基于端址重载的 SDN 包转发验证

吴平，常朝稳，马莹莹
（信息工程大学密码工程学院，河南 郑州 450004）

摘 要：针对软件定义网络（SDN）现有转发验证机制因嵌入额外的分组字段所带来的通信开销大的问题，提出基于端址重载的包转发验证机制。其核心思想是入口交换机重构数据分组端口和地址信息实现端址重载，下游交换机基于重载的端址信息实现数据分组的概率验证，控制器统计路径中节点验证有效和无效的数据分组信息并定位异常；理论分析给出了恶意注入与丢弃攻击异常检测阈值；最后实现了所提机制并对其进行了评估。实验结果表明，所提机制以引入不超过 10%的转发时延、低于 8%的吞吐率损失实现高效转发及有效的异常定位。

关键词：软件定义网络；路径向量；端址重载；概率验证；异常定位
中图分类号：TP393
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021108

Port address overloading based packet forwarding verification in SDN

WU Ping, CHANG Chaowen, MA Yingying
Department of Cryptogram Engineering, Information Engineering University, Zhengzhou 450004, China

Abstract: Aiming at the problem that the existing forwarding verification mechanisms in software-defined networking (SDN) incur significant communication overhead caused by embedding additional packet fields, a packet forwarding verification mechanism based on port address overloading was proposed, which key idea was the ingress switch implemented port address overloading by reconstructing port and address of packet, downstream switches executed packet probabilistic verification based on overloading port address, and the controller acquired valid and invalid packet statistics of node verification in the path and localized anomaly. Anomaly detection threshold of malicious injecting and dropping packets was presented by theoretical analysis. Finally, the proposed scheme was implemented and evaluated. Experiments demonstrate the proposed scheme achieves efficient forwarding and effective anomaly localization with less than 10% of additional forwarding delays and less than 8% of throughput degradation.

Keywords: software-defined networking, path vector, port address overloading, probabilistic verification, anomaly localization

1 引言

软件定义网络（SDN, software-defined networking）起源于 Clean Slate 研究课题并于 2008 年由 Mckeown 教授[1]提出，与传统网络相比，SDN 将数据平面与控制平面分离，其体系结构分为应用平面、控制平面和数据平面。SDN 体系架构如图 1 所示，应用平面位于 SDN 架构的顶层，主要运行不同类型的业务和应用；控制平面包括逻辑中心化的控制器，基于应用平面请求产生网络规则，集中维护网络状态，一方面，实时维护全网的拓扑和状态信息，并为应用平面提供可扩展的编程接口，另一方面，通过与数据平面基础设施之间的接口获取底层设施信息，对数据平面的资源进行编排，通过

收稿日期：2021–02–04；**修回日期**：2021–03–31
基金项目：国家自然科学基金资助项目（No.61572517）
Foundation Item: The National Natural Science Foundation of China (No.61572517)

标准南向接口（如 OpenFlow 等[2]）向交换机下发数据转发规则；数据平面由网络底层的转发设备组成，主要负责数据的处理、状态收集并依规则转发数据。开放式网络基金会（ONF, open networking foundation）按照接口与控制平面的位置关系，分别定义了 SDN 架构中的南向接口和北向接口。南向接口定义了开放的 OpenFlow 协议标准，而北向接口由于应用平面中各类业务和应用的多样性与复杂性，目前尚无统一的规范和标准。

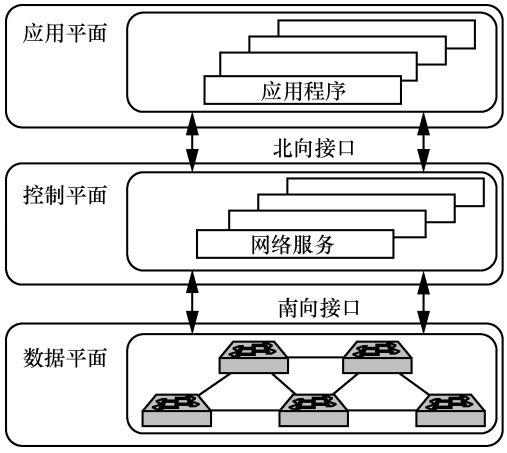


图 1　SDN 体系架构

SDN 以其灵活性、可编程性在为管理配置网络和快速部署新协议带来便利的同时，不可避免地产生了许多安全问题：应用层安全、北向接口安全、控制平面安全、南向接口安全、数据平面安全[3]，这些安全问题既包括传统网络安全问题，也包括 SDN 专有的安全问题，解决这些安全问题是未来部署和生产 SDN 的重要条件[3-4]。

当前，许多 SDN 安全问题的研究集中于控制平面[5-7]。然而，数据平面面临的安全威胁主要包括 DDoS（distributed denial of service）/DoS（denial of service）攻击、恶意/虚假流规则注入、非法访问、数据泄露、交换机自身的配置缺陷和身份假冒等，恶意的交换机能够在未授权的路径上重定向数据流，对数据流实施窃听、中间人攻击或者绕过安全中间设备，且能够针对其他设备实施 DoS 攻击、丢弃数据分组等攻击行为[3,8]。

在新兴的 SDN 中，交换机或路由器作为网络的核心设备，控制着数据分组的转发，恶意交换机可实施注入、丢弃、篡改或重放。因此，如何实现准确、高效的分组转发验证，保证分组转发的正确性并定位网络异常一直是被广泛研究的重要课题。

