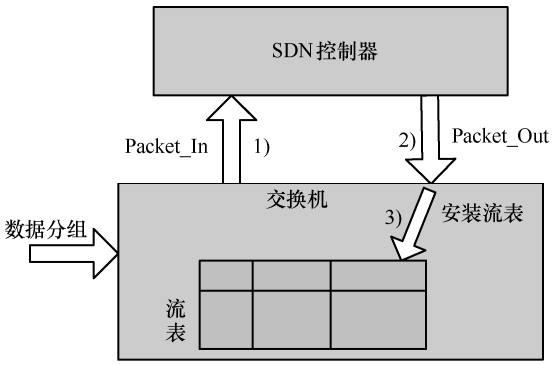


图 2 流表下发工作过程

3.2 防范攻击的优势位置

本文将检测和防御 DDoS 攻击的相关设备称为 DDoS 检测中间盒，从前面分析可知，DDoS 攻击可能会攻击 SDN 中的各个位置，而且无论攻击什么位置都会对 SDN 产生危害。DDoS 检测中间盒在 SDN 中部署的位置也会对防御的效果产生影响。目前，比较常见的检测和防御位置有 3 种。第一种是将检测与防御放置在控制平面。由于 SDN 集中式管理的特点，在控制平面进行检测和防御可以一次性监控网络中的所有流量，但是随之而来的问题是会导致控制平面的数据流过大，在受到攻击时，报文会大量进入控制平面，对交换机控制器之间的链路以及控制器造成影响。第二种是将检测程序放在控制平面，而将防御程序放在数据平面。将防御程序放在数据平面确实会减轻很多控制平面的负载，在检测到 DDoS 攻击之后，DDoS 的后续攻击流都不会对 SDN 造成影响，不过由于 DDoS 快而多的特点，往往在

DDoS 攻击被检测到的时候，网络已经受到很大影响甚至瘫痪。第三种是将检测中间盒部署在数据平面。目前常见的方案是通过修改 OpenFlow 交换机的代码，将检测中间盒部署在 OpenFlow 交换机之中；另一种方案是在数据传输路径中增加检测中间盒，通过分析进入流的分组，检测 DDoS 攻击并进行处理。前一种方案要修改交换机硬件和软件，使交换机不再符合标准规范；后一种方案检测和防御的效果好，但是实现成本有可能很高。

综上所述，将检测和防御功能放在控制平面的优缺点是：控制平面监控数据更全面，软件实现更加方便，成本低，但是集中式处理大流量会使控制器处理能力成为瓶颈，难以应对 DDoS 攻击。将检测和防御功能分别放在控制平面和数据平面的优缺点是：SDN 控制器的负载较低，但是处理 DDoS 攻击的实时性不好。而将检测和防御功能放在数据平面的优缺点是：利用数据平面传输能力强、处理时效性好的特点，以分布式方式分散 DDoS 攻击的影响，避免使控制器成为性能瓶颈，但是成本相对较高。为避免高成本使机制难以实施的缺点，需要找到降低实现防御机制成本的方法，详情参见第 4 节的讨论。

3.3　熵值计算

文献[14]提出的基于熵值的检测方法可以较好地检测出是否存在 DDoS 攻击报文。由于检测实时性和准确性的要求，在将上述熵值检测方法引入本文检测机制的过程中，对其进行调整，通过滑动窗口调整数据集大小，使该方法可以更好地适用于本文的分布式实时检测机制。

信息熵能够很好地表现出数据集的特征，熵值越高，数据集的随机性越大；熵值越低，数据集的随机性越小。Shannon 熵定义如下。

数据集 X 中共有 N 个数据，包含 n 个不同的取值，即 $\{x_1,x_2,\cdots,x_n\}$，各个值出现的概率为 p，即 $\{p_1,p_2,\cdots,p_n\}$，其中，p_i 可由式(1)计算得到。

