SDN 中 IP 欺骗数据分组网络溯源方法研究

魏松杰，孙鑫，赵茹东，吴超
（南京理工大学计算机科学与工程学院，江苏 南京 210094）

摘 要：IP 数据分组溯源方法是指从目的地址出发，逐跳找到源主机。该方法在软件定义网络(SDN, software defined network)框架下，通过控制器向网络中相关 SDN 交换机添加探测流表项，并根据目标数据分组触发的有效溯源 Packet-in 消息，找到目标数据分组的转发路径及源主机。所提方案可以为调试网络故障提供方便，使网络管理员可以得到任意一个数据分组的转发路径，应对 IP 地址欺骗等网络安全问题。实验证明，该溯源方法能够及时、准确地找到目标数据分组的转发路径，不影响网络中其他数据流转发，且无明显的系统开销。

关键词：SDN；IP 地址欺骗；IP 溯源；探测流表项
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2018243

Tracing IP-spoofed packets in software defined network

WEI Songjie, SUN Xin, ZHAO Rudong, WU Chao
School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Abstract: IP packets back tracing is to find the source host hop by hop from the destination. The method found the forwarding path of target packets and source host by adding probe entry into flow tables on SDN switches and analyzing the effective back tracing Packet-in messages sent by related switches. The proposed scheme can provide convenience for debugging network problems , so that the network administrator can obtain the forwarding paths of any data packets. Furthermore, it can help to solve the problem of IP spoofing. Experimental results prove that the traceability method can find the forwarding paths of target packets in a timely and accurate manner without affecting other traffic or significant system overhead.

Key words: SDN, IP address spoofing, IP trace back, flow table probe entry

1 引言

近年来，网络安全日益重要。分布式拒绝服务（DDoS, distributed denial of service）攻击是攻击方利用多台主机构成攻击平面对受害者实施攻击。攻击者通过发送大量数据分组，使受害者忙于处理大量的攻击数据分组，被迫减速甚至崩溃和关闭，从而使受害者拒绝向合法用户和计算机提供服务[1]。同时，DDoS 还可对网络设备（包括路由器、交换机等）和网络链路发起攻击，攻击者利用巨大的网络流量，造成路由器与交换机等网络设备过载，导致网络功能或正常服务瘫痪，网络性能大幅度下降。

网络攻击等网络恶意行为常使用 IP 地址欺骗。恶意主机使用伪造的 IP 地址不断给攻击目标发送大量数据分组，超出目标主机正常处理的能力范围，使受害者无法直接找到攻击者，达到隐藏恶意主机在网络中真实位置的目的[2]。此外，恶意主机使用 IP 地址欺骗绕过正常的数据分组请求者认证过程，取得目标系统信任，从而非法获得机密信息或访问受限制的网络服务。因此，如何及时检测具有伪造 IP 地址的数据分组，对网络安全有着重要意义。

收稿日期：2018−03−29；**修回日期**：2018−08−28
基金项目：国家自然科学基金资助项目（No.61472189）；赛尔网络下一代互联网技术创新项目（No.NGII20160105, No.NGII20170119）
Foundation Items: The National Natural Science Foundation of China (No.61472189), CERNET Innovation Project (No.NGII20160105, No.NGII20170119)

SDN[3]是一种转发控制分离的新型网络架构，SDN 的提出为解决网络攻击提供了新的思路。SDN 架构实现对网络集中控制，把转发规则的制定集中到控制器中。数据的转发由控制器下发的流表项统一指导，由交换机等网络转发设备实现转发过程，这极大地帮助网络提供商通过控制器以集中方式动态改变网络配置，而无需独立访问和重新配置分散在整个网络中的各个设备[4]。SDN 架构有利于网络功能的革新发展，其可编程与全局特性能够支持现有模块解决多种传统网络的现存问题[5]。

本文所提方案结合了 SDN 网络的优点，提出基于 SDN 的 IP 欺骗数据分组网络溯源方法。本方案完全依赖于 SDN 交换机的正常数据处理功能，不影响网络中其他正常数据流的转发，对交换机无需额外的软硬件修改，且本方案可以根据欺骗数据分组的实际转发情况，动态查找欺骗数据分组的转发路径，准确率高，系统开销小。

2 相关工作

IP 地址欺骗通过伪造 IP 协议数据分组组头中的源地址，使网络中数据分组转发节点和接收节点所看到的数据分组源 IP 地址并不是数据分组的实际发送者在网络中的真实 IP 地址。伪造数据分组的源 IP 地址，既能保证数据分组被网络正常路由转发到目标地址节点，又能使数据分组接收者难以定位真实来源。通过网络中不同节点间的协作，快速回溯欺骗数据分组的路由路径至真实发送主机或入口交换机，对提高网络安全水平、防范网络攻击威胁、调查取证网络安全事件等应用具有重要的技术意义。

IP 溯源的目的是找到发送目标数据分组的真实源主机及目标数据分组实际转发路径。IP 溯源常用的方法有链路测试法、数据分组标记法、日志记录法等。

Patel 等[6]使用链路测试法进行 IP 溯源，该方法首先由网络管理员构造一个覆盖网，当受害者受到攻击时，向受害者上游路由器注入洪流，加重攻击链路负载。在溯源时，根据受害者接收速率变化判断攻击路径，链路测试法需要人工完成很多工作，只能追溯单个攻击流。文献[7-8]使用数据分组标记法追踪目标数据分组。数据分组标记法对分组头空闲字段重定义，利用数据分组组头的一些字段，写入数据分组经过的路径信息，当带标记的数据分组到达受害主机且被确定为欺骗数据分组后，就可根据该数据分组中记录的路径信息找到源主机。Foroushani 等[9]使用日志记录法进行 IP 溯源。日志记录法是由路由器将经过的数据分组信息记录下来，以备溯源时使用，但这种方法要求系统具有较大的存储空间。