数据源认证与转发验证是缓解网络攻击（如 DoS 攻击、地址欺骗、数据流重定向）的有效机制[9]，使数据流路径与 ISP、企业或数据中心策略一致。传统 IP 网络数据源认证与转发验证典型方案主要方法包括通过标签或签名实现验证[10-11]、数据分组探测方法[12-13]以及通过计算交换机端口上的数据分组流量分析检测异常[14-15]等，这些方案或者在自治系统（AS, autonomous system）间通过 Diffifie-Hellman 计算共享密钥或者在源 AS 和目的 AS 间预先建立安全通道协商通信参数，因插入额外的数据分组头引入了巨大的通信与运算开销，且交换机或 AS 间交互式的通信参数与密钥协商也不适用于 SDN。

为解决 SDN 中现有的转发验证机制通过增加额外的分组验证字段而引入的通信与运算开销大的问题，利用 SDN 控制与转发分离、网络可编程等特性，提出基于端址重载的 SDN 包转发验证机制（PAO-PFV, port address overloading based packet forwarding verification）。本文贡献包括以下几点。

1) 提出一个基于端址重载的 SDN 包转发验证机制 PAO-PFV，通过重载端址，PAO-PFV 不产生任何额外的数据分组头开销，以较小的开销实现高效转发。

2) 针对 PAO-PFV 机制中恶意的注入与丢弃攻击，通过理论分析给出了有效的异常检测阈值。

3) 通过扩展可编程的协议无关报文处理（P4, programming protocol-independent packet processors）交换机模型[16]实现了 PAO-PFV，并在虚拟网络环境中对其性能进行了评估。

2　相关工作

SDN 中，为验证数据的来源、防止数据分组伪造，近年来，一些学者提出了许多优秀的方案。Yao 等[17]通过在控制器开发安全模块进行实时监控和检测，提出虚拟源地址验证边界（VAVE, virtual source address validation edge）安全框架来防止源 IP 地址欺骗，其在控制器中嵌入源地址验证模块对伪造地址进行过滤，边界交换机通过接收控制器安装的过滤规则实现敏捷灵活的源地址验证操作，但无法应对路径中恶意节点的篡改与丢弃攻击。Casado 等[18]提出了一个安全架构保护企业网络来自恶意交换机的攻击，但缺少数据流路径一致性验证机制，并将错误恢复机制交给终端用户。文献[19-20]依托 SDN 架构，将 SDN 交换机作为数据分组重定

向平台，其工作侧重于重定向网络流到安全设备，实现网络流量的分析、定位网络中的配置错误与网络调试。文献[21]提出了一种适用于 SDN 的高效的数据源认证与路径验证方案 OPT（origin and path trace），OPT 通过动态的密钥协商协议 DRkey（dynamically recreatable key），使用户终端与中间节点动态建立对称共享密钥，DRkey 需要额外的协议交互协商共享密钥，任一数据分组都需要源终端插入供后续节点的验证分组头，这带来了不小的通信开销，且该机制不能有效应对恶意节点实施的数据丢弃攻击。周启钊等[22]重点研究了 SDN 环境下动态的源地址配置技术。王首一等[23]提出了一个轻量级的数据分组转发验证机制 LPV（lightweight packet forwarding verification），该机制将数据流分为 2 类，一类是需要验证的数据流，另一类是不需要验证的数据流。LPV 任意分组都需要插入随机的标签信息且分组的验证都需要控制器干预，未能解决在敌对环境中恶意行为对中间节点的攻击。Dhawan 等[24]提出了一种基于流量分析的异常检测方法，通过对流图进行抽象，获取网络更新的实时信息和性能并检测对网络拓扑和数据转发已知和未知的威胁。该方法通过动态学习网络信息，对异常产生警报，然而这需要控制器记录所有流量的数量并实现在纳秒级内对这些数据进行比较，需要巨大的计算和通信开销。文献[25]提出了一个扩展的 SDNsec（SDN security extension）框架，SDNsec 中控制平面与数据平面中任一交换机共享对称密钥，控制器加密下发的流规则给路径上的源入口交换机，下游交换机通过共享密钥解密源交换机在分组头携带的流规则，依规则转发数据，同时，各交换机通过共享密钥计算消息认证码形成验证哈希链。SDNsec 存在的问题主要包括：数据流下游的交换机不能确认数据是否真实地来自源入口交换机；针对任一数据分组，目的出口交换机需向控制器发送 Packet-IN 消息，并最终由控制器验证路径的一致性，这必将大大增加控制器的负担；缺少有效的异常节点定位机制，未能抵御恶意节点的丢弃与劫持攻击。祝现威等[26]提出了基于属性基的访问控制方案 ACFlow，该方案通过修改终端用户网络协议栈并嵌入基于属性的标签，实现基于属性标签的转发验证，属性标签由基于身份属性的签名算法生成，任一数据分组都需源和目的交换机的签名验证，其缺点是数据分组头所占开销大，复杂的验证过程引入了巨大的网络传输时延，同样该机制未能有效应对恶意的丢弃与劫持攻击。

本文克服了现有机制中通过嵌入验证标签实现转发验证所引入的通信与运算开销大的问题，提出基于端址重载的包转发验证机制 PAO-PFV，其优点是不需要额外附加数据分组头即可实现高效的转发验证；此外，针对 PAO-PFV 中节点的注入与丢弃攻击，理论分析了恶意注入与丢弃攻击的异常检测阈值，能够有效实施异常定位；最后通过扩展的 P4 软件交换机模型实现了所提机制并在虚拟的网络环境中对其进行了评估。

