基于地址重载的 SDN 分组转发验证

吴平[1]，常朝稳[1]，左志斌[2]，马莹莹[1]
（1. 信息工程大学密码工程学院，河南 郑州 450004；2. 河南工业大学信息科学与工程学院，河南 郑州 450001）

摘 要：针对软件定义网络（SDN）中现有转发验证机制大多通过加入新的安全通信协议实现分组逐跳转发验证，出现通信与计算开销的问题，提出了一种基于地址重载的 SDN 分组转发验证机制。入口交换机通过重载分组地址信息将流运行时间划分为连续随机的时间间隔，各后继节点基于重载的地址信息转发分组；控制器采样间隔内流入口与出口交换机的转发分组，检测路径中的异常转发行为；最后，构建仿真网络实现了所提机制。实验结果表明，该机制以引入不超过 8%的转发延迟，可有效检测异常。

关键词：软件定义网络；地址重载；哈希采样；异常检测
中图分类号：TP393
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022047

Address overloading-based packet forwarding verification in SDN

WU Ping[1], CHANG Chaowen[1], ZUO Zhibin[2], MA Yingying[1]
1. Department of Cryptogram Engineering, Information Engineering University, Zhengzhou 450004, China
2. College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

Abstract：Aiming at the problem that the most existing forwarding verification mechanisms in software-defined network (SDN) verified packets hop-by-hop by incorporating new secure communication protocols, which incurred significant computation and communication overhead, an address overloading-based forwarding verification mechanism was proposed. The flow runtime was divided into consecutive random intervals by the ingress switch via overloading address fields of packet, basing on overloading address, packets were forwarded by each subsequent switch, and the controller sampled the packets forwarded by ingress and egress switch in the interval to detect abnormal behavior on the path. Finally, the proposed mechanism and simulation network was implemented and evaluated. Experiments show that the mechanism achieves efficient forwarding and effective anomaly detection with less than 8% of additional forwarding delays.
Keywords: software-defined networking, address overloading, hash-based sampling, anomaly detection

0 引言

软件定义网络[1]（SDN, software defined network）通过解耦数据平面与控制平面，重塑了传统网络僵化的体系结构。SDN 以其灵活性、可编程性为管理配置网络和快速部署新的协议带来了便利，也不可避免地产生了许多安全问题，如缺乏隔离的应用层程序、因遭受分布式拒绝服务（DDoS, distributed denial of service）攻击使控制器单点失效、数据平面基础设施资源的消耗以及开放的 API 隐患等，这些安全问题既包括传统网络的安全问题，也包括 SDN 专有的网络安全问题[2-3]。

不仅在传统的 IP 网络中，而且在新兴的 SDN 中，针对分组转发的攻击依然是严峻的问题[4-5]，攻击者可对一台受控的交换机节点中转发规则恶意的错误配置实施插入、删除、延迟、修改、重放、丢弃数据分组。

收稿日期：2021−10−16；**修回日期**：2021−12−26
通信作者：左志斌，zzbin_2000@163.com
基金项目：国家自然科学基金资助项目（No.61572517；河南省科技攻关基金资助项目（No.222102210070）
Foundation Items: The National Natural Science Foundation of China (No.61572517), Science and Technology Project of Henan Province (No.222102210070)

因此，准确、高效的分组转发验证一直是面对恶意攻击时确保转发正确性的关键课题。

SDN 中，虽然控制器可以通过交换机流表转发状态获取分组转发统计信息，但分组传输时延使流分组统计依赖于交换机之间的时间同步，且恶意节点可以通过篡改分组的方式规避此类基于分组转发状态统计的检测方法，攻击者也可通过丢弃一定数量的分组并伪造相同数量分组的方式实施攻击，控制器难以检测此类攻击。现有 SDN 转发验证机制通过为数据平面开发新的安全通信协议实现分组逐跳验证转发，但这类机制因插入额外的分组验证字段引入了不小的计算与通信开销[6]。

为解决控制器难以捕获上述恶意节点针对分组攻击的问题，同时避免嵌入额外的密码标签字段引入的计算与通信开销，通过引入有限的计算与通信开销实现高效分组转发，并有效检测分组传输过程中恶意的分组注入/篡改、丢弃/劫持攻击，本文提出一种基于地址重载的 SDN 分组转发验证 AO-PFV（address overloading-based packet forwarding verification）机制，该机制利用 SDN 集中控制、可编程的特点，克服了现有机制通过为数据平面开发新的安全通信协议并实现分组逐跳转发的缺点，通过地址重载，AO-PFV 实现了高效的分组转发，基于采样，控制器能有效检测分组转发的异常行为。

1　相关研究

基于加密或签名的方法在传统IP网络的分组转发验证中已被广泛应用，这类方法通常需要部署密码基础设施，通过为交换机或路由设备开发新的安全通信协议，在分组头中嵌入密码标签实现逐跳的转发验证[7-8]，分组传输中的验证引入了较大的运算开销、嵌入固定或可变长度的密码标签也极大地增加了网络通信开销。现有的 SDN 分组转发验证大多是通过引入传统 IP 网络的数据转发验证机制来实现的，如为数据平面交换机开发新的安全通信协议，向转发的分组头中嵌入可变或固定长度的密码标签，通过校验分组头的密码标签验证分组的有效性。文献[9]提出了 SDN 路径一致性规则验证机制 REV（rule enforcement verification），通过向转发的分组头中嵌入分组摘要信息保证其完整性，同时，路径中各交换机更新基于压缩消息认证码的标签信息，防止恶意分组重定向，控制器基于嵌入的分组字段信息验证其是否通过了控制器授权的完整路径，REV 引入了不小的通信开销。文献[10]提出基于属性签名标识的 SDN 分组转发验证机制，根据用户的身份属性，采用属性密码算法生成属性签名标识，可有效验证分组篡改、伪造等异常行为，但采用基于属性签名的密码学算法使复杂的验证过程引入了较大的计算开销和网络传输时延，且该机制未能有效应对恶意的丢弃与劫持攻击。LPV（lightweight packet forwarding verification）[11]中控制器向交换机下发校验表 LPV-table，各交换机节点基于 LPV-table 验证分组的完整性，利用 SDN 的消息机制以及组表读取转发节点的流转发统计信息实现定位异常，LPV 中交换机节点基于控制器下发的摘要信息验证分组完整性，其本质依然是采用逐跳验证的方式实现分组转发。文献[12]提出一种基于数据平面可编程的软件定义网络报文转发验证机制，通过为数据分组添加自定义的密码标识，将协议独立数据分组处理编程语言 P4 转发设备[13]加入基于 OpenFlow 的软件定义网络，对网络流采样，适用于小型网络的分组转发验证，机制在用户源端 IP 头中的选项字段嵌入自定义密码标识增加了网络通信开销，同样，该机制仅能检测恶意篡改/注入攻击。文献[14]基于流图抽象的概念，通过自定义算法处理网络更新和增量验证，检测网络拓扑和数据平面中已知或未知的安全威胁，对异常产生警报，该机制未能有效检测分组的篡改、延迟等恶意攻击。文献[15]提出了基于端址重载的转发验证机制，该机制通过重载分组端口和地址信息使分组传输中无任何额外的通信开销，可以有效检测传输中的篡改、丢弃等攻击，但机制引入的安全通信协议仍然采用了逐跳验证的方式，带来了一定的验证开销。表 1 对比分析了现有典型方案解决的问题、主要技术和各自的优缺点。

基于表 1 对比分析可以看出，传统网络或 SDN 现有转发验证机制为数据平面交换机开发新的安全通信协议，通过嵌入密码标签/标签实现分组的逐跳转发验证，或结合密码标签与分组抽样方式检测转发异常，需要额外分组头开销，引入了一定的计算和通信开销[7-12]，且未能有效检测恶意的丢弃/劫持攻击[7,10,12]；而通过统计路径中节点转发的数据分组计数值并检测异常仅仅可以检测网络中恶意节点针对分组的注入与丢弃攻击[14]，但不能有效应对恶意节点的篡改攻击。