$$p_i=\frac{x_i}{N} \tag{1}$$

那么数据集 X 的熵为

$$H(X)=-\sum_{i=1}^{n}p_i\log p_i \tag{2}$$

当数据集中所有元素相同时，熵值为 0；而当数据集中所有元素都不相同时，熵值为最大。在所有情况下，数据集的熵值都在$[0,H_{\max}]$区间内。为了使熵值不受数据集大小 N 的影响，通过式(3)将数据集的熵值标准化到[0,1]区间内。

$$H_N(X)=\frac{-\sum_{i=1}^{n}p_i\log p_i}{\log N} \tag{3}$$

通常情况下，网络中普通流量的源 IP 地址在一个连续采集的数据集中相对稳定，但是当遇到伪造地址的 DDoS 攻击时，源 IP 地址的随机性会增加，熵值会显著变大。因此在得到网络中正常流量的源 IP 地址熵值区间之后，即可根据该熵值区间来设置本文的 DDoS 检测阈值。

数据集大小（即滑动窗口大小）N 是一个可调性很高的参数。DDoS 攻击报文在网络中的速率为 v_a，采集到攻击报文的时长为 t_a，而正常报文在网络中的速率为 v_n，采集到正常报文的时长为 t_n。DDoS 攻击报文在数据集中占的比例 P_a 为

$$P_a=\frac{v_at_a}{v_at_a+v_nt_n} \tag{4}$$

由于 DDoS 攻击突发性的特点，滑动窗口采集数据集的时间不宜过长。由式(4)可知，若采集时间过长，可能会导致 $t_n>>t_a$，DDoS 攻击报文在数据集中的占比会减小，导致熵值检测的准确性降低。在熵值检测中，数据集中的各个值出现的概率 p_i 至关重要，由于滑动窗口采集数据集的不稳定性，不同数值在不同数据集中出现的次数会有一定的偏差，由式(1)可知，N 越大，计算得到的 p_i 偏差越小，最后得到的熵值越逼近于该时间段内中间盒转发的报文的熵值。综上所述，在保证合理的数据采集时间的前提下，进行 DDoS 熵值检测算法的数据集越大，检测的准确性越高。

然而 N 增大同样带来了一个问题，先进入样本集的报文需要等待后续的报文进入样本集后才能进行分析处理，这样会增加报文在网络中的传播时延。若网络中 t 时间内发送报文的总数量为 S，检测算法中设置的数据集容量为 N，那么样本集中第 i 个报文因等待其他报文进入数据集额外增加的传播时延为

$$T_i(N)=(N-i)\frac{t}{S} \tag{5}$$

由式(5)可知，检测算法对第一个进入样本集的

报文影响最大，额外增加传播时延 $T_{\max}$；而最后一个进入样本集的报文额外增加的传播时延为 0，只增加了计算熵值的处理时延。所有报文额外增加的传播时延都在[0,$T_{\max}$]区间内。平均增加的传播时延 $T_{\text{avg}} \approx \frac{T_{\max}}{2}$。因此，在保证 $T_{\max}$ 和 T_{avg} 满足网络系统对传播时延要求的情况下，可以适当提高数据集大小 N 来提高检测算法的准确性。

3.4 前置检测中间盒机制

前置检测中间盒（UDM, upfront detection middlebox）机制定义为：1)检测中间盒靠前部署在用户设备与相连的 OpenFlow 交换机某端口之间，而不是放在交换机甚至控制器之前；2)检测中间盒具有检测 DDoS 攻击流的功能；3)检测中间盒对检测到的 DDoS 攻击流具有拒止的功能。

SDN 所有的用户流量直接进入前置的检测中间盒，中间盒的检测算法分析是否存在 DDoS 攻击。如果存在 DDoS 攻击流则丢弃相关报文，交换机乃至控制器就不可能收到这些攻击报文。如果是正常的流量则将其转发给交换机进行后续处理。为了降低处理成本，可以区分用户的可信度，将可信度高的用户的流量直接转交给交换机处理，而只处理可信度较低的那部分用户的流量。使用传统方法采用专用硬件和软件来设计实现检测中间盒，将存在严重的性价比问题，部署系统需要大量的检测中间盒将提高系统的造价，但如果检测中间盒的成本变得很低时，系统的性价比将不再是主要问题。本文将在第 4 节研究解决该问题的具体技术方案。