3 问题描述

3.1 攻击模型

一个典型的 SDN 是由控制器下发转发策略，数据平面交换机遵循控制器授权的策略转发数据。当存在非诚实的交换机时，数据流未必遵循既定的转发策略。一个运算能力有限的攻击者通过篡改、丢弃、重放、注入等攻击破坏终端之间的通信，SDN 可能面临如下的攻击行为破坏数据分组的转发。

数据分组劫持。攻击者控制数据流路径中的某一交换机，重定向数据流至另一非控制器所授权的路径。

数据分组丢弃。攻击者控制某一交换机并丢弃某一特定路径数据流的数据分组。

数据分组注入。攻击者控制路径中某一交换机并向路径中的后继节点注入数据。

数据分组篡改。攻击者控制的交换机可以篡改数据分组的任意部分，如数据分组头信息、负载数据等。

数据分组重放。攻击者控制某一交换机并重放以前的数据分组。

攻击者的目标是控制某一路径上的恶意交换机并破坏数据流的正常转发。数据分组的劫持攻击可等价于丢弃原路径上正常转发的数据分组；数据分组的重放可等价于重复的数据分组注入攻击；数据分组的篡改等同于丢弃原数据分组并注入伪造的数据分组。

以上的几类攻击可等价划分为注入和丢弃攻击这 2 类攻击，因此，恶意的数据分组注入与丢弃是 SDN 数据平面面临的最基本的安全威胁，针对这 2 类攻击，给出如下定义。

定义 1 路径向量 $\mathrm{PATH}_{\mathrm{FlowSN}}=(R_0,R_1,\cdots,R_L)$ 表示某一流 FlowSN 通过控制器授权的路径，其中 R_i

为路径中各交换机的身份标识，相邻节点分别称为前驱和后继，路径向量中可能存在恶意的交换机节点。

定义 2　任意数据流 FlowSN 持续运行时间 t 是由一系列连续的、随机的时间间隔 t_i 组成的，即 $t=t_1+t_2+\cdots+t_n$。

定义 3　注入攻击 $A_{\text{inject}}(P,R_i,x)$ 表示在时间 $t=\sum_{j=1}^{n}t_n$ 即随机的时间间隔内，路径 P 上存在一恶意交换机 R_i 向其后继节点注入 x 个数据分组。

定义 4　丢弃攻击 $A_{\text{drop}}(P,R_i,y)$ 表示在时间 $t=\sum_{j=1}^{n}t_n$ 内，路径 P 上存在一恶意交换机 R_i 丢弃了比例 y 的数据分组。

本文假设逻辑集中的控制器是安全的，控制器与数据平面的通信信道是安全的，数据流入口与出口交换机为合法交换机，攻击者无法获取合法节点的数据信息与流表项等信息，任一转发路径中不存在相邻的 2 个或 2 个以上恶意节点。此外，数据的机密性由终端用户保证。

3.2　安全目标

本文目标是实现高效的 SDN 包转发验证与异常定位，增强数据平面安全，具体包括以下几个方面。

1) 数据转发可验证

目的主机收到正确的、未经篡改的数据，实现数据源认证与数据完整性验证。

2) 异常定位

检测恶意节点注入与丢弃攻击，能够有效定位网络异常，确保网络功能正常。

3) 性能目标

摒弃现有典型机制中因嵌入固定或可变长度的验证标签信息引入较大开销的方式，以较低的运算与通信开销，实现高效转发。

4　端址重载的转发验证 PAO-PFV

本文设计实现 PAO-PFV 存在的挑战主要包括以下几点。

1) 如何以很少甚至无任何额外的数据分组头开销实现数据的转发验证？现有机制大多通过在数据分组中插入密码标签或其他验证信息实现转发验证。

2) 如何高效实现数据分组源认证与完整性验证？简单的签名机制验证会导致数据分组转发过程中引入巨大的计算开销，降低网络性能。

3) 如何检测恶意的数据注入和丢弃攻击并有效定位网络异常。

本节将阐述如何解决 PAO-PFV 面临的上述设计挑战并实现安全目标。表 1 为 PAO-PFV 部分标识符及含义。

表 1　PAO-PFV 部分标识符及含义

标识符	含义
R_i	交换机标识
FlowSN	流标识号
Load	网络数据分组负载
L	流路径长度
$\text{MAC}_K(\cdot)$	使用密钥 K 的消息认证码
$K(i,j)$	R_i 与 R_j 共享密钥
$H(\cdot)$	哈希函数
Padding	填充域
IP_{INVAR}	IP 头中不变的字段域
$<Q_R, S_R>$	节点公私密钥对
$\text{Cnt}_i^{\text{suc}}(t_j)$	间隔 t_j 内，R_i 验证成功的数据分组数
$\text{Cnt}_i^{\text{fal}}(t_j)$	间隔 t_j 内，R_i 验证失败的数据分组数

4.1　PAO-PFV 机制概述

本节简述 PAO-PFV 中如何实现数据分组端址重载、基于端址重载的概率验证以及异常定位。

数据分组端址重载。当数据流进入网络第一跳入口交换机后，交换机查询流表，若失配，表明这是一条新数据流；交换机计算分组头相关字段生成流标识符，将流标识符、数据分组以及二者的消息认证码向控制器发送 Packet-IN 消息，控制器验证通过后计算新流的路径向量，以 Packet-Out 消息和 Flow-Mod 消息向入口交换机发送路径向量，并向路径上的交换机发送以流标识符为关键字的转发策略，控制器保存如图 2 所示的流路径向量以及地址与端口信息。

FlowSN_1	srcip_1, dstip_1, sport_1, dport_1	Path Vector_1
FlowSN_2	srcip_2, dstip_2, sport_2, dport_2	Path Vector_2
⋮	⋮	⋮
FlowSN_x	srcip_x, dstip_x, sport_x, dport_x	Path Vector_x