2　攻击模型

SDN 逻辑集中的控制器下发转发策略，数据平面交换机遵循控制器授权策略转发数据。通常，存

表 1 **典型方案分析**

方案	解决的问题	主要技术	存在的优缺点
文献[7]	分组转发、路径验证	消息验证码、哈希链	嵌入的分组头随路径长度线性增加，难以抵御恶意节点的丢弃与劫持攻击
文献[8]	分组转发、路径验证	轻量级密钥分发、采样	协议通信开销大，源节点接收中间节点回传的采样信息易受干扰
文献[9]	传输路径一致性检测	压缩消息验证码	基于流的压缩验证码减小了通信开销，分组丢失时易发生误报，嵌入的分组头随路径线性增加
文献[10]	分组转发控制	属性密码学、基于密码标签转发	计算与通信开销大，可以检测恶意的篡改/注入，不能检测丢弃/劫持攻击
文献[11]	分组转发验证	消息验证码、基于特定的标签采样	插入的标签增加了通信开销，控制器下发分组验证码实现逐跳验证
文献[12]	分组转发验证	密码标识、采样	通信开销较小，可检测篡改，不能检测丢弃/劫持
文献[14]	网络异常检测	抽象流图	难以检测丢弃并注入或篡改等复杂攻击
文献[15]	分组转发验证	消息验证码、端址重载	通信开销较小，源节点需保存流路径信息

在非诚实的交换机或敌手控制某一交换机时，可以构造以下攻击来破坏数据分组转发，以中断主机或网络之间的通信。1) 注入分组，即攻击者注入虚假分组数据；2) 篡改分组，即攻击者篡改数据分组，如负载数据；3) 丢弃分组，即攻击者丢弃传输中的任意分组；4) 劫持分组，即攻击者将分组重定向至非控制器授权的路径节点；5) 重放分组，即攻击者重放以前的分组。攻击者可以实施上述某单一的攻击方式或若干攻击方式结合的组合攻击。

针对上述的攻击类型，本文目标是以较小的开销实现高效的分组转发，以较低的误报率（FPR, false positive rate）与漏报率（FNR, false negative rate），实时、准确检测数据流传输中存在的异常。

本文假定逻辑集中的控制器是安全的，任一数据流入口与出口交换机为诚实交换机节点，控制器与数据平面的通信信道是安全的，数据的机密性由终端用户保证。

3 地址重载的 SDN 分组转发验证 AO-PFV

本节详细阐述基于地址重载的 SDN 分组转发验证机制 AO-PFV 如何实现高效转发、实时检测异常的目标。

3.1 AO-PFV 概述

与现有机制通过开发新的安全通信协议并嵌入密码标签实现逐跳验证的方式相比，AO-PFV 通过地址重载与哈希采样实现分组转发验证、检测异常。AO-PFV 机制的控制器运行基本流程如图 1 所示。

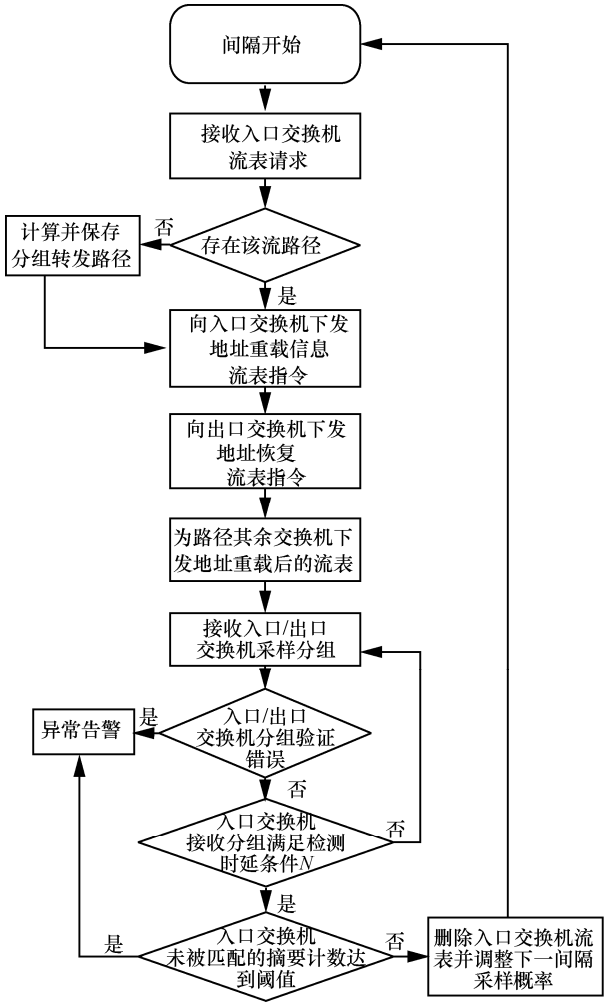


图 1 AO-PFV 控制器运行流程

具体来说，AO-PFV 是一种抽样检测机制，针对每条流的每个分组实施采样检测，但由于网络分组传输时延，流分组统计严重依赖于转发设备之间

严格的时间同步[6]，为此，AO-PFV 借鉴移动目标防御[16]（MTD, moving target defense）中地址跳变思想，设计一种地址重载技术，通过将分组头地址字段重载为不同的流表匹配标识，使控制器可将流 FlowSN 运行时间 t 分割为一系列连续的时间间隔 t_i，即 $t=t_1+t_2+\cdots+t_n$，t_i 依赖于入口交换机接收的分组数满足某一预设值 N（也称为检测时延），由于各时间间隔 t_i 采用的地址重载匹配字段不同，因而达到了时间分割的作用，解决了分组统计依赖交换机时间同步问题。

流在任一间隔 t_i 开始，控制器向入口交换机发送地址重载流表指令，向路径中各后继交换机发送地址重载后的流表，并向出口交换机发送地址恢复的流表指令；间隔 t_i 内，入口交换机基于该指令重载数据分组的地址信息，流路径中的各后继交换机基于重载后的流表转发数据分组，同时，控制器接收入/出口交换机采样的分组验证数据分组的完整性。若验证错误，则表明存在恶意注入/篡改攻击；若间隔 t_i 结束之后、下一间隔 t_{i+1} 开始之前，采样的入口交换机分组摘要未被验证的计数值超出了预设的阈值，表明存在丢弃/劫持攻击；若上述检测未发生验证错误，并且采样的入口交换机分组摘要未被验证的计数值低于预设的阈值（考虑由于网络传输中存在的自然分组丢失率），则表明在 t_i 内无恶意攻击，可减小下一间隔 t_{i+1} 采样概率，以此提高网络传输效率。以下将详细阐述 AO-PFV 中基于地址重载的分组转发和基于哈希采样异常检测机制。

3.2　基于地址重载的分组转发

当数据流进入第一跳入口交换机，入口交换机查询流表，若失配，表明是新的数据流或是该流下一个时间间隔 t_i 开始，此时，交换机将向控制器发送 Packet-In 消息请求流表，控制器根据网络拓扑信息计算或已保存的该流传输路径，向路径中的各交换机下发流表。控制器提取并保存入口交换机 Packet-In 消息中分组头中的协议号、源/目的地址、源/目的端口号五元组信息，按式(1)计算间隔 t_i 流标识号。

$$\mathrm{FlowSN}_i=\mathrm{Hash}(\mathrm{proto}\,\|\,\mathrm{srcip}\,\|\,\mathrm{dstip}\,\|\,\mathrm{sport}\,\|\,\mathrm{dport}\|t_i)_{32} \tag{1}$$

控制器生成间隔 t_i 标识 $\mathrm{TM}_i=\mathrm{Hash}\,(t_i)_{32}$，$i=1,2,\cdots,n$，向路径中的各后继交换机发送 Flow-Mod 消息，安装以 FlowSN_i 及时间间隔标识 TM_i 为匹配域的流表项，在入口交换机安装的流表项中通过指令（如 OpenFlow 中 SET_FIELD 标准命令）将分组头中的地址信息分别设置为 FlowSN_i 和 TM_i，实现分组地址信息重载，并设定分组的出端口，重载后的 IP 分组头地址字段如图 2 灰色部分所示，其中 FlowSN 为流标识符，TM 为时间间隔标识，每个间隔 t_i 重载的地址字段信息 FlowSN_i 和 TM_i 不同，用于流时间的分割及分组采样统计的时间同步。