前置检测中间盒的流处理流程如算法 1 所示。该算法首先需要为 Packet_View 设置一个具有一定容量的缓存区，本文将该缓存区称为滑动窗口。对于持续进入中间盒的流分组，先短暂缓存在 Packet_View 中，依次提取分组的源 IP 地址信息。每当 Packet_View 中存储的信息量达到一定数量时，就开始计算这些报文的目的地址的熵值 H。若该熵值在正常范围以内，则判断这些报文不存在 DDoS 攻击，将缓存 Packet_View 中所有的报文转发到交换机；若该熵值异常，则判断其中具有大量的 DDoS 攻击流分组，向系统报警并将缓存中的所有报文丢弃，这些 DDoS 攻击报文就根本没有机会进入 SDN 中，也不可能形成对控制器的攻击。与此同时，系统将在一定时间内封闭与该用户连接的端口，要求该用户或网络管理员解决 DDoS 攻击来源问题。尽管这种处理策略可能会对正常应用流产生影响，但由于每个检测中间盒只面对一个用户，因此对系统中的其他用户不会产生任何影响，并且要求用户在正常使用网络之前，先解决自身存在 DDoS 攻击流的问题也是合理的。

当用户流分组较少时，缓存 Packet_View 中的报文数量在一定时间间隔内仍未填满，为了保证应用流的实时性，此时系统将其判断为无 DDoS 攻击，并将转发缓存中的所有报文。这是因为此时用户到达的报文量很少，它们不会对控制器性能造成大的影响。

算法 1 前置检测中间盒的流处理

输入 流 F 的报文 p，视图容量 N

输出 对流 F 中报文放行或者阻断

1) 设置 Packet_View 容量为 N
2) for 每一条新到的报文 p do；
3) 在报文 p 中提取源 IP 地址的信息 h；
4) 将 h 保存到 Packet_View 中；
5) if （Packet_View 饱和）then
6) 通过式(3)计算熵值 H；
7) if （H 超出正常范围）then
8) 判断为 DDoS 攻击，将 Packet_View 中数据相关的报文丢弃，报警，关闭端口；
9) else
10) 将 Packet_View 中数据相关的报文全部放行，转发；
11) end if
12) end if
13) end for
14) if （无后续报文 && Packet_View 未饱和）then
15) 窗口中报文全部放行，转发；
16) end if

算法 1 中如何使用缓存区还有一些值得研究的问题，例如，缓存区应当设置的大小，检测中间盒对流报文产生的时延的影响，本文将在第 5 节进行深入研究。

4 前置检测中间盒机制及其实现

DDoS 检测中间盒的实现技术是本文研究的另一个重要问题。近年来快速发展的 NFV 技术为本文采用虚拟化技术实现检测中间盒功能提供了一

个很好的思路。一方面，NFV 是一种在标准服务器硬件上以特定软件来实现网络功能的技术，在特定的条件下能够以很低的成本提供所需的各种虚拟网络功能（VNF, virtualized network function），而且 NFV 技术的灵活性和可扩展性能够为实现分布式检测防御提供便利；另一方面，用一个检测中间盒来处理一个用户的流量，VNF 通常是有能力来应对的。因此，本节设计了一种基于 NFV 与 SDN 综合的技术，以分布式方式部署 DDoS 检测中间盒的方案。首先，设计了一种基于 NFV 与 SDN 综合的分布式网络环境；然后，设计实现了一种检测和防御 DDoS 攻击的中间盒。

4.1　NFV/SDN 环境

在 NFV 与 SDN 综合的环境中，存在下列情况：一种是基于专用设备的 SDN 与基于虚拟化的 NFV 的综合，主要是虚拟设备和真实设备的交互互联，将虚拟化的 NFV 作为专用设备 SDN 中的一部分；另一种是基于虚拟化的 NFV 和 SDN 之间的综合，主要是将 SDN 的专用设备也用虚拟设备替换使用。其实两者从逻辑交互方式上看并无太多不同，并且现有的虚拟化技术都具有虚拟设备与实体设备交互的能力。为简化讨论，本文仅讨论在虚拟化环境下的情况。