图 2　流标识符、端址及路径向量映射表

入口交换机在接收到控制器发送的路径向量后，计算与路径上的各交换机共享密钥并生成短消

息认证码，重构数据分组头域中的端口和地址信息实现端址重载，依流表项转发至下一跳。重载后的端地址信息如图 3 所示，中间阴影部分是重载前的原始 TCP/UDP-IP 相关域：源和目的地址，源和目的端口号。重载后数据分组头格式如图 3 虚线框所示，各字段含义如下。

FlowSN：20 bit 流标识号。

TimeID：16 bit 随机时间间隔 t_j 标识。

MacOffset：6 bit 短消息认证码偏移指针。

MAC_i：η bit 短消息认证码。

Padding：填充字段。

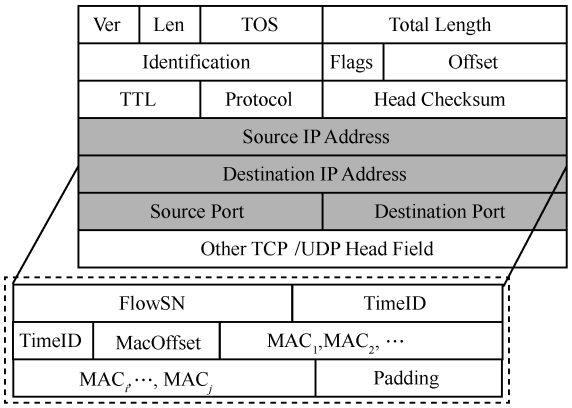


图 3 重载的数据分组头端址字段

图 3 为基于 IPv4 的数据分组头端址重载，对于 IPv6 的处理方式类似，不同的是 IPv6 基本首部长度为固定的 40 B，源与目的地址为 16 B，与 IPv4 相比，重载的端址信息长度由 12 B 增加为 36 B。

基于端址重载的概率验证。路径中的后继交换机在接收到数据分组后，解析数据分组头，基于重载的端址信息，计算短消息认证码 MAC_i'，若与分组头中对应的 MAC_i 相等，表明此消息以概率 $\rho=1/2^{\eta}$ 通过数据源认证与完整性验证，交换机查询流表并转发数据；若验证失败，表明这是一个非法的数据分组，丢弃。同时，交换机更新在时间间隔 t_j 内接收到的验证成功与失败的数据分组计数信息。

异常定位。控制器统计收集在时间 t 内路径上各交换机的转发验证成功与失败的数据分组信息，当某一交换机接收的数据分组统计信息超过丢弃检测阈值 ψ_{drop} 或注入检测阈值 ψ_{inject} 时，表明路径中存在恶意节点，控制器将定位并从网络拓扑中删除恶意链路、重新规划数据流路径。

4.2 数据分组端址重载

基于 Hess-IBS[27]中 Setup 和 Extract 算法，密钥生成中心 PKG 生成主密钥 s，为控制器 C、数据平面交换机分配密钥，完成系统初始化，并公开系统参数 $\mathrm{Params}:<G_1,G_2,\hat{e},P,\mathrm{Pub},H_1,H_2>$，节点 R_0 对应公私钥对为

$$<Q_{R_0},S_{R_0}>=<H_2(R_0),sH_2(R_0)> \tag{1}$$

下面，详细讨论入口交换机实现端址重载的基本算法步骤。在数据流到达第一跳交换机时，交换机重构数据分组头并完成端址重载，其执行如下操作。

1) 根据数据分组头信息，执行操作 $\mathrm{FlowSN}=H(\mathrm{srcIP}\|\mathrm{dstIP}\|\mathrm{sport}\|\mathrm{dport})$，查询以 FlowSN 流标识符为关键字的交换机路径向量表 $\{\mathrm{FlowSN},\mathrm{PATH}_{\mathrm{FlowSN}}=(R_0,\cdots,R_i,\cdots,R_L)\}$ 和流表项，若存在此数据流的路径向量和流表项，执行步骤 2)，否则交换机将计算与控制器的共享密钥，生成流标识符。将流标识符、数据分组和通过共享密钥与二者所生成的消息认证码以 Packet-IN 消息向控制器 C 发送。控制器通过验证后发送新流的路径向量信息和流表信息保存并更新图 2 所示的流标识与分组头端址信息字段的映射表。

2) 解析数据流的路径向量 $\mathrm{PATH}_{\mathrm{FlowSN}}$，获取数据流路径上交换机标识，重构如图 3 虚线框所示的数据分组头域，实现端址信息重载，依流表项转发送数据。

对路径向量中的各交换机 R_i，R_0 可通过一次双线性对[27]计算得到与 R_i 的对称会话密钥，即 R_0 与其路径向量中的所有后继节点 R_i 均通过双线性对计算实现非交互的密钥协商，以此密钥计算 MAC_i 并重构分组头中的地址与端口字段域，各后继交换机验证对应的 MAC_i 域实现基于该短消息认证码的数据源认证与完整性概率验证。