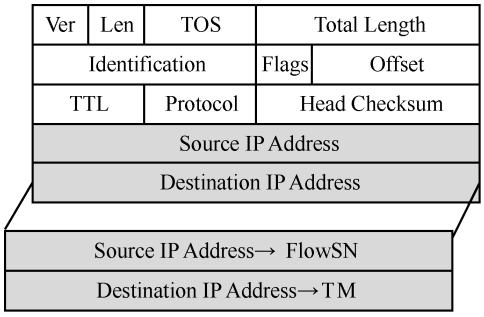


图 2　重载后的 IP 分组头地址字段

控制器获取入口交换机接收的分组数 N 且满足某一条件后，向入口交换机发送 Flow-Mod 消息修改并更新下一间隔 t_{i+1} 流表项，同时向路径中各后继交换机安装 t_{i+1} 间隔的流表项，并在一个往返时延后删除各后继交换机在间隔 t_i 内的流表项，使分组可以无损、可靠地传输[17]。通过地址重载，控制器将流运行时间分割为一系列连续的间隔 t_i，保证控制器对流路径中入口/出口交换机采样分组的时间同步，各后继交换机将根据控制器安装的流表转发数据。控制器保存数据流的原始地址信息，并在出口交换机安装的流表项中指定恢复分组原始地址信息指令，当流分组到达出口交换机时，出口交换机基于该流表指令恢复分组地址，并转发分组至终端。AO-PFV 中入口和出口交换机的分组处理流程分别如图 3 和图 4 所示。入口交换机接收流的原始分组，基于控制器下发的流表规则，重载分组地址字段，通过哈希采样，向控制器发送被采样的分组并转发分组至下一节点；中间交换机基于控制器下发的、地址重载后的流表规则转发分组；分组到达出口交换机时，出口交换机匹配分组并通过哈希采样向控制器发送被采样的分组，基于控制器流表指令恢复分组原始地址，转发分组至终端。

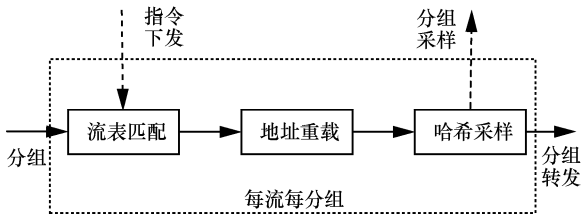


图 3　入口交换机分组处理流程

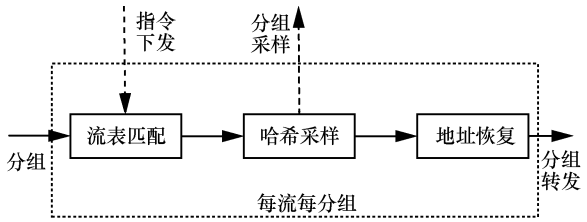


图 4　出口交换机分组处理流程

3.3　基于哈希采样的异常检测

对某一流 FlowSN，通过交换机逐跳验证所有的数据分组，增加了数据平面的计算与通信开销。AO-PFV 中控制器通过地址信息重载将流运行时间分割为连续的时间间隔 t_i，解决分组计数依赖交换机时间同步问题，控制器获取在间隔 t_i 内入口与出口交换机基于哈希采样[18-19]的转发分组并验证其完整性，检测流路径中存在的异常转发行为。第 2 节问题描述中分组的重放可等价于重复的分组注入攻击，分组的篡改等同于丢弃原分组并注入伪造的分组，分组的劫持攻击可等价于丢弃原路径上正常转发的分组；因此，本文主要考虑恶意节点实施的篡改以及丢弃这两类攻击。

控制器获取入口/出口交换机通过哈希采样的分组，基于如下的简单统计算法检测异常：设入口交换机在间隔 t_i 内接收的转发分组计数为 N（N 为一足够大的值），基于相同的哈希采样算法，控制器获取入口与出口交换机以相同概率 λ 采样的转发分组，对比出口交换机采样分组与入口交换机采样分组的消息摘要，验证其完整性，检测网络异常。对于恶意的篡改攻击，若被篡改的分组在出口交换机被采样，则其摘要值必定不能与入口交换机所采样的分组摘要值相匹配，控制器可判定存在篡改攻击行为；对于恶意丢弃分组攻击，被丢弃的分组必定不能被出口交换机所采样，令间隔 t_i 内入口交换机采样的分组摘要中未被匹配的计数值为 ΔS，在无恶意攻击且链路无因拥塞等原因产生自然分组丢失的条件下，未被匹配的摘要计数值必定为 $\Delta S=0$，或者分组传输中存在一定概率的自然分组丢失率条件下，若计数值 $\Delta S>0$，但在某一可接受的阈值范围内，可认为分组未被恶意丢弃，分组转发正常，若未被匹配的摘要计数值 ΔS 大于某一阈值，表明数据流传输中存在恶意的丢弃行为。

AO-PFV 中入口/出口交换机按式(2)计算分组摘要值，并按式(3)采样时间间隔 t_i 内的分组。

$$\Phi = H(\mathrm{IP}_{\mathrm{INVAR}} \,\|\, \mathrm{FlowSN} \,\|\, \mathrm{TM} \,\|\, \mathrm{Payload}) \oplus K_{t_i} \tag{2}$$

$$\mathrm{hashsample}\,(\Phi) = \begin{cases} 1, & \Phi < \lambda_i 2^n \\ 0, & \Phi \geqslant \lambda_i 2^n \end{cases} \tag{3}$$

式(2)通过单向哈希函数 H（如 SHA-1）计算传输分组 IP 头中不变的字段 $\mathrm{IP}_{\mathrm{INVAR}}$、分组负载 Payload 及重载后地址信息字段（流标识 FlowSN 及间隔标识 TM）的摘要。其中，K_{t_i} 为间隔 t_i 的临时密钥，在 t_i 结束前，仅控制器与入口、出口交换机持有该密钥，$K_{t_i}=H(K\,\|\,\mathrm{TM}_i)$，这里 K 为数据流 FlowSN 入网时，控制器与入口及出口交换机的共享密钥，因此路径中的各交换机节点无法预知入口/出口交换机所采样的分组；Φ 为分组摘要值与 K_{t_i} 摩尔加。入口/出口交换机按式(3)实现分组哈希采样，$n=\mathrm{lb}n$，η 为 Φ 位长度，λ_i 为哈希采样概率，$0<\lambda_i<1$。AO-PFV 中入口、出口交换机以概率 λ_i 采样转发分组 P，控制器维护一个流入口交换机数据分组的消息摘要表 $\mathrm{Table}_{\mathrm{mac}}$，基于哈希采样的异常检测算法如算法 1 所示。

算法 1　基于哈希采样异常检测算法

输入　间隔 t_i 采样概率 λ_i，采样分组 P_i 和 P_o

输出　间隔 t_{i+1} 采样概率 λ_{i+1} 或异常告警

```
1) if 控制器接收到采样分组 then
2)   if 采样分组 P_i 来自入口交换机 then
3)     计算分组的摘要值 MAC_{P_i} 并存储于表 Table_mac
4)     更新入口交换机摘要计数值 size ++
5)   end if
6)   else if 采样分组 P_o 来自出口交换机 then
7)     计算其摘要值 MAC_{P_o} 并在 Table_mac 中查找该值
8)     if 查找成功 then
9)       更新成功匹配摘要计数值 S ++
10)      删除 Table_mac 中的 MAC_{P_i}
11)    else
```