首先，选用硬件适宜的服务器作为承载大型虚拟计算环境的宿主机。在其上运行 Linux 操作系统以提供虚拟化计算环境，提供构建虚拟网络功能和虚拟 SDN 环境的硬件条件。

其次，生成配置网络虚拟机。优选高效低耗的虚拟机如 Linux 容器（LXC[15]），通过为虚拟机配置 IP 地址、路由选择协议、性能参数等，使其成为某种网络功能的虚拟网络设备，如虚拟路由器、虚拟主机等。

第三，生成配置网络中间盒。在上述虚拟网络设备上安装特定网络功能软件，使其成为具有特定网络功能的虚拟网络中间盒，例如防火墙、入侵检测系统或本文所需 DDoS 检测中间盒等。

第四，部署 SDN 互联网络虚拟设备和网络中间盒。根据应用需求，通过在虚拟计算环境部署虚拟 OpenFlow 交换机，将所有虚拟网络设备与相应的 OpenFlow 交换机相连；安装 SDN 控制器，设置 OpenFlow 交换机到 SDN 控制器的通信链路，使 SDN 控制器与 OpenFlow 交换机保持连接，形成 NFV 与 SDN 综合的网络环境。注意到，SDN 控制器既能安装在实体主机上，也能安装在虚拟主机上。

由于 NFV 的灵活性，本文不仅可以将网络中间盒部署在虚拟网络的各个部分，也可以将网络中间盒与真实网络进行虚实互连。将运行虚拟网络功能中间盒的服务器与真实的 OpenFlow 交换机、服务器、PC 主机等网络设备用网线连接起来，并将网络功能中间盒的虚拟网卡配置为与服务器实际网络端口相连，这样就能实现外部实际网络设备与服务器中的虚拟网络功能中间盒的连接。在 NFV 网络中的虚实互连使得虚拟网络功能能够与真实网络融为一体，提高了它的实用价值。

4.2　检测中间盒实现

用软件实现检测中间盒的虚拟网络功能也是本文研究的要点之一。本文以 LXC 作为检测中间盒的运行环境，基于 Netfilter 框架[16]和 HOOK 技术设计了检测功能的程序。众所周知，常用的 HOOK 点有 5 种，分别是 PRE_ROUTING、LOCAL_IN、FORWARD、LOCAL_OUT 和 POST_ROUTING。其中的 FORWARD 是所有中间盒转发报文所必经的 HOOK 点，本文选用它作为设计检测中间盒的基础。

libnetfilter_queue 是一个用户态库，本文设计的用户态程序使用它来处理 NF_QUEUE 队列传输来的分组。检测中间盒程序使用 libnetfilter_queue 依次处理 NF_QUEUE 中的报文，让报文依次进入缓存区，通过 DDoS 检测算法进行数据检测，如果检测到 DDoS 攻击，则将它们丢弃，若检测到正常流量则进行转发。

图 3 给出了检测中间盒功能模块及其工作过程。当某 UDP 流进入检测中间盒后，Linux 的 iptables 指令使所有经过 FORWARD 点的报文进入 NF_QUEUE 队列中。接着，中间盒调用 libnetfilter_queue 用户态库函数来处理所有 NF_QUEUE 中的报文，依次采集报文的特征信息，检测程序分析报文中特征信息的熵值，判断其是否为 DDoS 攻击数据流。如果是，将其全部丢弃；如果不是，将其转发至 OpenFlow 交换机。

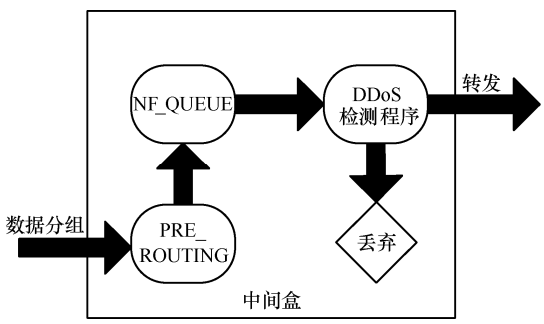


图 3　检测中间盒功能模块及其工作过程

5 实验与性能评价

本节结合了第 3 节防范 DDoS 攻击的模型以及第 4 节基于 NFV 的实现技术，在 Linux LXC 环境下使用 NFV 与 SDN 综合的方法，在数据平面按分布式方式在每个用户主机与 SDN 交换机之间部署了一个检测中间盒，实现了一种防范 DDoS 攻击 SDN 控制器的原型系统。通过对该原型系统进行实验测试，验证该系统的可行性，并对实验结果进行分析，评价系统的性能指标。