SDN 网络具有逻辑中心化和控制管理可编程策略，这为解决 IP 欺骗数据分组的溯源问题提供了新的技术支持和实施平台。Francois 和 Festor[10]在 SDN 架构下提出通过查看流表项的方法找到源主机，但是由于流表项的匹配精度不确定，找到的主机往往是一个集合，无法准确地定位源主机。夏彬[11]提出了使用 Renyi 熵的方法判断一个交换机是否转发过欺骗数据分组，Cui 等[12]提出通过查询交换机中一些统计数据判断一个交换机是否在欺骗数据分组的转发路径上。这两种溯源方法都受设置的阈值和实际网络中数据分组发送情况的影响较大。Agarwal 等[13]提出通过给交换机标号并添加流表项的方法从源主机追踪数据分组的发送路径。Tammana[14]和 Narayan[15]通过添加流表项，修改数据分组组头某些字段，使该字段记录路径信息，这种方法需要额外添加的流表项数目较多，且受数据分组组头字段长度的限制。各个溯源技术的详细性能评估，如表 1 所示。

本文在 SDN 框架下提出一种通过控制器向网

表 1 各个溯源技术评估

性能	传统溯源			SDN 溯源		
	链路测试法	数据分组标记法	日志记录法	Renyi 熵方法	CherryPick 方法	本文溯源方法
溯源速度	较慢	较快	较慢	一般	较快	快
路由开销	高	高	很高	低	低	低
网络开销	中	低	中	低	低	低
溯源受限	限追溯单个流	对路由器性能要求高	受限于存储空间	受限于窗口大小和阈值	受限于数据分组头部字段长度	特征集需人为提取

络中相关的交换机中添加探测流表项的方法，在网络中追踪欺骗数据分组的流入端口，找到发送 IP 欺骗数据分组的源主机。添加的探测流表项具有最高优先级，与溯源的欺骗数据分组特征进行精确匹配，对应的操作为发送 Packet-in 消息给控制器。当目标欺骗数据分组经过添加了探测流表项的交换机时，会触发相应的 Packet-in 消息，控制器根据该消息并利用全局网络拓扑信息得到上一跳信息。

3　系统设计

3.1　名词定义

本节定义或明确一些名词和概念，用以在后续章节中描述提出的溯源方案。

定义 1　欺骗数据分组的特征集(简称特征集)：特征集由溯源请求者提出。以 OpenFlow[16]v1.0 为例，特征集是流表项分组头域元素的集合的子集。

定义 2　探测流表项：探测流表项的匹配域是特征集以及入端口（入端口是必备项），在流表中具有最高匹配优先级，探测流表项的动作是发送 Packet-in 消息给控制器。

定义 3　溯源路径树：控制器中有一个树型结构，用以记录溯源数据分组的转发路径。每当查找到一个交换机（主机）时，都将该交换机的 DPID（主机的 IP）加入到溯源路径树中。由树的根节点到子节点得到溯源路径。

定义 4　有效溯源 Packet-in 消息：溯源开始后，触发该消息的数据分组符合溯源请求者提出的特征，控制器之前已向发送该消息的交换机添加过探测流表项，且发送该消息的交换机 in_port 端口连接的交换机或主机未被溯源过。

定义 5　无效溯源 Packet-in 消息：溯源开始后，触发该消息的数据分组符合溯源请求者提出的特征，但控制器之前未向发送该消息的交换机添加过探测流表项，或发送该消息的交换机 in_port 端口连接的交换机或主机已被溯源过。

3.2　溯源机制

本文通过添加探测流表项的方法找到欺骗数据分组的转发路径及源主机。图 1 是本溯源算法的具体流程图。溯源过程如下。

1) 溯源请求者给出特征集和溯源起点交换机的 DPID，控制器将溯源起点交换机 DPID 作为根节点加入溯源路径树中。溯源起点交换机为当前交换机。

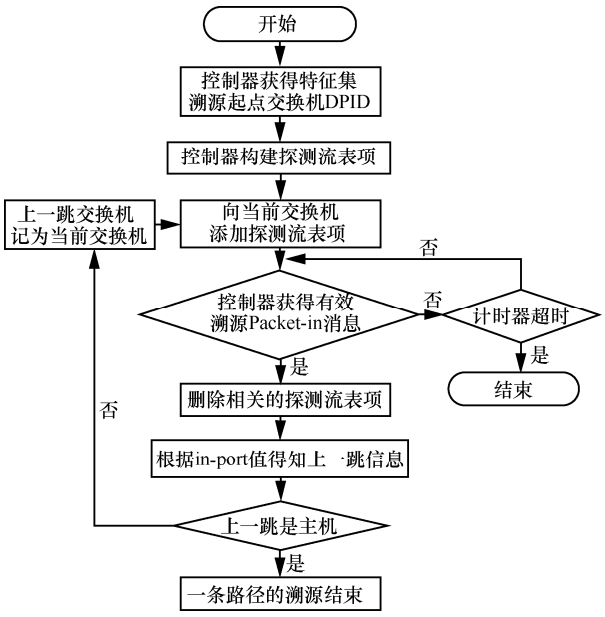

图 1　溯源流程

2) 控制器向当前交换机中添加 i 条探测流表项(i 为当前交换机的端口数目-1。除了数据分组经过的下一跳端口外，其余每个端口都对应一条探测流表项)，监听有效溯源 Packet-in 消息。