12)　return 异常告警
13)　end if
14)　end if
15) end if
16)if 入交换机接收分组数 N 满足检测时延值 then
17)　计算 $\text{Table}_{\text{mac}}$ 未匹配的摘要计数 $\Delta S=\text{size}-S$
18)　if $\Delta S>0$ 且 $\Delta S>\lambda_i N(\theta+\Delta\theta)$ then
19)　return 异常告警
20)　else then
21)　更新 t_{i+1} 采样概率：$\lambda_{i+1}=\omega\lambda_i$
22)　删除摘要表 $\text{Table}_{\text{mac}}$
23)　return　success
24)　end if
25) end if

算法 1 中采样的分组来自入口交换机时，计算并存储该分组的消息摘要 MAC_{P_i}，$\text{MAC}_{P_i}=H(\text{IP}_{\text{INVAR}}\|\text{FlowSN}\|\text{TM}\|\text{Payload})$，更新采样分组计数值 size；采样的分组 P_o 来自出口交换机的转发分组时，计算该分组的消息摘要 MAC_{P_o}，并在 $\text{Table}_{\text{mac}}$ 中查找是否存在与此匹配的摘要值，若成功匹配，表明该分组同时被入口与出口交换机采样且被正确传输，此时将更新成功匹配成功计数值 S（第 1)～9)行），间隔 t_i 内，$\text{Table}_{\text{mac}}$ 中已被匹配成功的摘要值将不再参与下一次匹配（第 10)行）；若所采样的出口交换机分组摘要 MAC_{P_o} 未能与表 $\text{Table}_{\text{mac}}$ 中某一 MAC_{P_i} 匹配，表明流分组传输路径中必存在虚假分组的注入或分组篡改攻击行为（第 11)～13)行）。当 FlowSN 入口交换机接收的分组数 N 满足检测时延条件时（第 16)～25)行），控制器将根据在 t_i 内所采样的入口交换机摘要计数值 size、成功匹配的摘要计数值 S，计算在 $\text{Table}_{\text{mac}}$ 中未被匹配的计数值 $\Delta S=\text{size}-S$，检测分组传输中的丢弃或劫持攻击，根据检测结果调整下一间隔 t_{i+1} 采样概率 λ_{i+1} 或异常告警（第 18)～24)行）。设入口交换机在 t_i 内接收并转发的分组数为 N，若未被匹配的摘要计数值 $\Delta S>0$，且其值 ΔS 大于某一阈值 $\lambda_i N(\theta+\Delta\theta)$（第 18)行），$\theta$ 为无恶意攻击时路径中的自然分组丢失率，$\Delta\theta$ 为可调节参数，$0<\Delta\theta<\theta$，表明路径中存在恶意的丢弃或劫持的攻击；否则，表明 t_i 间隔内路径中无恶意转发行为，删除保存的该间隔内采样分组的摘要表，并减小下一时间间隔 t_{i+1} 的采样概率 λ_{i+1}，即 $\lambda_{i+1}=\omega\lambda_i$，其中 $0<\omega<1$。

定理 1　算法 1 在任意时间间隔 t_i 内，哈希采样概率为 λ，自然分组丢失率为 θ 的流分组传输路径上，使在无恶意丢弃攻击时的误报率 FPR 以及当存在以大于概率 θ 对分组实施丢弃时的漏报率 FNR 皆低于设定值 $\varepsilon(0<\varepsilon<1)$，入口交换机接收的分组数 N 必须满足 $N\geqslant\max\left\{\dfrac{4\theta}{(\theta-\Delta\theta)^2\lambda}\ln\dfrac{1}{\varepsilon},\dfrac{3\theta}{\Delta\theta^2\lambda}\ln\dfrac{1}{\varepsilon}\right\}$，在此检测时延条件下，给定漏报率 ε，当恶意节点注入/篡改分组数 $x>\dfrac{\ln\varepsilon}{\ln(1-\lambda)}$ 时，控制器可以 $1-\varepsilon$ 的概率检测到该类攻击。

证明　以下分别考虑恶意节点的篡改以及丢弃攻击检测。

1) 恶意节点的篡改攻击。在间隔 t_i 内，节点篡改单个分组且不被出口交换机所采样的概率为 $1-\lambda$，若节点注入或篡改 x 个分组，且皆未被采样，此时恶意节点可有效规避控制器检测，检测算法将发生漏报（控制器检测节点的注入或篡改不会发生误报），FNR 满足

$$P_{\text{FNR}}=\varepsilon=(1-\lambda)^x \tag{4}$$

因此，在给定的漏报率 ε 的条件下，恶意节点最多可注入/篡改的分组数为

$$x\leqslant\frac{\ln\varepsilon}{\ln(1-\lambda)} \tag{5}$$

所以，当节点注入/篡改 $x>\dfrac{\ln\varepsilon}{\ln(1-\lambda)}$ 个分组时，控制器可以 $1-\varepsilon$ 的概率检测到恶意篡改攻击。

2) 恶意节点的丢弃攻击。当流 FlowSN 路径中无恶意节点的丢弃攻击且无自然分组丢失时，控制器按式(3)获取入口与出口交换机以概率 λ 哈希采样的数据分组，验证成功的计数值应为 $S=\lambda N$，未能匹配的计数值则为 $\Delta S=0$（N 为入口交换机在 t_i 内接收并转发的分组数）。考虑网络中存在的自然分组丢失率，假定数据传输路径中因网络拥塞等原因使分组在传输中以最大概率 θ 被自然丢弃。

当无恶意攻击时，基于哈希采样，控制器获取的出口交换机的采样分组摘要未能与入口交换机

采样的分组摘要匹配的计数值为 ΔS，$\Delta S = \text{size} - S$，其期望值应满足 $\mathrm{E}(\Delta S) \leqslant \lambda N\theta$，若 $\Delta S > \text{thr}$，这里取阈值 $\text{thr} = \lambda N(\theta + \Delta\theta) = \lambda N\theta + \lambda N\Delta\theta > \mathrm{E}(\Delta S)$，其中 $\Delta\theta$ 为可调节参数，$0 < \Delta\theta < \theta$，此时必将引发误报（FP, false positives）。异常检测算法使在无恶意丢弃攻击时的 FPR 满足

$$\begin{aligned} P_{\text{FPR}} = & P\left[\Delta S > \text{thr} \middle| \mathrm{E}(\Delta S) \leqslant \lambda N\theta\right] = \\ & P\left[\Delta S > \lambda N(\theta + \Delta\theta) \middle| \mathrm{E}(\Delta S) \leqslant \lambda N\theta\right] = \\ & P\left[\Delta S > \lambda N\theta(1 + \frac{\Delta\theta}{\theta}) \middle| \mathrm{E}(\Delta S) \leqslant \lambda N\theta\right] \end{aligned} \tag{6}$$

基于切诺夫界[20]可得

$$P_{\text{FPR}} \leqslant \mathrm{e}^{-\frac{\lambda N\theta}{3}\left(\frac{\Delta\theta}{\theta}\right)^2} = \mathrm{e}^{-\frac{\Delta\theta^2}{3\theta}\lambda N} \tag{7}$$

其中，$0 < \Delta\theta < \theta$。

使在无恶意丢弃攻击时误报率低于某一定值 ε，需要满足

$$N \geqslant \frac{3\theta}{\Delta\theta^2\lambda}\ln\frac{1}{\varepsilon} \tag{8}$$

同理，当路径中存在恶意攻击者以大于 θ 的概率对分组实施攻击时，考虑存在的最大自然分组丢失率 θ，则 $\text{Table}_{\text{mac}}$ 中未被匹配计数值 $\Delta S = \text{size} - S$，其期望值满足 $\mathrm{E}(\Delta S) > 2\lambda N\theta$，若此时 $\Delta S < \text{thr}$，$\text{thr} = \lambda N(\theta + \Delta\theta) < 2\lambda N\theta < \mathrm{E}(\Delta S)$，则必将引发漏报（FN, false negatives），控制器异常检测算法使当存在以概率 θ 实施恶意丢弃攻击时的漏报率 FNR 满足

$$\begin{aligned} P_{\text{FNR}} = & P\left[\Delta S < \text{thr} \middle| \mathrm{E}(\Delta S) > 2\lambda N\theta\right] = \\ & P\left[\Delta S < \lambda N(\theta + \Delta\theta) \middle| \mathrm{E}(\Delta S) > 2\lambda N\theta\right] = \\ & P\left[\Delta S < 2\lambda N\theta\left(1 - \frac{\theta - \Delta\theta}{2\theta}\right) \middle| \mathrm{E}(\Delta S) > 2\lambda N\theta\right] \end{aligned} \tag{9}$$