5.1 原型系统实现

原型系统以一台 Intel(R) Xeon(R) X5647 服务器作为宿主机，其主频为 2.93 GHz，内存为 32 GB，安装 Ubuntu 16.04 server 版系统。以 LXC 作为虚拟机，构建 3 台由 Open vSwitch[17] 生成的 OpenFlow 交换机 S_1、S_2 和 S_3，系统的 SDN 控制器采用运行在 LXC 中的 ONOS 控制器，控制器分别与 S_1、S_2、S_3 相连并控制它们的运行。基于 LXC 生成虚拟主机，其中 U_1、U_2、U_3、U_4、U_5 和 U_6 为可信用户，A_1、A_2、A_3 和 A_4 为不可信用户，并且 A_1、A_2、A_3 可能生成 DDoS 流。生成 DDoS 流的攻击工具是目前流行的 TFN2K[18]。在所有虚拟主机上安装 iperf 软件产生 TCP 或 UDP 流。在 LXC 中安装 DDoS 检测中间盒软件作为前置中间盒如 D_1、D_2、D_3 和 D_4 等，并将它们部署在不可信用户与 SDN 交换机之间。图 4 显示了一个基于 NFV 的防范 DDoS 攻击 SDN 控制器的原型系统的实例。

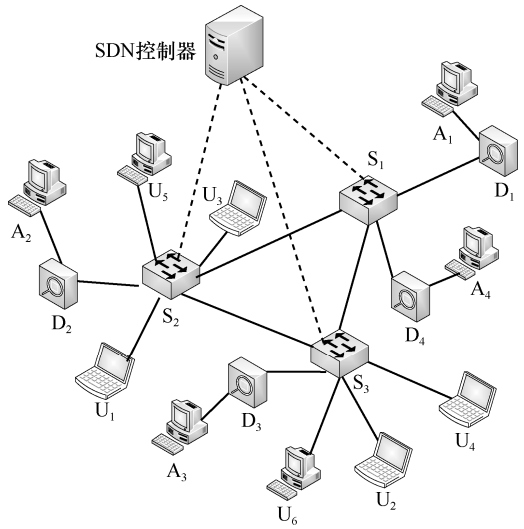


图 4 一个基于 NFV 的防范 DDoS 攻击 SDN 控制器的原型系统的实例

5.2 实验结果与分析

1) 滑动窗口大小对正常流时延的影响

设置整个实验网络各个端口普通情况下的报文传输速率约为 100 Mbit/s，每秒发送的报文数大约为 8 000~9 000 个。由于期望中间盒产生的平均网络时延大致为 6 ms，利用式(5)估算出检测中间盒的滑动窗口的大小约为 100 个报文。

实验过程：使用 A_4 向 U_4 发送约 100 Mbit/s 的 UDP 报文，每秒发送的报文数约 8 500 个，D_4 中间盒的滑动窗口大小在[50,150]区间内变动，统计不同大小的滑动窗口采集、分析处理报文的时长。实验重复 10 次，取平均值作为实验结果，如图 5 所示。