R_0 按式(2)获取与路径上的后继节点共享密钥，并按式(3)计算短消息 MAC_i 验证域并重载分组头，其中，TTL 为 IP 头 TTL 值。R_0 实现端址重载基本流程如图 4 所示。

$$K(0,i)=\hat{e}(S_{R_0},Q_{R_i}),\ 1\leqslant i\leqslant L \tag{2}$$

$$\mathrm{MAC}_L\leftarrow\mathrm{MAC}_{K(0,L)}(\mathrm{FlowSN}\|\mathrm{Load}\|\mathrm{TimeID}\|\mathrm{TTL}_L\|\mathrm{MacOffset}_L\|\mathrm{IP}_{\mathrm{INVAR}})$$

$$\begin{aligned}&\mathrm{MAC}_{L-1}\leftarrow \mathrm{MAC}_{K(0,L-1)}(\mathrm{FlowSN}\,\|\,\mathrm{Load}\,\|\,\mathrm{TimeID}\,\|\\&\qquad \mathrm{TTL}_{L-1}\,\|\,\mathrm{MacOffset}_{L-1}\|\,\mathrm{IP}_{\mathrm{INVAR}}\,\|\,\mathrm{MAC}_L)\\&\mathrm{MAC}_i\leftarrow \mathrm{MAC}_{K(0,i)}(\mathrm{FlowSN}\|\mathrm{Load}\|\mathrm{TTL}_i\,\|\\&\mathrm{TimeID}\,\|\,\mathrm{MacOffset}_i\,\|\,\mathrm{IP}_{\mathrm{INVAR}}\,\|\mathrm{MAC}_{i+1}\cdots\|\,\mathrm{MAC}_L)\end{aligned}\tag{3}$$

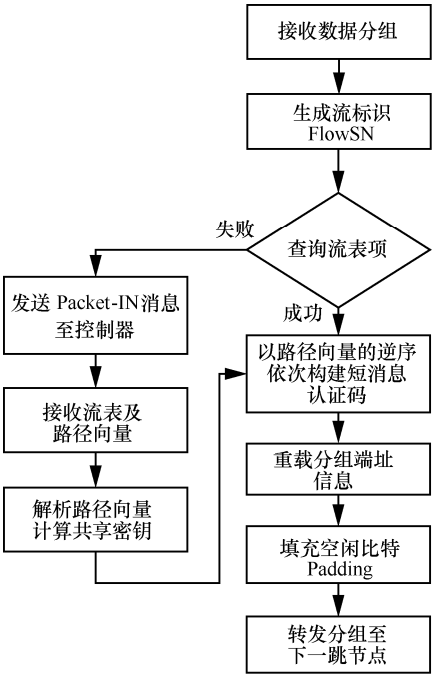


图 4　入口交换机 R_0 重载端址

由式(3)可知，MAC_i 除对数据分组头和负载实现短消息认证，同时也包含了对 $\mathrm{MAC}_{i+1},\cdots,\mathrm{MAC}_L$ 的验证，路径中恶意交换机 R_j 对其后继交换机对应的 η bit 短消息认证码的改变，R_i 能以概率 $\rho=1-1/2^{\eta}$ 检测出 MAC_i 的非一致性并更新数据分组验证计数 $\mathrm{Cnt}_i^{\mathrm{fal}}(t_j)$，否则，数据易遭受假冒攻击，致使异常定位失败。在重载端址信息，转发数据至下一跳交换机之前，需要填充端址字段空闲的比特，即 $\mathrm{Padding}\leftarrow\mathrm{Hash}(\mathrm{Nonce})$，Nonce 是随机数。

4.3　基于端址重载的概率验证

入口交换机 R_0 重构数据分组头实现端址信息重载后，向其后继节点发送端址重载后的数据分组，路径向量中后继交换机 $R_i(1\leqslant i\leqslant L)$ 基于重载的端址信息中 η bit 短消息认证码实现数据源认证与完整性概率验证。对新数据流的首个数据分组，R_i 按式(4)计算与源入口交换机的共享密钥，也即路径向量上交换机 R_i 与其入口交换机 R_0 计算共享密钥 $K(i,0)$。

$$K(i,0)=\hat{e}(S_{R_i},Q_{R_0}),\ 1\leqslant i<L\tag{4}$$

注意到，根据双线性对性质有 $K(i,0)=K(0,i)$，在后续的数据传输中，不需要重复计算。基于端址重载的概率验证算法流程如图 5 所示。

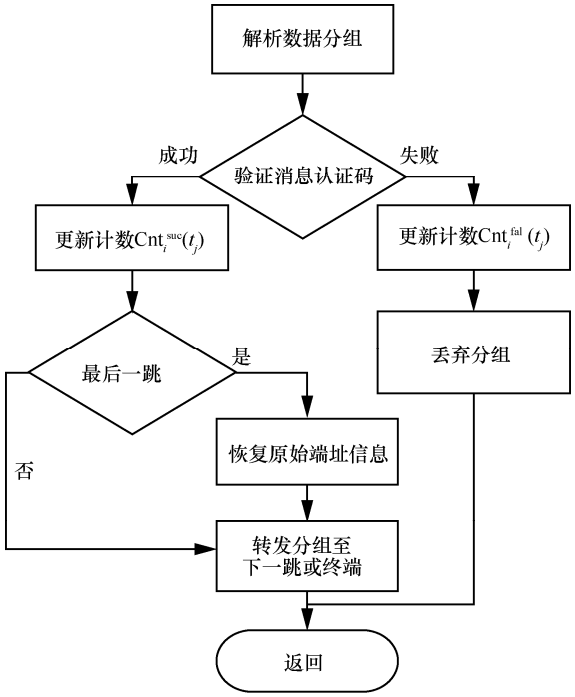


图 5　交换机 R_i 概率验证

在时间间隔 t_j 内，R_i 将按式(3)对数据源和数据完整性进行验证，计算短消息认证码 MAC_i'，若 $\mathrm{MAC}_i'=\mathrm{MAC}_i$，则表明数据通过验证且以概率 $\rho=1/2^{\eta}$ 为有效数据，R_i 依流表转发数据至下一跳交换机 R_{i+1}，若 $\mathrm{MAC}_i'\neq\mathrm{MAC}_i$，表明验证失败，则丢弃数据；不论成功与否，$R_i$ 将更新计数 $\mathrm{Cnt}_i^{\mathrm{suc}}(t_j)$ 和 $\mathrm{Cnt}_i^{\mathrm{fal}}(t_j)$，在下一个间隔 t_{j+1} 开始后，R_i 向控制器发送在间隔 t_j 内的分组验证计数信息 $\mathrm{Cnt}_i^{\mathrm{suc}}(t_j)$ 和 $\mathrm{Cnt}_i^{\mathrm{fal}}(t_j)$。