基于切诺夫界可得

$$P_{\text{FNR}} \leqslant \mathrm{e}^{-\frac{2\lambda N\theta}{2}\left(\frac{\theta - \Delta\theta}{2\theta}\right)^2} = \mathrm{e}^{-\frac{(\theta - \Delta\theta)^2}{4\theta}\lambda N} \tag{10}$$

使存在攻击者以大于 θ 的概率对分组实施丢弃攻击时漏报率低于某一定值 ε，需要满足

$$N \geqslant \frac{4\theta}{(\theta - \Delta\theta)^2\lambda}\ln\frac{1}{\varepsilon} \tag{11}$$

若使在无恶意丢弃攻击时检测误报率 FPR 以及存在以大于 θ 的概率对分组实施丢弃攻击时检测算法漏报率皆低于某一定值 ε，联立式(8)及式(11)可得

$$N \geqslant \max\left\{\frac{4\theta}{(\theta - \Delta\theta)^2\lambda}\ln\frac{1}{\varepsilon},\ \ \frac{3\theta}{\Delta\theta^2\lambda}\ln\frac{1}{\varepsilon}\right\} \tag{12}$$

其中，$0 < \Delta\theta < \theta$，$0 < \varepsilon < 1$。

综上可知，在任意间隔 t_i 内，对于丢弃攻击检测，若流分组传输路径中最大自然分组丢失率为 θ，在入口交换机接收的数据分组 N 在满足式(12)条件下，无恶意丢弃时的 FPR 以及存在以大于 θ 的概率恶意丢弃分组时的 FNR 低于设定值 ε；而对于恶意注入/篡改检测，在式(12)检测时延下，当节点注入/篡改的分组数 $x > \dfrac{\ln\varepsilon}{\ln(1-\lambda)}$ 时，控制器可以 $1-\varepsilon$ 的概率检测到恶意攻击。因此，算法 1 基于哈希采样检测算法可以有效检测分组传输中的恶意攻击行为，证毕。

4 实验与分析

本节构建一个简单的仿真网络环境，通过扩展交换机行为模型（BMV2, behavioral-model version 2）中 ExternType 类实现基于哈希采样的 P4 可编程交换机，评估 AO-PFV 机制的有效性，内容包括异常检测准确度以及机制网络性能，最后简要分析 AO-PFV 与现有类似机制的优缺点。

4.1 实验环境

实验采用联想 Lenovo E580 作为运行平台，操作系统运行 64 位 Ubuntu 14.06，配置为 Inter(R) Core(TM)i7-8550 CPU、8 GB 内存、1.8 GHz，使用 Mininet 模拟网络环境，可编程 P4 软件交换机、基于 P4Runtime 接口实现的控制器相关组件，本文扩展了交换机行为模型 BMV2 使其支持数据分组哈希采样，通过 p4c 编译器编译 P4 程序生成 JSON 格式描述文件并导入 BMV2 交换机行为模型，模拟网络采用分布式拓扑，由 20 个虚拟 P4 交换机及若干虚拟主机终端组成。

4.2 异常检测准确度

实验分别选择路径长度 L=8 和 L=10 的两条简单路径 $\text{PATH}_1 = (\text{R}_0, \text{R}_1, \cdots, \text{R}_8)$、$\text{PATH}_2 = (\text{R}_0, \text{R}_1, \cdots, \text{R}_{10})$。恶意节点 R_5 实施数据篡改、丢弃两类攻击，本节实验用于检验定理 1 在理论设定的误报与漏报率 ε 条件下，基于哈希采样分组的异常检测算法的有效性。以下将主要评估针对恶意的篡改、丢弃这两类攻击实施检测的有效性，验证其在恶意节点实施篡改攻击时检测算法的准确性、在无恶意丢弃攻击时的 FPR 以及存在恶意丢

弃攻击时的 FNR。

仿真网络传输路径 $PATH_1$ 最大自然分组丢失率 $\theta_1 = 0.008$，调节参数 $\Delta\theta_1 = \frac{1}{2}\theta = 0.004$；传输路径 $PATH_2$ 最大自然分组丢失率 $\theta_2 = 0.010$，调节参数 $\Delta\theta_2 = \frac{1}{2}\theta_2 = 0.005$。

4.2.1　恶意篡改 FNR

实验验证在预设漏报率 $\varepsilon = 0.03$ 下，恶意节点实施篡改攻击时检测算法的实际漏报率。通过网络性能测试工具 iperf 持续向入口交换机节点 R_0 发送分组数据。路径长度 L=8 和 L=10，采样概率 $\lambda = 0.10$ 和 $\lambda = 0.20$，攻击者 R_5 以概率 $\alpha = 0.02\%$～0.08% 对分组实施篡改攻击时实际漏报率分别如图 5 和图 6 所示。

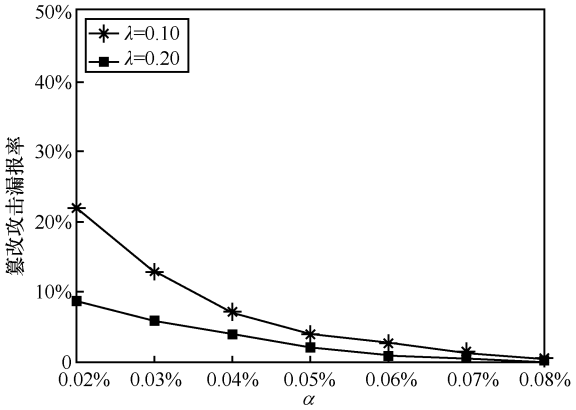


图 5　L=8 时不同篡改率下的漏报率

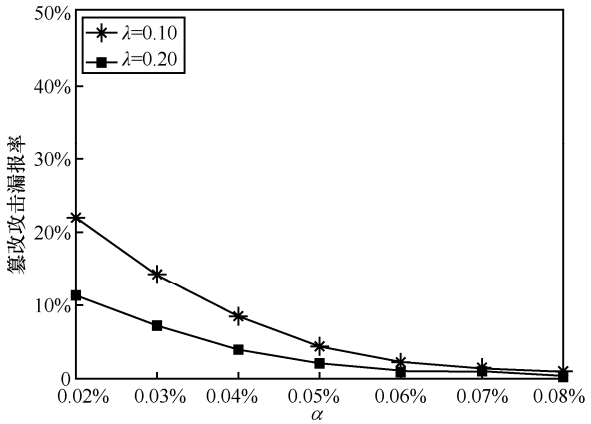


图 6　L=10 时不同篡改率下的漏报率

由图 5 和图 6 可以看出，在节点以 $\alpha = 0.02\%$～0.03% 的极小概率实施篡改攻击时，检测算法存在 8%～22%的漏报，随着节点攻击概率的提高，在以 $\alpha = 0.05\%$～0.06% 实施篡改攻击时，存在约 4%的漏报率，在以 $\alpha = 0.08\%$ 实施攻击时，检测算法准确率达到 99%以上，漏报率低于 1%。此外，图 5 与图 6 是在相同的检测时延 N 下所得到的检测结果，可以看出，在保持 N 恒定的条件下，通过提高采样概率 λ，可以减小篡改攻击的漏报率。

4.2.2　无恶意丢弃攻击时的误报率 FPR

路径长度 L=8 和 L=10，采样概率 $\lambda = 0.10$～0.55，设定在无恶意丢弃攻击时理论误报率分别为 $\varepsilon = 0.05$ 和 $\varepsilon = 0.03$。实验结果如图 7 所示。从图 7 可以看出，由于检测阈值的影响，在无恶意丢弃攻击时，检测算法存在一定概率的误报，实际误报率低于 1.0%，均低于预设值 ε。