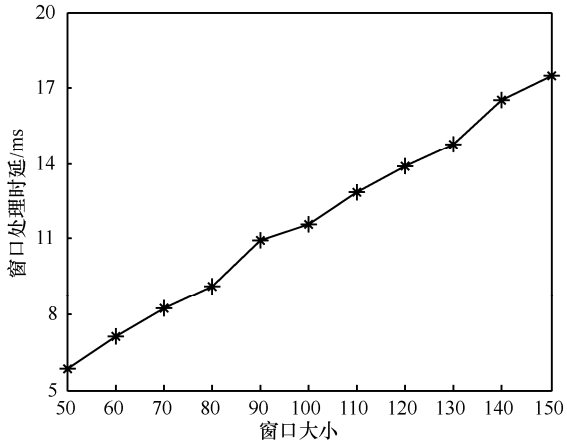


图 5 窗口大小对窗口处理时延的影响

从图 5 可知，随着滑动窗口的增大，处理滑动窗口中报文时延也会随之增加。当滑动窗口大小为 50 时，滑动窗口的处理时延不足 6 ms；而当滑动窗口大小为 100 时，时延增加到 11.56 ms；当滑动窗口大小为 150 时，处理时延已经接近 18 ms，实验表明，时延与窗口大小大致为线性关系。因此，根据网络应用的需求选择合适的滑动窗口大小是十分重要的。

根据本文实验环境的要求，需要将报文的平均网络时延控制在 6 ms 左右，所有报文时延区间为 0~12 ms，因此，本文实验中间盒的滑动窗口大小设置为 100。

2) DDoS 攻击流对控制器的影响

实验过程：在 T_0=0 s 时刻，使 A_1、A_2、A_3 和 A_4 向 SDN 中发送约 100 Mbit/s 的 TCP 或 UDP 报文，在 T_1=10 s 时刻，A_1、A_2、A_3 启动典型 DDoS 攻击应用 TFN2K 进行源 IP 地址随机变换的 DDoS 攻击。在 T_2=20 s 时刻实验结束。为了方便对实验数据进

行分析对比，分别在启动和不启动 DDoS 检测中间盒功能的 2 种情况下，统计各个 DDoS 检测中间盒端口转发报文到 SDN 中的报文数量。实验重复了 10 次，取平均值作为实验结果，如图 6 所示。