图 5 中当交换机是目的节点交换机时，需要将端址重载后的数据分组头恢复为原始的数据分组头，由图 2 可知，在控制器为新的数据流计算路径向量时，保存了原始数据分组头的端址域字段信息，通过如 OpenFlow 协议命令 OPFAT_SET_NW_SRC、OPFAT_SET_NW_DST、OPFAT_SET_TP_SRC、OPFAT_SET_TP_DST[2]等命令，在最后一跳目的交换机执行以上命令恢复原始

数据分组的端址字段域，转发数据至终端。

4.4 异常定位

网络启动后，控制器建立 SDN 全局视图，实时获取全网的状态信息，并根据网络策略向交换机下发流表。本文实现了控制器路径向量计算组件（PVCC, path vector computation component），收到来自入口交换机 Packet-IN 请求信息时，为新流计算路径向量 $\mathrm{PATH}_{\mathrm{FlowSN}}=(R_0,\cdots,R_i,\cdots,R_L)$，构建如图 2 所示的路径向量映射表 $\{\mathrm{FlowSN},\mathrm{PATH}_{\mathrm{FlowSN}}\}$。

在敌对环境中的异常定位只能定位与恶意节点相邻的链路，而非恶意节点本身。PAO-PFV 中控制器通过统计在时间 $t=\sum_{j=1}^{n}t_j$，即路径节点在 n 个连续随机间隔内转发验证成功和失败的包计数信息 $\sum_{j=1}^{n}\mathrm{Cnt}_i^{\mathrm{suc}}(t_j)$ 与 $\sum_{j=1}^{n}\mathrm{Cnt}_i^{\mathrm{fal}}(t_j)$，$i\in\mathrm{PATH}_{\mathrm{FlowSN}}$ 定位异常，异常定位算法流程如图 6 所示，其总的复杂度为 $\Theta(L)$。

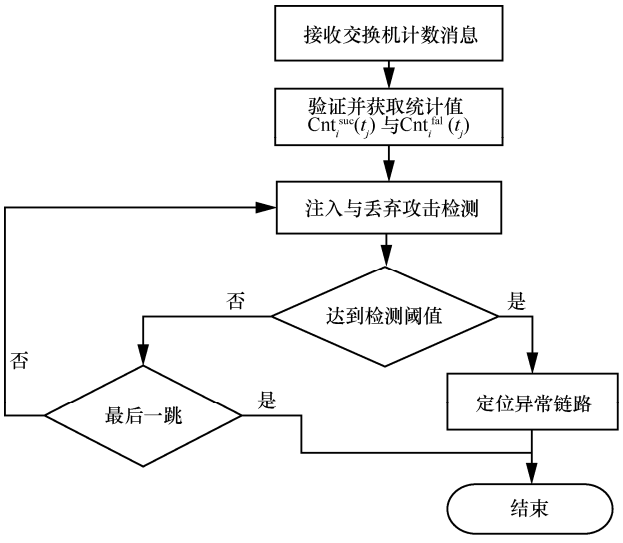


图 6 控制器异常检测

图 6 中控制器接收 R_i 在随机间隔 t_j 内的计数消息 $\mathrm{MSG}\|\mathrm{MAC}_{K(i,C)}(\mathrm{MSG})$，其中，$K(i,C)$是 R_i 与控制器 C 通过式(2)计算所得到的共享密钥、$\mathrm{MSG}=(R_i,t_j,\mathrm{FlowSN},\mathrm{Cnt}_i^{\mathrm{suc}}(t_j),\mathrm{Cnt}_j^{\mathrm{fal}}(t_j))$。控制器在收到路径中的交换机计数消息 MSG、验证并获取计数 $\mathrm{Cnt}_i^{\mathrm{suc}}(t_j)$ 和 $\mathrm{Cnt}_i^{\mathrm{fal}}(t_j)$，执行异常判定算法，当下游节点接收验证错误的数据分组超过阈值 ψ_{inject} 时，即 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)>\psi_{\mathrm{inject}}$，其中 $t=\sum_{j=1}^{n}t_j$，表明有来自上游节点的恶意注入与篡改，与此节点连接的链路将被判定为恶意链路；当相邻的 2 个节点 R_{i-1} 与 R_i 接收的有效数据分组个数与 $\mathrm{Cnt}_{i-1}^{\mathrm{suc}}(t)-\mathrm{Cnt}_i^{\mathrm{suc}}(t)$ 和上游节点 R_{i-1} 接收的有效数据分组 $\mathrm{Cnt}_{i-1}^{\mathrm{suc}}(t)$ 的比例超过阈值 ψ_{drop}，也即 $\mathrm{Cnt}_i^{\mathrm{suc}}(t)<\mathrm{Cnt}_{i-1}^{\mathrm{suc}}(t)(1-\psi_{\mathrm{drop}})$，表明 R_{i-1} 与 R_i 间的链路为恶意链路。