3) 若控制器接收到有效溯源 Packet-in 消息，控制器根据全局网络拓扑结构信息及 in_port 值得知数据分组经过的上一跳节点信息，将上一跳节点信息加入溯源路径树中。若上一跳节点是交换机，则将上一跳交换机记为当前交换机。控制器将该数据分组 Packet-out 至下一跳节点。

4) 控制器删除发送有效溯源 Packet-in 消息的交换机中对应的探测流表项。

5) 重复 2)～4)，当查找到的交换机 in_port 端口连接的是主机时，一条路径的溯源结束。

6) 控制器长时间没有收到有效溯源 Packet-in 消息，计时器超时，溯源结束。

3.3　溯源过程示例

本节以图 2 中所示 SDN 网络拓扑结构为例，对本文提出的通过添加探测流表项溯源 IP 数据分组的方法进行示范说明。

在拓扑网络中，主机 $h1$ 使用虚假的 IP 地址向主机 $h4$ 发送数据分组，在查找该数据分组的转发路径时，溯源请求者 $h4$ 需告知控制器溯源起点交换机的 DPID（交换机 $s4$ 的 DPID）及数据分组的特征集，从主机 $h4$ 开始溯源。溯源请求者给出的特征集是：源 IP 地址、目的 IP 地址、数据分组的类型以及目的端口号。

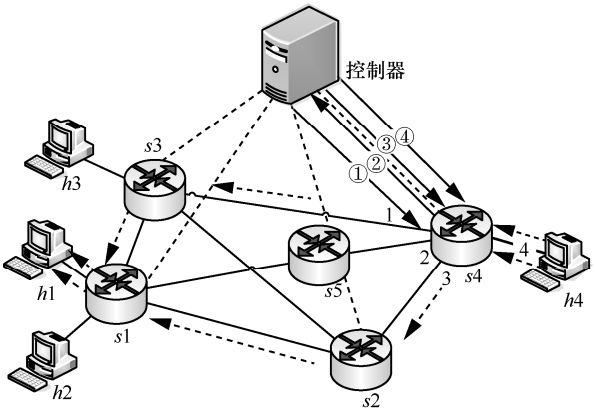

图 2 算法实现过程示例

控制器构造的探测流表项如下。

1) 匹配域：数据分组入端口及与特征集对应的 2～4 层网络控制信息。

2) 优先级：65 535（最高优先级）。

3) 动作：将数据分组封装为 Packet-in 消息并转发给控制器。

在溯源过程中，将欺骗分组经过的交换机标记为目标节点，将目标节点加入到溯源路径树上，遍历溯源路径树得到目标节点匹配到的欺骗数据分组数量，由此得到欺骗分组来自溯源目标的概率。

控制器向交换机 *s*4 添加 3 条探测流表项（如图 2 中①所示，添加的 3 条探测流表项分别对应于 *s*4 的端口 1、端口 2 和端口 3），当欺骗数据分组经过 *s*4 时，会触发有效溯源 Packet-in 消息（如图 2 中②所示）。控制器根据全局网络拓扑信息及该 Packet-in 消息中 in_port 值得知 *s*4 的上一跳记为 *s*′。图 2 中 *s*4 是起点交换机，欺骗分组经过的概率为 1，*s*′为交换机 *s*3 和 *s*2，欺骗分组经过的概率与溯源路径树上匹配到的欺骗数据分组的数目有关，没有欺骗数据分组经过交换机 *s*5，则 *s*5 的概率为 0。然后控制器将对应端口的探测流表项的动作修改为转发给下一跳（如图 2 中③所示），并将触发该消息的数据分组 Packet-out 至 *s*4 的端口 4（如图 2 中④所示）。接下来，控制器向 *s*′添加探测流表项，当欺骗数据分组经过 *s*′时会触发有效溯源 Packet-in 消息，控制器得知 *s*′的上一跳是 *s*″，同样根据溯源路径树中对应节点经过的欺骗数据分组数目得到路由器概率。控制器将对应端口的探测流表项的动作修改为转发到下一跳并将触发该消息的数据分组 Packet-out 至相应端口。接下来控制器会向 *s*″添加探测流表项，当欺骗数据分组经过 *s*″时会触发有效溯源 Packet-in 消息，控制器根据全局网络拓扑信息及该 Packet-in 消息中 in_port 值得知 *s*″的上一跳是主机 *h*1，一条路径的溯源结束。

3.4 流表项管理

本文提出的 IP 溯源方法中，转发和溯源均使用了控制器获取全局网络拓扑结构的功能。控制器利用拓扑发现协议获取全局网络拓扑信息，将全局网络拓扑信息保存在一个图结构中，并利用路径计算方法计算数据分组的转发路径。当数据分组触发 Packet-in 消息时，若正常转发该数据分组，控制器会利用全局网络拓扑信息计算该数据分组经过的路径，并向该路径上所有交换机下发转发该数据分组的流表项。

流表项指导数据分组转发。传统网络中路由表项是路由器运行路由协议自动构建的，而 SDN 交换机中的流表项由控制器下发，且 SDN 交换机中的流表项可更精确地匹配，进行更复杂的数据分组处理动作。如图 3 所示，流表项由匹配域、优先级、计数器、指令集、超时计时器、缓存等字段构成。

匹配域	优先级	计数器	指令集	超时计时器	Cookie

图 3 SDN 交换机中流表项结构

本文主要使用了流表项的匹配域、优先级和指令集 3 个字段。匹配域包括 1 到 4 层的网络信息，与符合条件的数据分组匹配。优先级值的范围是 [0, 65 535]，数据分组优先与优先级高的流表项匹配。指令集中包含一组指令，当匹配到该流表项时，按一定顺序执行这些指令。本文只使用一级流表结构，即所有流表项均位于流表 0 中。用于处理正常数据分组转发的流表项（不包括 table-miss 流表项）中，匹配域为以太网协议类型和目的 IP 地址，优先级为 1，指令集动作是转发到交换机某一端口。