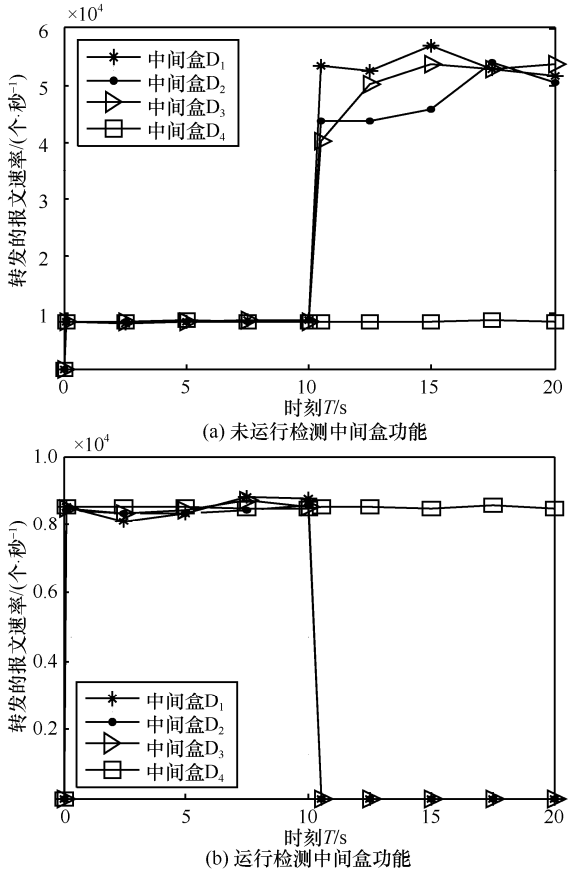


图 6　检测中间盒对转发报文数量的影响

由图 6(a)可知，在 0~10 s，4 个主机由于都在使用相对比较稳定的 iperf 发送报文，因此各个中间盒转发的报文的速率比较稳定，都保持在 8 000~9 000 个/秒。而 T_1 时刻，A_1、A_2 和 A_3 都启动 TFN2K 进行攻击，每个攻击源发送报文的速率大大提升，达到近 60 000 个/秒，大量攻击报文进入检测中间盒，由于此时检测功能并未打开，因此没有对攻击的报文进行检测防御，所有的攻击报文进入 OpenFlow 交换机和 SDN 控制器。由于 SDN 控制器集中式的控制，最终 SDN 控制器接收到的 DDoS 攻击报文的总量为所有攻击源攻击报文之和，约为 170 000 个/秒。由图 6(b)可知，在其他条件完全相同的情况下，当 T_1 时刻 DDoS 攻击报文大量涌入中间盒后，DDoS 检测中间盒 D_1、D_2、D_3 能够迅速地检测到 DDoS 攻击，并对后续所有报文进行丢弃，切断攻击源，使几乎所有的 DDoS 攻击报文都无法进入 OpenFlow 交换机和 SDN 控制器，SDN 得到了很好的保护。另外，检测中间盒 D_4 由于没有检测到攻击，因此可以继续转发主机 A_4 发送的正常报文，不会影响正常端口的数据传输。

该实验表明，在数据平面部署分布式前置检测中间盒可以有效地检测和预防 DDoS 攻击，将 DDoS 攻击报文阻挡在整个 SDN 的外部，使它们无法对 SDN 控制器甚至 OpenFlow 交换机造成危害。另一方面，分布式的前置检测防御不会因为 DDoS 攻击源的增加而造成 SDN 控制器的报文冗余，这种工作方式大大提高了网络系统的安全性和顽健性。

3) DDoS 攻击流对正常用户的影响

在上述实验 2)的过程中，使可信用户 U_5 不断尝试与可信用户 U_6 建立 TCP 连接，获取不同时刻 TCP 建立连接的时延，以此来反映 SDN 控制器在不同时刻的性能状况。实验结果如图 7 所示。

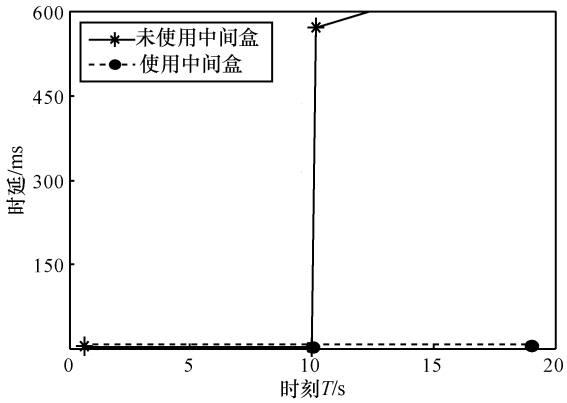


图 7　TCP 连接建立时延

图 7 给出了 2 种情况下 TCP 连接建立的时延。由于 SDN 中 TCP 在建立连接的时候需要 SDN 控制器下发流表这一重要步骤，因此，TCP 建立连接的时延大小与 SDN 控制器的运行状态有着紧密的联系。从图 7 可知，在 T_1 时刻前，由于网络环境没有受到 DDoS 攻击，2 种情况下的时延基本相同。而在 T_1 时刻，DDoS 攻击到来，若未使用检测中间盒功能，TCP 建立连接的时延大大增加，甚至会出现连接建立失败的情况；而在使用了检测中间盒情况下，DDoS 攻击报文被中间盒完全阻挡在了 SDN 之外，SDN 控制器完全没有受到影响，因此 TCP 建立连接的时延完全没有变化。

该实验表明，在数据平面部署分布式前置检测中间盒可以有效地保护 SDN 控制器，使其在遭遇 DDoS 攻击时依然可以安全稳定地工作，从而维持

整个 SDN 的正常运营。

6 结束语

尽管 SDN 技术与应用已经得到空前的发展，但 SDN 控制器受到 DDoS 攻击威胁的问题一直没有得到很好的解决。一旦 DDoS 攻击报文进入 SDN，就会消耗大量的控制器资源，使整个网络效率降低甚至瘫痪。利用本文提出的防范 DDoS 攻击 SDN 控制器的前置检测中间盒机制（UDM），一方面将 DDoS 检测和过滤功能分布式地设置在数据平面的前端，避免了大量攻击报文进入 SDN，从而有效地检测和预防 DDoS 攻击。另一方面，利用 NFV 虚拟网络功能的实现成本低、部署灵活的特点，系统具有较高的性能价格比。原型系统的实验结果表明，本文机制和方法具有可行性和有效性，能够很好地保护 SDN 的核心控制器，提高网络系统的安全性和顽健性。下一步，将思考如何更加灵活有效地部署检测中间盒，提高检测中间盒的工作效率，使其可以更好地服务于各种不同的网络环境。

参考文献：

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

[2] MIJUMBI R, SERRAT J, GORRICHO J L, et al. Network function virtualization: state-of-the-art and research challenges[J]. IEEE Communications Surveys & Tutorials, 2017, 18(1): 236-262.