对于如何设定恶意注入与丢弃攻击检测阈值 ψ_{inject} 与 ψ_{drop}，将关系到异常定位的准确性，第 5 节将对此给出详细的理论分析。

5 分析与讨论

本节将通过理论分析给出 PAO-PFV 机制异常定位检测阈值，并讨论机制的安全性、算法复杂度以及相关机制的分组头开销和运算开销。

5.1 检测阈值

在执行异常检测时，由于网络拥塞或传输出错引发的链路自然数据分组丢失等原因，可能存在的情况是在时间 $t=\sum_{j=1}^{n}t_j$ 内，R_i 接收且验证有效的数据分组计数值 $\mathrm{Cnt}_i^{\mathrm{suc}}(t)>0$ 小于 R_{i-1} 计数 $\mathrm{Cnt}_{i-1}^{\mathrm{suc}}(t)>0$，即 $\mathrm{Cnt}_i^{\mathrm{suc}}(t)<\mathrm{Cnt}_{i-1}^{\mathrm{suc}}(t)$，但这也并不意味着 R_{i-1} 执行了恶意丢弃动作。同样，若某一 R_i 返回的 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)>0$，意味着存在恶意交换机注入数据的行为，但这并不等价于其前驱节点交换机 R_{i-1} 为恶意节点。考虑到基于端址重载的概率验证，设短消息认证码为 η =3 bit MAC，可能存在的情况是交换机 R_{i-2} 为恶意节点，其注入的数据恰好在 R_{i-1} 处以概率 $1/2^{\eta}=0.125$ 通过了概率验证，并转发给节点 R_i，在 R_i 处验证失败，此时 R_i 将更新 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)$ 计数，但 R_{i-1} 为非恶意节点，即 R_{i-1} 与 R_i 之间的链路 lk_i 并非异常链路。

本节将针对恶意节点注入与丢弃的异常检测，在设定的误报率与漏报率下，通过理论分析，给出异常检测定位的阈值 ψ_{drop} 和 ψ_{inject}。

定理 1 对于给定的误报率 $P_{\mathrm{fp}}\leqslant\lambda$，恶意节点注入数据异常检测阈值 $\psi_{\mathrm{inject}}=\dfrac{2\ln\left(\dfrac{L-d}{\lambda}\right)}{\rho^2}$。

证明 对于节点 R_i 接收到虚假的数据分组，验

证失败且 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)$ 计数值增加的概率 $\rho=1-(1/2^{\eta})$（η bit MAC），若此数据分组通过 R_i 验证且转发至下一跳节点 R_{i+1}，则 $\mathrm{Cnt}_{i+1}^{\mathrm{fal}}(t)$ 计数值增加的概率为 $\rho(1-\rho)$。假设存在一个恶意节点 R_{i-1}，其后继节点 R_i 为诚实节点，R_{i-1} 向 R_i 注入 x 个数据分组，R_i 验证数据分组，当 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\psi_{\mathrm{inject}}$ 且 $\mathrm{Cnt}_j^{\mathrm{fal}}(t)\geqslant\psi_{\mathrm{inject}}$ $(j\geqslant i+1)$ 时，将引发误报，此时定位 R_i 与 R_{i+1} 间的链路 lk_{i+1} 为异常链路，设误报率为 P_{fp}，则

$$P_{\mathrm{fp}}=\sum_{j=i+1}^{L}P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\psi_{\mathrm{inject}},\mathrm{Cnt}_j^{\mathrm{fal}}(t)\geqslant\psi_{\mathrm{inject}})=$$
$$(L-d)P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\mathrm{Cnt}_{i+1}^{\mathrm{fal}}(t)) \tag{5}$$

其中，L 为路径向量长度，d 为后继交换机 R_i 与 R_0 的距离。R_{i-1} 注入 x 个数据分组，则 R_i 与 R_{i+1} 验证失败计数值 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)$ 与 $\mathrm{Cnt}_{i+1}^{\mathrm{fal}}(t)$ 期望值分别为

$$u_i=\mathrm{E}(\mathrm{Cnt}_i^{\mathrm{fal}}(t))=\rho x$$
$$u_{i+1}=\mathrm{E}(\mathrm{Cnt}_{i+1}^{\mathrm{fal}}(t))=\rho(1-\rho)x \tag{6}$$

联立式(5)，有

$$P_{\mathrm{fp}}=(L-d)P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\mathrm{Cnt}_{i+1}^{\mathrm{fal}}(t))=$$
$$(L-d)P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<u_{i+1})=$$
$$(L-d)P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<(1-\rho)u_i) \tag{7}$$

基于切诺夫界[28]，可得

$$P_{\mathrm{fp}}=(L-d)P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<(1-\rho)u_i)\leqslant$$
$$(L-d)\mathrm{e}^{-\frac{\rho^2u_i}{2}}=(L-d)\mathrm{e}^{-\frac{\rho^2\rho x}{2}} \tag{8}$$

令 $P_{\mathrm{fp}}\leqslant\lambda$，即 $(L-d)\mathrm{e}^{-\frac{\rho^2\rho x}{2}}\leqslant\lambda$，可得

$$x\geqslant\frac{2\ln\left(\frac{L-d}{\lambda}\right)}{\rho^3} \tag{9}$$

取 R_{i-1} 注入数据且 R_i 验证失败的期望值为检测阈值，即

$$\psi_{\mathrm{inject}}=\rho x=\frac{2\ln\left(\frac{L-d}{\lambda}\right)}{\rho^2} \tag{10}$$