本文中，控制器对探测流表项的下发和修改都是通过 OpenFlow 协议中 Flow-mod 消息实现的。通过 Flow-mod 消息，控制器能够指定下发给网络交换机的流表项的匹配域、优先级、生存时间等，也可以修改某一个交换机中特定的流表项。

3.5 多径处理

本文溯源得到的数据分组的转发路径会构成一个树型结构（若溯源只得到一条路径，则溯源产生的路径构成一条直线）。为了能够处理多条路径的情况，控制器中保存了用于记录多条路径的树型数据结构，当前得到的节点是溯源路径树中节点的子节点。

在多径处理中，对有效溯源 Packet-in 消息的判断，保证了溯源得到的节点父子关系的准确性。

在溯源过程中，控制器向交换机添加探测流表项条数为其端口数目-1，对应于交换机不同的端口。在同一个交换机中，数据分组触发的每一个有效溯源 Packet-in 消息对应一个溯源路径树的分支。若有多个主机发送目标数据分组，数据分组会在路径汇合的交换机中与多个探测流表项匹配，从而触发多个有效溯源 Packet-in 消息。

3.6　目标可能性计算

根据溯源路径树上每个节点的有效溯源 Packet-in 消息与上一跳的节点信息，可以统计出到达当前节点的欺骗分组中属于上一节点的概率。基于这个统计数据，本文使用贝叶斯网络模型计算溯源起点的欺骗分组来自每个溯源目标的概率，即贝叶斯网络中目标节点的先验概率。控制器将对应端口的探测流表项的动作修改为转发给下一跳，在回溯到溯源终点后，遍历溯源路径树得到相应节点匹配到的欺骗数据分组数量。

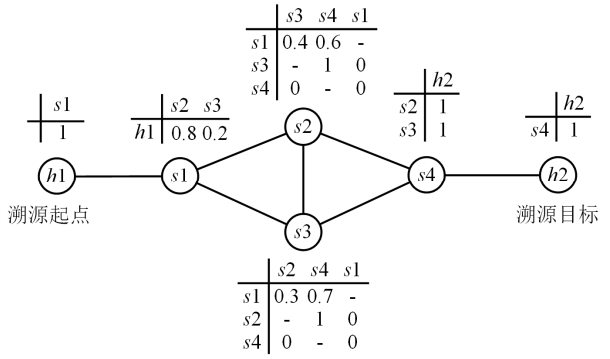

图 4　贝叶斯网络模型

贝叶斯网络是一个有向无环图，它的节点表示随机变量，两个节点间的箭头包含一个条件概率值，该值表示节点的依赖关系。通过溯源路径树可以直接构造贝叶斯网络拓扑图，当前节点来自上一节点的有效溯源 Packet-in 消息概率为条件概率值。

如图 4 所示，*h*1 是溯源起点，*h*2 是溯源目标，*s*1、*s*2、*s*3、*s*4、*s*5、*s*6 是交换机，根据贝叶斯联合概率分配函数

$$P(X_i=x_i,\cdots,X_n=x_n)=\prod_{i=1}^{n}P\left(X_i=x_i\middle|X_j=parent\left(x_i\right)\right) \tag{1}$$

可以得到

$$P(s4)=P(s4\,|\,s2)P(s2)+P(s4\,|\,s3)P(s3) \tag{2}$$

$$P(h2)=P(h2\,|\,s4)P(s4) \tag{3}$$

4　实验与分析

本文实验均基于 Ubuntu 15.10 版本 Linux 系统搭建，数据平面使用 mininet 网络虚拟平台，控制器使用 ryu，ryu 和 mininet 分别安装在两台虚拟机上，使用的分组发送工具是 hping3。

4.1　特征集设定

图 5 所示为实验网络拓扑。实验中主机 *h*1、*h*3、*h*5、*h*7 的 IP 地址分别为 10.0.0.1、10.0.1.3、10.0.2.5、10.0.3.7，交换机 *s*1～*s*8 的 DPID 分别为 1～8。在实验过程中，*h*1、*h*3、*h*5 都向 *h*7 发送数据分组，其中，主机 *h*1 冒用主机 *h*3 的 IP 地址向 *h*7 发送数据分组，数据分组转发路径为 *h*1—*s*1—*s*2—*s*3—*s*4—*s*7—*h*7，同时，主机 *h*3、*h*5 也向 h7 发送数据分组，数据分组的转发路径分别为：*h*3—*s*5—*s*4—*s*7—*h*7，*h*5—*s*6—*s*8—*s*7—*h*7，如图 5 所示。

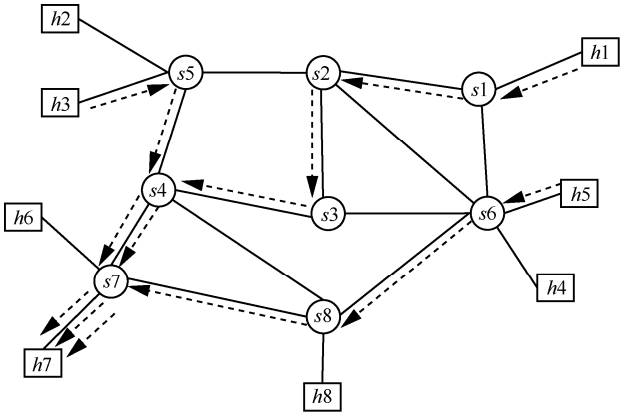

图 5　实验网络拓扑

各主机发送数据分组的字段情况如表 2 所示，本文进行了两组实验，如表 3 所示，两组实验中溯源请求者 *h*7 分别给出了两组不同的特征集，表中“—”表示没有给出该项内容，需要控制器找到符合该特征集的数据分组的转发路径。