[3] TOOTOONCHIAN A, GORBUNOV S, SHERWOOD R, et al. On controller performance in software-defined networks[C]//Usenix Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. 2012: 10.

[4] JARSCHEL M, OECHSNER S, SCHLOSSER D, et al. Modeling and performance evaluation of an OpenFlow architecture[C]//Teletraffic Congress. 2011: 1-7.

[5] ZHANG P, WANG H, HU C, et al. On denial of service attacks in software defined networks[J]. IEEE Network, 2016, 30(6): 28-33.

[6] SHIN S, YEGNESWARAN V, PORRAS P, et al. AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks[C]//ACM Sigsac Conference on Computer & Communications Security. 2013: 413-424.

[7] WANG H, XU L, GU G. FloodGuard: a DoS attack prevention extension in software-defined networks[C]//IEEE/IFIP International Conference on Dependable Systems and Networks. 2015: 239-250.

[8] KEROMYTIS A D, MISRA V, RUBENSTEIN D. SOS: secure overlay services[C]//ACM SIGCOMM '02 Conference. 2002: 61-72.

[9] ZHOU L, GUO H. Applying NFV/SDN in mitigating DDoS attacks[C]//2017 IEEE Region 10 Conference. 2017: 2061-2066.

[10] FUNG C J, MCCORMICK B. VGuard: a distributed denial of service attack mitigation method using network function virtualization[C]//International Conference on Network and Service Management. 2015: 64-70.

[11] JAKARIA A H M, YANG W, RASHIDI B, et al. VFence: a defense against distributed denial of service attacks using network function virtualization[C]//Computer Software and Applications Conference. 2016: 431-436.

[12] FUTAMURA K, KARASARIDIS A, NOEL E, et al. vDNS closed-loop control: a framework for an elastic control plane service[C]//Network Function Virtualization and Software Defined Network. 2016: 170-176.

[13] WANG R, JIA Z, JU L. An entropy-based distributed DDoS detection mechanism in software-defined networking[C]//IEEE Trustcom/bigdatase/ispa. 2015: 310-317.

[14] KUMAR K, JOSHI R C, SINGH K. A distributed approach using entropy to detect DDoS attacks in ISP domain[C]//International Conference on Signal Processing, Communications and Networking. 2007: 331-337.

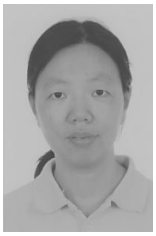
[15] BERNSTEIN D. Containers and cloud: from LXC to docker to kubernetes[J]. IEEE Cloud Computing, 2015, 1(3): 81-84.

[16] YANG Y, WANG Y. A software implementation for a hybrid firewall using linux netfilter[C]//Software Engineering. 2011: 18-21.

[17] PFAFF B, PETTIT J, KOPONEN T. The design and implementation of open vSwitch[C]//USENIX Networked System Design and Implementation. 2015: 117-130.

[18] DOULIGERIS C, MITROKOTSA A. DDoS attacks and defense mechanisms: classification and state-of-the-art[J]. Computer Networks, 2004, 44(5): 643-666.

[作者简介]



钱红燕（1973− ），女，江苏常州人，博士，南京航空航天大学副教授、硕士生导师，主要研究方向为计算机网络、信息安全等。

薛昊（1991− ），男，安徽宁国人，南京航空航天大学硕士生，主要研究方向为计算机网络、网络安全。

陈鸣（1956− ），男，江苏无锡人，博士，南京航空航天大学教授、博士生导师，主要研究方向为未来网络、网络功能虚拟化、无人机网络、网络安全等。