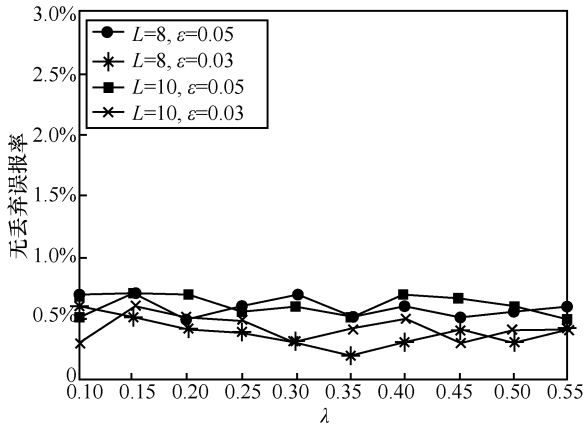


图 7　无恶意丢弃攻击时的误报率

4.2.3　恶意丢弃攻击时的漏报率

实验验证在预设恶意丢弃攻击理论漏报率分别为 $\varepsilon = 0.05$、$\varepsilon = 0.03$，采样概率为 $\lambda = 0.10$，$\lambda = 0.15$，$\lambda = 0.20$ 及 $\lambda = 0.25$ 时的实际漏报率。攻击者 R_5 分别以概率 $\alpha = 0.1\%$～1.0% 对分组实施丢弃，路径长度 L=8 和 L=10 的实验结果分别如图 8 和图 9 所示。

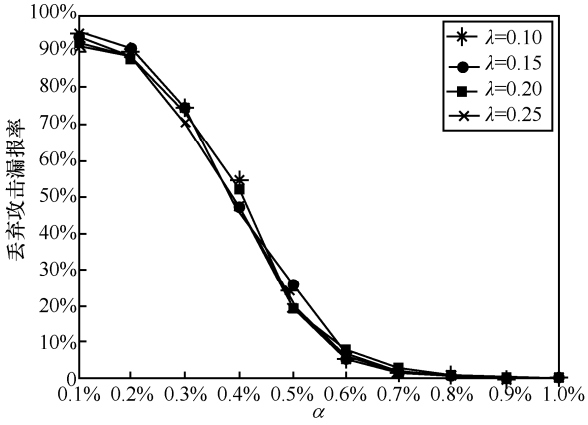


图 8　L=8, ε=0.03 时丢弃攻击漏报率

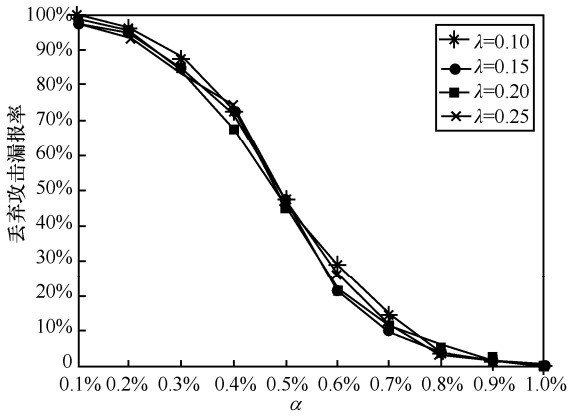


图 9　L=10, ε=0.05 时丢弃攻击漏报率

从图 8 和图 9 可以看出，在恶意节点在以极小概率$\alpha \ll \theta$、$\alpha = 0.1\%$～0.2% 对分组实施丢弃攻击时，检测算法漏报率达到 90%左右；随着攻击概率的增大，检测算法漏报率随之降低，当攻击概率$\alpha < \theta$、$\alpha = 0.5\%$～0.7% 时，漏报率减小为 5%～20%；当攻击者丢弃概率$\alpha \geqslant \theta$（L=8 时，$\alpha \geqslant 0.8\%$；L=10 时，$\alpha \geqslant 1\%$），漏报率为 0.1%～0.5%。

由上述分析可知，在理论设定的误报与漏报率ε（$\varepsilon = 0.03$，$\varepsilon = 0.05$）条件下，当攻击者以极小概率（$\alpha \ll \theta$）实施丢弃攻击时，受检测阈值影响，检测算法漏报率较高；当路径中无恶意丢弃攻击时，受网络自然分组丢失率影响，检测算法依然存在低于 1%平均误报率；而攻击者以$\alpha \geqslant \theta$的概率对分组实施丢弃攻击时，存在低于约 0.5%的漏报率。因此，在无恶意丢弃时的误报率以及存在以$\alpha \geqslant \theta$的概率实施丢弃时的漏报率皆低于理论预设值ε，这一实验结果与定理 1 相吻合。

由于检测阈值和网络自然分组丢失的影响，当阈值设定过低时（减小调节参数$\Delta\theta$），会增加无恶意丢弃攻击时的检测误报率，减小存在丢弃攻击时的漏报率；相反，当阈值设定过高时（增大调节参数$\Delta\theta$），则会减小在无恶意丢弃攻击时的误报率，而增加存在攻击时的漏报率，因此，检测算法可以减小但无法也不可能同时避免针对丢弃攻击检测的这两类误差。而对于恶意的篡改攻击，在保持检测时延 N 一定的情况下，可以通过增加采样概率，以此减小篡改攻击漏报率。

4.2.4　检测时延

基于哈希采样分组异常检测算法根据上一间隔t_i的检测结果，调节下一间隔t_{i+1}的采样概率λ。图 10 为不同采样概率下，哈希采样分组检测算法的检测时延（即分组数）。

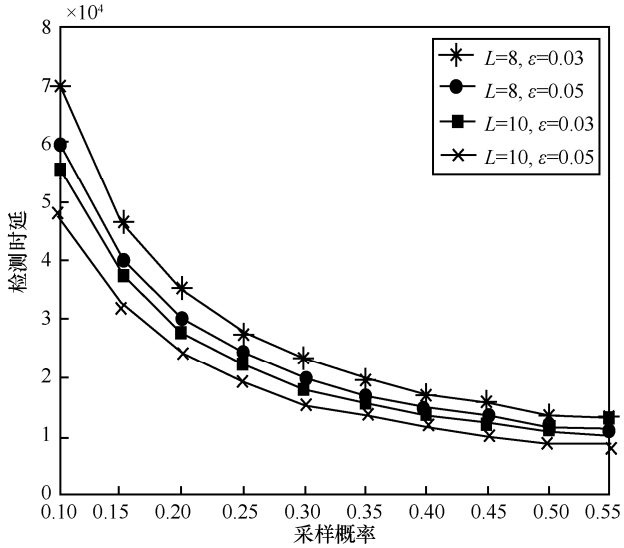


图 10　不同采样概率下的检测时延

从图 10 可看出，当 L=10、采样概率$\lambda = 0.50$、$\varepsilon = 0.05$以及$\varepsilon = 0.03$时，N=10 000～12 000。N为一个时间间隔内，流入口交换机所接收的分组数，由式(12)及图 10 可知，检测时延受采样概率及预设ε值影响，采样概率越小，检测时延越大，采样概率越大，则检测时延越小；ε值越大，检测时延越小。

4.3　性能评估

本节主要评估机制 AO-PFV 网络性能，包括计算开销、分组转发时延以及网络吞吐率等。

计算开销。数据平面交换机计算开销是影响转发时延的主要因素，与本文相似方案单次分组传输数据平面节点计算复杂度如表 2 所示。

在长度为 L 的路径上，文献[7,9]方案中路径各节点需两次消息认证码计算（用 H 表示），其总的计算开销为分别为 4LH 和 2LH；文献[10]方案中的任一分组的传输，源端需要嵌入基于身份属性密码的标签消息，其总开销至少为 2KP（K为属性元个数，P 为双线性对运算）；文献[11]方案中各节点需一次摘要计算（用 M 表示），总计算开销为 LM；文献[12]方案中一次分组传输，用户终端及 P4 交换机各需一次消息认证码计算，总的开销为 3H；文献[15]方案中入口交换机需 LH 次运算，各后继交换机各需一次认证码验证计算，总的开销为 2LH，而 AO-PFV 的一次分组传输，入口/出口交换机各需要一次摘要运算，总的运算开销为 2M。AO-PFV 一次分组传输节点计算开销低于现有相似机制。