表 2　　各主机发送数据分组情况

数据分组字段	*h*1	*h*3	*h*5
源 IP 地址	10.0.1.3	10.0.1.3	10.0.2.5
目的 IP 地址	10.0.3.7	10.0.3.7	10.0.3.7
IP 协议类型	UDP	UDP	UDP
源端口号	6000	6500	7000
目的端口号	80	80	80

表 3 两组实验中的特征集

特征集元素	第一组	第二组
以太网协议类型	2048	2048
源 IP 地址	10.0.1.3	10.0.1.3
目的 IP 地址	10.0.3.7	10.0.3.7
IP 协议类型	17(UDP)	17(UDP)
源端口号	—	6000
目的端口号	80	80

图 6 和图 7 分别是使用表 3 第一组特征集和第二组特征集进行实验得出的结果。在实验中，各主机发送分组的分组间间隔均为 1 s。由于第一组的特征集无法区分开 *h*1 和 *h*3 发出的数据分组，因此控制器找到两条路径。第二组可以区分开欺骗数据分组和其他类数据分组，因此可以精确找到欺骗数据分组的转发路径。由此可知，溯源请求者给出的特征集匹配精度越高，本方案越能精确地找到被溯源数据分组的转发路径。

```
==========================================
07:10:01.87378  7  07:10:01.88959  4
07:10:02.87704  4  07:10:02.89286  5
07:10:02.89394  10.0.1.3  07:10:03.88661  3
07:10:04.89108  2  07:10:05.89768  1
07:10:05.89916  10.0.0.1
10.0.3.7->7->4->5->10.0.1.3
10.0.3.7->7->4->3->2->1->10.0.0.1
==========================================
```

图 6 第一组实验溯源结果

```
======================================
07:09:05.04092  7  07:09:06.04367  4
07:09:07.04833  3  07:09:08.05620  2
07:09:09.06164  1  07:09:09.06243  10.0.0.1
10.0.3.7->7->4->3->2->1->10.0.0.1
======================================
```

图 7 第二组实验溯源结果

4.2 结合贝叶斯网络模型的溯源

图 8 为实验网络拓扑表示欺骗分组的分布，该网络拓扑可以实现任意两台交换机的多条路径连接，方便贝叶斯网络模型下目标节点先验概率的计算。

在网络拓扑中，*s*1 到 *s*8 的 DPID 分别为 1～8，为了考虑复杂情况下的分组发送情况，实验中增加背景流量，使用 D-ITG 作为背景流量发起工具。D-ITG 可以发送不同协议、不同大小、不同速率的数据分组。实验过程如下。

1) 主机 *h*1 到 *h*11 中任两台主机相互进行分组发送，其中主机 *h*1 使用虚假的 IP 地址发送数据分组，*h*1、*h*2、*h*5 的发送分组速率分别是 30 packet/s、5 packet/s、3 packet/s，其余主机发送分组速率为 5 packet/s，各主机所发送的分组大小不同。

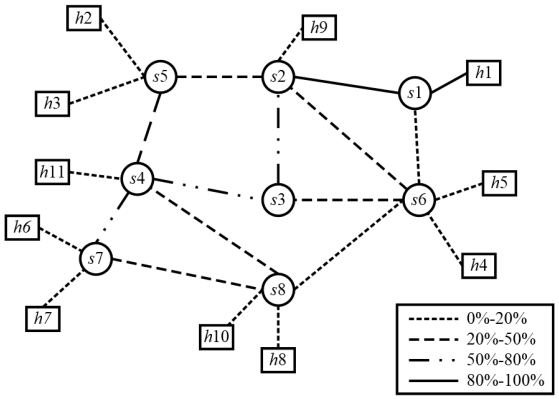

图 8 实验网络拓扑

2) 在网络中选取主机作为溯源起点，本次选取 *h*7 进行实验，欺骗主机 *h*1 向 *h*7 发送欺骗数据分组的转发路径有 3 条，分别是 *h*1—*s*1—*s*2—*s*5—*s*4—*s*7—*h*7、*h*1—*s*1—*s*2—*s*3—*s*4—*s*7—*h*7、*h*1—*s*1—*s*6—*s*8—*s*7—*h*7，同时，主机 *h*2 到 *h*6 向主机 *h*7 正常发送分组。

3) 对于网络多变的特性，欺骗数据分组的转发路径可能有多条，每条路径上的路由器转发的欺骗分组数量也不同。在交换机的组表项内设置 select 类型，在组表项中设置多个行动桶，每个行动桶中指定面向物理端口的 Output 行动，数据分组通过行动桶处理，在各行动桶中设置不同权重作为各桶的使用频率。在交换机 *s*1 中，端口 *s*1～*s*2 与端口 *s*1～*s*6 的行动桶权重比为 7:3, 在交换机 *s*2 中，端口 *s*1～*s*2 与端口 *s*1～*s*6 的行动桶权重比为 4:1。

控制器中的溯源路径树记录数据分组的转发路径，通过溯源路径树构造贝叶斯网络拓扑图，使用贝叶斯网络模型计算溯源起点的欺骗分组中来自每个溯源目标的概率，实验结果如图 9 所示。

为了在复杂情况下进行溯源准确性的研究，本实验拓扑具有任意两个主机之间存在多条可达路径的特点。实验中多台主机向溯源起点发送数据分组，控制器向交换机下发探测流表项，并把目标节点加入到溯源路径树上。

控制器根据特征集从溯源起点逐跳找到对应的目标主机。对于有欺骗分组经过的交换机，控制器计算不同路径发送过来的欺骗分组的数量，统计出不同路径上欺骗分组的概率。根据实验结果，从溯源起始点 *h*7 开始，标记到达当前节点的欺骗分组中属于上一节点的概率，实验结果与理论值接近。

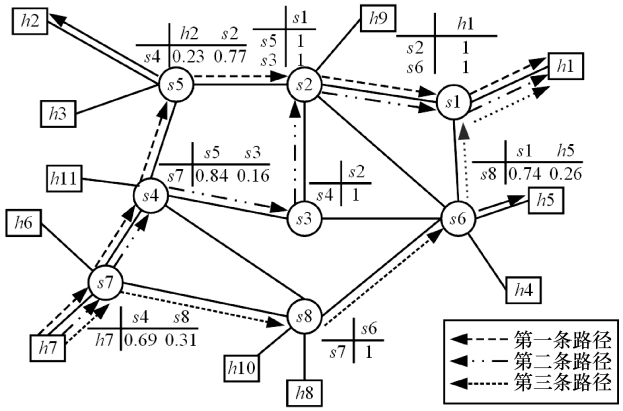

图 9　实验结果

4.3　溯源准确性

为了更直接地运用贝叶斯理论，整理图 9 实验结果如表 4 所示。

表 4　　实验结果

事件	概率	意义
$P(h7)$	100%	溯源起点
$P(s7\|h7)$	100%	溯源第一台交换机
$P(s4\|s7)$	69%	第一、二条路径
$P(s5\|s4)$	84%	第一条路径
$P(s2\|s5)$	77%	第一条路径
$P(h2\|s5)$	23%	溯源终点 $h2$
$P(s1\|s2)$	100%	第一、二条路径
$P(h1\|s1)$	100%	溯源终点 $h1$
$P(s3\|s4)$	16%	第二条路径
$P(s2\|s3)$	100%	第二条路径
$P(s1\|s2)$	100%	第二条路径
$P(h1\|s1)$	100%	溯源终点 $h1$
$P(s8\|s7)$	31%	第三条路径
$P(h6\|s8)$	100%	第三条路径
$P(s1\|s6)$	74%	第三条路径
$P(h5\|s6)$	26%	溯源终点 $h5$
$P(h1\|s1)$	100%	溯源终点 $h1$

结合表 4 与贝叶斯联合概率分配函数，在 3 条溯源路径的终端，$h1$ 的概率为 78.6%，$h2$ 的概率为 13.3%，$h5$ 的概率为 8.1%，则 $h1$ 是欺骗主机的可能性为 78.6%，相应地，本次实验的溯源准确性为 78.6%。当目标欺骗数据分组经过添加了探测流表项的交换机时，会触发相应的 Packet-in 消息，控制器根据该消息并利用全局网络拓扑信息得到上一跳信息，当欺骗数据分组的特征集符合该网络的主机发送分组内容时，本文的 IP 欺骗数据分组网络溯源方法认为该主机有一定的可能性是 IP 欺骗者，且可能性与该主机的发送分组情况有关。为了能更好地反映发送分组速率与溯源准确性的关系，只改变 $h1$ 发送分组的速率，通过实验得到各个溯源目标的概率，获得溯源准确性，如表 5 所示。

表 5　　速率与溯源准确性

速率/(packet·s^{-1})	溯源准确性
2	49.1%
20	70.4%
100	91.2%
500	98.6%

可以得到，使用虚假 IP 地址的主机 $h1$ 发送分组速率增大时，溯源准确性对应增加。本文所提的 IP 溯源方法适用于在网络攻击中找到发送目标数据分组的真实源主机及目标数据分组实际转发路径。

通过以上实验可以看出，本文的 IP 欺骗数据分组网络溯源能够快速找到欺骗数据分组的转发路径，定位欺骗数据分组的真实源地址，并在贝叶斯网络模型下计算目标节点的概率。本文的溯源方法能够在网络繁忙、流量复杂的情况下快速地进行溯源并计算目标节点的概率，提供可靠的网络溯源结果。

4.4　网络攻击对溯源影响

为了测试网络攻击对溯源的影响，本文使用 TFN2K 工具。DDoS 攻击工具 TFN2K 可以发起 UDP flood 攻击、TCP/SYN flood 攻击、ICMP/PING flood 攻击、ICMP/SMURF flood 攻击。实验网络拓扑如图 8 所示，主机 $h2$、$h3$、$h5$ 作为攻击者向 $h7$ 发起 UDP/TCP/ICMP 混合 flood DDoS 攻击。$h1$ 假冒 $h4$ 的 IP 地址向主机 $h7$ 发送欺骗分组，欺骗分组转发路径为 $h1$—$s1$—$s2$—$s3$—$s4$—$s7$—$h7$，发送分组间隔为 10 ms，$h7$ 作为溯源请求者进行溯源，溯源特征集为：$h7$ 的 IP 地址，$h1$ 虚假 IP 地址（即 $h4$ 的 IP 地址），数据分组类型以及目的端口号。

实验在 3 种不同情况下进行测试，分别是网络没有攻击时、网络有攻击、网络有攻击且攻击者 $h3$ 冒用 $h4$ 的 IP 地址向 $h7$ 发起攻击，其中，第三种情况下攻击路径被溯源，攻击对溯源路径造成干扰。实验溯源结果如图 10 所示。

由实验结果可以得到，未被攻击时溯源时间最短，溯源效率最高；当网络有攻击时，攻击影响交换机，溯源效率受到一定影响，溯源时间增加；当攻击者冒用 $h4$ 的 IP 地址向 $h7$ 发送分组时，可根