表 2　　相似方案计算复杂度

方案	源端	中间节点	目的端	总开销
文献[7]	*L*H	2H	*L*H	4*L*H
文献[9]	2H	2H	2H	2*L*H
文献[10]	*K*P	—	*K*P	2*K*P
文献[11]	M	M	M	*L*M
文献[12]	H	—	H	3H
文献[15]	*L*H	H	H	2*L*H
AO-PFV	M	—	M	2M

4.3.1 转发时延

本节实验对比测试了运行 AO-PFV 协议及未运行 AO-PFV 协议即标准协议的网络性能，测试其在不同路径长度下的转发验证往返时延（RTT, round trip time），同时测试了在相同网络仿真环境中表 2 中节点计算开销较小的机制[12]和开销最大的机制[10]的往返时延，测试结果如图 11 所示。

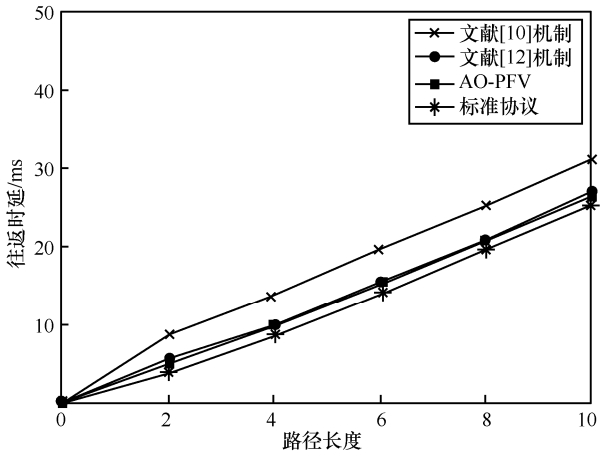


图 11　不同路径长度下的转发时延

图 11 中 *L*=10 时，运算标准协议平均往返时延约为 26 ms，AO-PFV 的平均往返时延低于 28 ms，AO-PFV 平均引入了 7%～8%的单向网络转发时延，文献[12]机制略高于 AO-PFV，其引入了 9%左右转发时延，AO-PFV 远低于文献[10]机制，*L*=10 时，其时延约为 32 ms，平均引入了 19%～20%的时延。

4.3.2 网络吞吐率

本节实验对比测试了运行 AO-PFV 协议及未运行 AO-PFV 协议也即标准协议、计算开销较小的文献[12]机制和开销最大的文献[10]机制在分组负载为 300～1 200 B 下的网络吞吐率，*L*=8 和 *L*=10 的实验结果分别如图 12 和图 13 所示。在同一负载条件下，文献[12]机制和文献[10]机制吞吐率损失分别约为 16%和 8%，而 AO-PFV 损失 6%～7%的吞吐率。

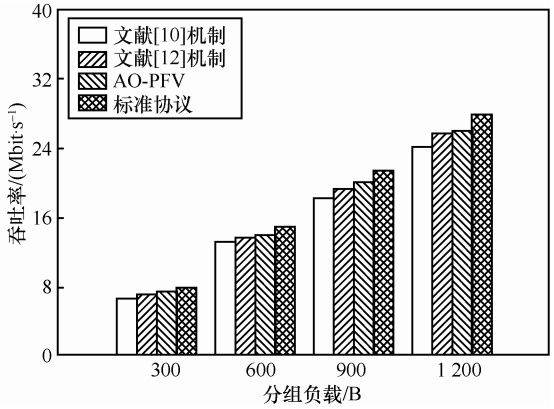


图 12　*L*=8 时不同负载下的网络吞吐率

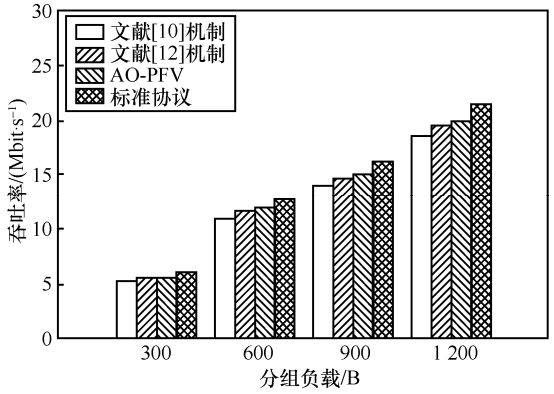


图 13　*L*=10 时不同负载下的网络吞吐率

4.4 机制比较

本节分别从定性与定量分析两方面对比分析 AO-PFV 与现有典型的机制之间的差异。

现有机制旨在解决 SDN 中数据平面分组转发验证，如表 3 所示，将 AO-PFV 与这几类机制相比较，从所提供的安全检测功能、有无分组头开销及以及数据平面是否需要额外的安全通信协议几方面定性比较分析其各自的优缺点。

文献[7]中节点执行两次验证码操作实现分组验证与路径验证域哈希链更新，文献[9]通过压缩的消息认证码实现分组路径一致性验证，文献[10]机制嵌入了基于属性密码的身份属性标签实现验证，文献[7,9-10]引入了新的安全通信协议以支持数据平面分组转发验证，可以有效检测分组的恶意的篡改、注入攻击，但无法应对恶意的丢弃攻击；LPV[11]为数据分组插入了一系列用于混淆攻击者的随机标签并采样某一类特定随机标签的分组，验证其完整性以此检测网络异常，该机制通过控制器下发分组摘要至交换机节点，采用的依然是逐跳验证的方式，可以检测分组的篡改与丢弃攻击，但插入的随

表 3　　**相似机制定性对比分析**

方案	篡改	丢弃	注入	无额外通信协议	无分组头开销
文献[7]	√	×	√	×	×
文献[9]	√	×	√	×	×
文献[10]	√	×	√	×	×
文献[11]	√	√	√	√	×
文献[12]	√	×	√	×	×
文献[15]	√	√	√	×	√
AO-PFV	√	√	√	√	√

机标签同样引入了一定的通信开销；文献[12]通过用户终端在 IP 头选项字段嵌入 36 B 的自定义密码标识，能有效检测恶意的篡改攻击，无法检测丢弃攻击。文献[15]可以有效检测针对分组的各类攻击，不需要额外的分组通信开销，但采用了逐跳验证的转发方式，增加了计算开销。

IP 分组负载为 800 B，L=8 时，各机制在分组头通信开销、数据平面分组单向传输计算时间、恶意分组篡改漏报率、丢弃检测时的漏报率与无丢弃攻击时的误报率定量分析对比如表 4 所示。在本文实验平台环境中一次消息摘要运算约为 6 μs，一次消息认证码计算约为 10～12 μs，一次双线性对计算约为 0.50～0.60 ms。

文献[7,9]中嵌入的分组头开销随着路径长度的增加而线性增加，分组头开销长度分别为 52+16L、32+L，单位为 B；文献[10]机制嵌入了基于身份属性密码的验证标签，其分组头开销至少为 144 B，这三类机制采用消息验证码或签名验证的方式转发分组，篡改攻击检测漏报率为 0，但无有效的丢弃攻击检测机制。LPV[11]同样插入了一系列用于混淆攻击者的随机标签，并采样某一类特定标签的分组，该机制随采样因子的不同，其篡改漏报率低于 10%，丢弃检测误报与漏报率低于 4%。文献[12]在 IP 选项字段插入了固定长度为 36 B 的密码验证标识，在攻击者以 10%的概率实施篡改攻击时，漏报率低于 10%，但该机制不能检测恶意的丢弃攻击。文献[15]不需要额外分组头通信开销，在攻击者以 1%～4%的概率实施篡改时，其漏报率低于 1%，在