据溯源特征集溯源到攻击路径。结果表明，网络有攻击时溯源仍能进行，但是溯源时间增加。

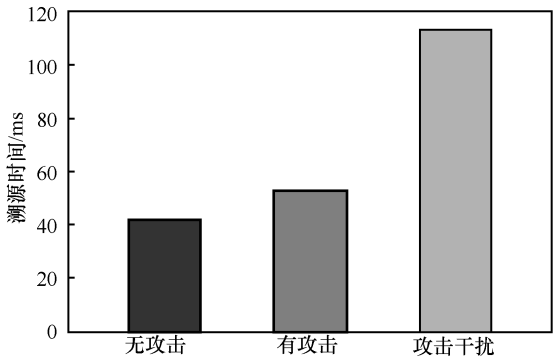

图 10 网络攻击对溯源影响

4.5 溯源效率

本节实验分别测试了控制器处理有效溯源 Packet-in 消息和无效溯源 Packet-in 消息所用时间。获得控制器处理有效溯源 Packet-in 消息用时的方法是在溯源算法 PacketInHandler 函数处理有效溯源 Packet-in 消息的代码段中，分别获得进入该代码段的时间和离开该代码段的时间，即可得到控制器处理该有效溯源 Packet-in 消息所用时间。本实验用获得的 50 次控制器处理有效溯源 Packet-in 消息所用时间的平均值作为控制器处理有效溯源 Packet-in 消息所用时间。获取控制器处理无效溯源 Packet-in 消息所用时间的方法与之类似。

实验得到控制器处理有效溯源 Packet-in 消息所用时间为 1.61 ms，控制器处理无效溯源 Packet-in 消息所用时间为 0.64 ms。控制器处理处理有效溯源 Packet-in 消息和无效溯源 Packet-in 消息所用时间都较小，即在溯源过程中，给控制器增加的额外系统开销较小。

图 11 和图 12 都是在特征集能区分开被溯源数据分组和其他类数据分组时进行实验得出的结果。

如图 11 所示，相比于跳数较多的情况，跳数在前两跳时，分组间间隔 1 ms 时的时延接近于分组间间隔 5 ms 的时延。原因主要是分组间间隔较小，从欺骗数据分组触发 Packet-in 消息到控制器修改探测流表项这段时间，欺骗数据分组会触发较多 Packet-in 消息，这些消息都是无效溯源 Packet-in 消息，控制器耗费部分时间用于处理这些无效消息。跳数较少时，分组间间隔为 1 ms 时额外增大的时延占主要部分，因此小于 3 跳时发送分组间隔为 1 ms 的时延接近于发送分组间隔为 5 ms 的时延；而高于 3 跳，两者时延与跳数呈线性相关，不再接近。

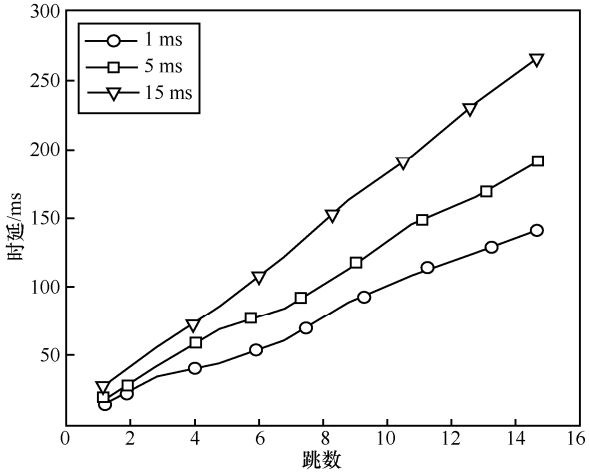

图 11 时延测试

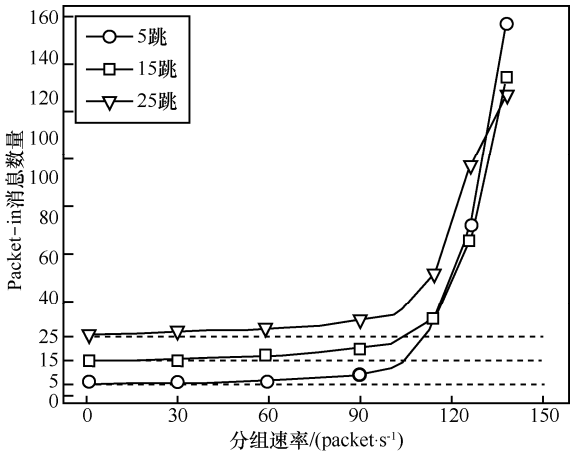

图 12 溯源过程中产生的 Packet-in 消息数量测试

图 12 是在欺骗数据分组从源主机到目的主机中间分别经过了 5 跳、15 跳、25 跳交换机的网络拓扑中实验得出的结果。从图 12 可以看出，分组速率较小时，欺骗数据分组触发的 Packet-in 消息逐渐接近于欺骗数据分组经过的交换机数目。

4.6 溯源性能

为了评估本文溯源算法对控制器的 CPU 使用率的影响，分别在不使用溯源算法、使用溯源算法以及使用 CherryPick 溯源算法 3 种情况下测试控制器的 CPU 使用情况。实验开始后通过加快发送分组速率，增加网络流量，观察控制器 CPU 使用率的增长情况，如图 13 所示。

从图 13 中可以看出，在发送分组速率增加的情况下，不使用溯源算法控制器的 CPU 使用率基本维持不变；使用本文溯源算法溯源时，控制器的 CPU 使用率有所增加，但增加速度十分缓慢；相比之下使用 CherryPick 溯源算法 CPU 使用率大幅增加，需要消耗较多 CPU 资源。