表 4　　**相似机制通信开销、计算时间、漏报率与误报率分析（L=8，θ=0.008）**

方案	通信开销	计算时间/μs	篡改攻击		丢弃攻击		
			α	漏报率	α	漏报率	误报率
文献[7]	23%	300～380	—	0	—-	—	—
文献[9]	4.58%	160～190	—	0	—	—	—
文献[10]	14.4%	3 800～4 000	—	0	—	—	—
文献[11]	0.2%～0.5%	50	10%～30%	4%～10%	10%	2%	3%～4%
文献[12]	4.14%	30～36	10%	7%～10%	—	—	—
文献[15]	0	160～190	1%～4%	<1%	<0.5% 1%～4%	> 10% < 2%	1%～2%
AO-PFV	0	10～12	0.02%～0.05% 0.06%～0.07% > 0.08%	5%～15% 1% < 0.5%	<0.5θ 0.5 θ～θ >θ	40%～90% 10%～30% <1%	0.3%～1%

攻击者以小于 0.5%的概率实施丢弃攻击时，漏报率大于 10%，在攻击者以 1%～4%的概率丢弃分组时，其漏报率低于 2%，误报率低于 2%。AO-PFV 在攻击者大于 0.08%的概率实施篡改时的漏报率低于 0.5%，而以大于路径自然分组丢失率 θ 的概率实施丢弃攻击时，漏报率低于 1%，在无恶意丢弃攻击时的误报率低于 1%。

基于以上分析可知，通过入口交换机实现地址信息重载，AO-PFV 不需要额外的数据平面安全通信协议及额外的分组头开销，其分组通信开销与计算开销小于现有机制，基于哈希采样，控制器可以有效检测恶意节点的注入、丢弃、篡改等攻击。AO-PFV 机制与逐跳验证机制相比，在恶意节点实施篡改/丢弃攻击时，存在一定概率的漏报或误报，但在同等攻击概率条件下，AO-PFV 检测准确性优于现有机制。

5　结束语

针对现有 SDN 转发验证机制通过开发新的安全通信协议并嵌入密码标签实现逐跳验证引入较大的计算与通信开销的缺点，本文提出一种基于地址重载的分组转发验证机制 AO-PFV，AO-PFV 中入口交换机重载 IP 分组头中的地址信息将流运行时间划分为连续随机的时间间隔，后继节点基于重载的地址信息转发分组，控制器获取间隔内流入口及出口交换机基于哈希采样的转发分组并验证其完整性，检测路径中存在的异常转发行为；最后，构建仿真网络实现了所提机制并对其有效性进行了评估。与现有机制相比，AO-PFV 计算开销小，不需要额外安全通信协议和分组头开销，以引入不超过 8%的转发时延和约 7%的吞吐率损失，可有效检测分组转发中恶意注入、篡改、丢弃等异常转发行为。

参考文献：

[1] SARASWAT S, AGARWAL V, GUPTA H P, et al. Challenges and solutions in software defined networking: a survey[J]. Journal of Network and Computer Applications, 2019, 141: 23-58.

[2] 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.

[3] 岳猛, 王怀远, 吴志军, 等. 云计算中 DDoS 攻防技术研究综述[J]. 计算机学报, 2020, 43(12): 2315-2336.
YUE M, WANG H Y, WU Z J, et al. A survey of DDoS attack and defense technologies in cloud computing[J]. Chinese Journal of Computers, 2020, 43(12): 2315-2336.

[4] MIZRAK A T, CHENG Y C, MARZULLO K, et al. Detecting and isolating malicious routers[J]. IEEE Transactions on Dependable and Secure Computing, 2006, 3(3): 230-244.

[5] AKHUNZADA A, GANI A, ANUAR N B, et al. Secure and dependable software defined networks[J]. Journal of Network and Computer Applications, 2016, 61: 199-221.

[6] SHAGHAGHI A, KAAFAR M A, BUYYA R, et al. Software-defined network (SDN) data plane security: issues, solutions, and future directions handbook of computer networks and cyber security[J]. arXiv Preprint, arXiv: 1804.00262, 2018.

[7] KIM T H J, BASESCU C, JIA L M, et al. Lightweight source authentication and path validation[J]. ACM SIGCOMM Computer Communication Review, 2015, 44(4): 271-282.

[8] WU B, XU K, LI Q, et al. RFL: robust fault localization on unreliable communication channels[J]. Computer Networks, 2019, 158: 158-174.

[9] ZHANG P, WU H, ZHANG D, et al. Verifying rule enforcement in software defined networks with REV[J]. IEEE/ACM Transactions on Networking, 2020, 28(2): 917-929.

[10] 祝现威, 常朝稳, 朱智强, 等. 基于身份属性的 SDN 控制转发方法[J]. 通信学报, 2019, 40(11): 1-18.
ZHU X W, CHANG C W, ZHU Z Q, et al. SDN control and forwarding method based on identity attribute[J]. Journal on Communications, 2019, 40(11): 1-18.

[11] 王首一, 李琦, 张云. 轻量级的软件定义网络数据包转发验证[J]. 计算机学报, 2019, 42(1): 176-189.
WANG S Y, LI Q, ZHANG Y. LPV: lightweight packet forwarding verification in SDN[J]. Chinese Journal of Computers, 2019, 42(1): 176-189.

[12] 左志斌, 常朝稳, 祝现威. 一种基于数据平面可编程的软件定义网络报文转发验证机制[J]. 电子与信息学报, 2020, 42(5): 1110-1117.
ZUO Z B, CHANG C W, ZHU X W. A software-defined networking packet forwarding verification mechanism based on programmable data plane[J]. Journal of Electronics & Information Technology, 2020, 42(5): 1110-1117.

[13] 林耘森箫, 毕军, 周禹, 等. 基于 P4 的可编程数据平面研究及其应用[J]. 计算机学报, 2019, 42(11): 2539-2560.
LIN Y S X, BI J, ZHOU Y, et al. Research and applications of programmable data plane based on P4[J]. Chinese Journal of Computers, 2019, 42(11): 2539-2560.

[14] DHAWAN M, PODDAR R, MAHAJAN K, et al. SPHINX: detecting security attacks in software-defined networks[C]//Proceedings of 2015 Network and Distributed System Security Symposium. Virginia: the Internet Society, 2015: 1-15.

[15] 吴平, 常朝稳, 马莹莹. 基于端址重载的 SDN 包转发验证[J]. 通信学报, 2021, 42(7): 70-83.
WU P, CHANG C W, MA Y Y. Port address overloading based packet forwarding verification in SDN[J]. Journal on Commu-

nications, 2021, 42(7): 70-83.

[16] SENGUPTA S, CHOWDHARY A, SABUR A, et al. A survey of moving target defenses for network security[J]. IEEE Communications Surveys & Tutorials, 2020, 22(3): 1909-1941.

[17] JAFARIAN J H, AL-SHAER E, DUAN Q. Formal approach for route agility against persistent attackers[M]. Berlin: Springer, 2013: 237-254.

[18] DUFFIELD N G, GROSSGLAUSER M. Trajectory sampling for direct traffic observation[J]. IEEE/ACM Transactions on Networking, 2001, 9(3): 280-292.

[19] GOLDBERG S, XIAO D, BARAK B, et al. Measuring path quality in the presence of adversaries: the role of cryptography in network accountability[R]. 2007.

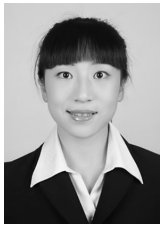
[20] HAGERUP T, RÜB C. A guided tour of Chernoff bounds[J]. Information Processing Letters, 1990, 33(6): 305-308.

[作者简介]

吴平（1979− ），男，安徽宿松人，信息工程大学博士生，主要研究方向为 SDN 安全、网络安全、数据平面编程。

常朝稳（1966− ），男，河南滑县人，博士，信息工程大学教授、博士生导师，主要研究方向为移动信息安全、物联网安全。

左志斌（1979− ），男，河南滑县人，博士，河南工业大学讲师，主要研究方向为网络安全。



马莹莹（1988− ），女，河南漯河人，信息工程大学博士生，主要研究方向为 SDN 安全、网络安全。