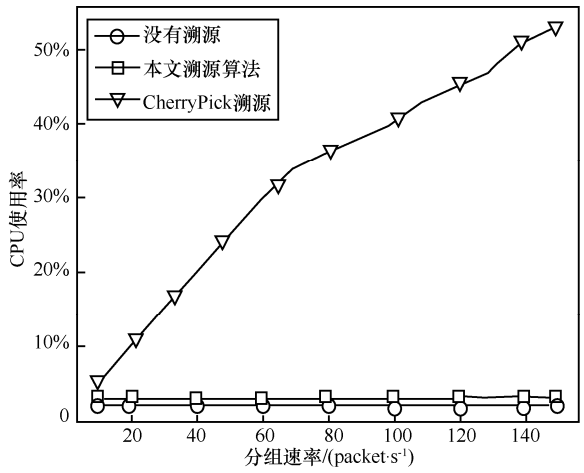

图 13　溯源过程控制器 CPU 使用率

5　结束语

本文介绍了传统网络和 SDN 网络中常用的 IP 数据分组溯源方法，提出了 SDN 中通过控制器向网络中相关的交换机添加探测流表项的 IP 数据分组溯源方法，以找到发送 IP 欺骗数据分组的真实主机或入口交换机。实验结果证明，该方法能准确地找到源主机及数据分组转发路径，系统开销小，不依赖于特定设备或网络拓扑结构，且不影响网络中其他正常数据流的转发。本文所提算法中特征值通过人工对数据分组的分析提取而获得，还没有实现对特征值的自动获取，下一步需研究自动化提取数据分组的特征集，而且，发送分组速率太大时，会触发较多 Packet-in 消息，需要进一步研究限制 Packet-in 消息速率的方案。

参考文献：

[1] METI N, NARAYAN D G, BALIGAR V P. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks[C]//International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2017: 1366-1371.

[2] 阎冬. IP 网络溯源方法及协作模式相关技术研究[D]. 北京：北京邮电大学,2012.
YAN D. Research on IP traceback techniques and collaboration pat terns[D]. Beijing: Beijing University of Posts and Telecommunications, 2012.

[3] FEAMSTER N, REXFORD J, ZEGURA E. The road to SDN: an intellectual history of programmable networks[J]. Acm Sigcomm Computer Communication Review, 2014, 44(2):87-98.

[4] SIEBER C, OBERMAIR A, KELLERER W. Online learning and adaptation of network hypervisor performance models[C]//IFIP/IEEE Symposium on Integrated Network and Service Management (IM). 2017: 1204-1212.

[5] 王涛, 陈鸿昶, 程国振. 软件定义网络及安全防御技术研究[J]. 通信学报, 2017, 38(11):133-160.
WANG T, CHEN H C, CHENG G Z. Research on software-defined network and the security defense technology[J]. Journal on Communications, 2017, 38(11):133-160.

[6] PATEL B, MENARIA S. Survey of traceback methods[J]. Journal of Engineering Computers & Applied Sciences, 2015, 4(1):22-26.

[7] 黄琼, 熊文柱, 阳小龙, 等. 分层次的无状态单分组 IP 溯源技术[J]. 通信学报, 2011, 32(3):150-157.
HUANG Q, XIONG W Z, YANG X L, et al. Hierarchical stateless sin-gle-packet IP traceback technique[J]. Journal on Communications, 2011, 32(3):150-157.

[8] YAN D, WANG Y, SU S, et al. A precise and practical IP traceback technique based on packet marking and logging[J]. Journal of In-formation Science & Engineering, 2012, 28(3):453-470.

[9] FOROUSHANI V A, ZINCIR-HEYWOOD A N. Deterministic and authenticated flow marking for IP traceback[J]. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 2013, 30(1): 397-404.

[10] FRANCOIS J, FESTOR O. Anomaly traceback using software defined networking[C]// IEEE International Workshop on Information Forensics and Security. IEEE, 2014.

[11] 夏彬. 基于软件定义网络的 WLAN 中 DDoS 攻击检测和防护[D]. 上海: 上海交通大学, 2015.
XIA B. Research on the detection and defence of DDoS attack in SDN-based WLAN[D]. Shanghai: Shanghai Jiaotong University, 2015.

[12] CUI Y, YAN L, LI S, et al. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks[J]. Journal of Network and Computer Applications, 2016, 68: 65-79.

[13] AGARWAL K, ROZNER E, DIXON C, et al. SDN traceroute: tracing SDN forwarding without changing network behavior[C]// The Workshop on Hot Topics in Software Defined NETWORKING. 2014: 145-150.

[14] TAMMANA P, AGARWAL R, LEE M. CherryPick: tracing packet trajectory in software-defined datacenter networks[C]// ACM SIGCOMM Symposium on Software Defined NETWORKING Research. 2015:1-7.

[15] NARAYANA S, REXFORD J, WALKER D. Compiling path queries in soft-ware-defined networks[C]// The Workshop on Hot Topics in Software Defined NETWORKING.. 2014:181-186.

[16] NICK M K,ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. Acm Sigcomm Computer Communication Review, 2008, 38(2):69-74.

[作者简介]

魏松杰（1977−），男，江苏南京人，博士，南京理工大学副教授、硕士生导师，主要研究方向为信息安全、无线网络与移动计算、物联网区块链等。

孙鑫（1993−），女，河南周口人，南京理工大学硕士生，主要研究方向为软件定义网络、异常检测等。

赵茹东（1992−），男，山东枣庄人，南京理工大学硕士生，主要研究方向为软件定义网络、网络流量分析等。

吴超（1994−），男，江苏扬州人，南京理工大学硕士生，主要研究方向为计算机网络、流量混淆等。