因此，当误报率 $P_{\mathrm{fp}}\leqslant\lambda$，恶意节点注入数据定位检测阈值 $\psi_{\mathrm{inject}}=\dfrac{2\ln\left(\frac{L-d}{\lambda}\right)}{\rho^2}$，证毕。

定理 2　对于给定的漏报率 λ，恶意节点注入 x 个数据分组的最大上界

$$x=\frac{\psi_{\mathrm{inject}}-\ln\lambda+\sqrt{(\psi_{\mathrm{inject}}-\ln\lambda)^2-\psi_{\mathrm{inject}}{}^2}}{\rho}$$

若节点注入数据大于 x，则检测算法将以概率 $P\geqslant1-P_{\mathrm{fn}}=1-\lambda$ 定位异常。

证明　恶意节点 R_{i-1}，其下游节点 R_i 为诚实节点，R_{i-1} 向 R_i 注入 x 个数据分组，R_i 验证数据分组，若 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\psi_{\mathrm{inject}}$ 且 $\mathrm{Cnt}_{i+1}^{\mathrm{fal}}(t)<\psi_{\mathrm{inject}}$（$j\geqslant i+1$），此时将会引发漏报，漏报率满足

$$P_{\mathrm{fn}}=P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\psi_{\mathrm{inject}})=$$
$$P(\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\left[1-\left(1-\frac{\psi_{\mathrm{inject}}}{u_i}\right)\right]u_i) \tag{11}$$

其中，$u_i=\rho x$，基于切诺夫界可得

$$P_{\mathrm{fn}}\leqslant\mathrm{e}^{-\frac{\left(1-\frac{\psi_{\mathrm{inject}}}{u_i}\right)^2u_i}{2}}=\lambda \tag{12}$$

式(12)可化简为

$$x=\frac{\psi_{\mathrm{inject}}-\ln\lambda+\sqrt{(\psi_{\mathrm{inject}}-\ln\lambda)^2-\psi_{\mathrm{inject}}{}^2}}{\rho} \tag{13}$$

当 R_{i-1} 注入不超过 x 个数据分组时，若 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)<\psi_{\mathrm{inject}}$，恶意节点可规避定位检测，引发漏报；但当 R_{i-1} 注入超过 x 个数据分组时，$\mathrm{Cnt}_i^{\mathrm{fal}}(t)>\psi_{\mathrm{inject}}$，此时，$R_{i-1}$ 与 R_i 间的链路 lk_i 将会以概率 $P\geqslant1-\lambda$ 被检测定位为异常，证毕。

定理 3　对于某一给定误报率或漏报率 λ，异常丢弃检测阈值 $\psi_{\mathrm{drop}}=\bar{\alpha}+\sqrt{\dfrac{2\ln(1/\lambda)}{(1-\bar{\alpha})^{d+2}N}}$。

证明　传输链路由于网络拥塞、传输错误等原因，会引发网络自然数据分组丢失，设某一路径向量 $\mathrm{PATH}=(R_0,R_1,R_2,\cdots,R_L)$，各节点间链路 lk_1，lk_2，lk_i，lk_L 平均自然数据分组丢失率为 $\alpha_i(1\leqslant i\leqslant L)$。恶意节点丢弃异常定位监测阈值 ψ_{drop} 设定过高或过低将引发较大的漏报或误报。当节点 R_{i-1} 与 R_i 间链路数据分组丢失率 α^* 超过了检测阈值 ψ_{drop}，链路 lk_i 将被判定为异常链路。在无恶意丢弃的条件下，节点接收验证成功的分组数的期望值 $u_i=\mathrm{E}(\mathrm{Cnt}_i^{\mathrm{suc}}(t))$ 满足如下关系

$$u_i=(1-\alpha_1)(1-\alpha_2)\cdots(1-\alpha_i)N \tag{14}$$

其中，$\alpha_i(1\leqslant i\leqslant L)$ 为自然数据分组丢失率，N 为第

一跳交换机在时间 t 内接收的终端数据分组数。假设各链路自然数据分组丢失率相等，即 $\alpha_1=\alpha_2=\cdots\alpha_i=\overline{\alpha}$，上式可表示为 $u_i=(1-\overline{\alpha})^i N$，当 R_{i-1} 存在恶意的丢弃使链路 lk_i 数据分组丢失率 α^* 超过了检测阈值 ψ_{drop}，R_i 接收的分组 $\mathrm{Cnt}_i^{\mathrm{suc}}(t)$ 有如下关系

$$\mathrm{Cnt}_i^{\mathrm{suc}}(t)=(1-\alpha^*)(1-\overline{\alpha})^{i-1}N<(1-\psi_{\mathrm{drop}})(1-\overline{\alpha})^{i-1}N \tag{15}$$

为使漏报率 $P_{\mathrm{fn}}\leqslant\lambda$，满足式(16)成立。

$$\begin{aligned}&P_{\mathrm{fn}}=P\left[\mathrm{Cnt}_i^{\mathrm{suc}}(t)<(1-\psi_{\mathrm{drop}})(1-\overline{\alpha})^{i-1}N\right]=\\&P\left[\mathrm{Cnt}_i^{\mathrm{suc}}(t)<\frac{1-\psi_{\mathrm{drop}}}{1-\overline{\alpha}}(1-\overline{\alpha})^{i}N\right]=\\&P\left[\mathrm{Cnt}_i^{\mathrm{suc}}(t)<\left(1-\frac{\psi_{\mathrm{drop}}-\overline{\alpha}}{1-\overline{\alpha}}\right)u_i\right]\leqslant\lambda\end{aligned} \tag{16}$$

基于切诺夫界，则有

$$P_{\mathrm{fn}}=\mathrm{e}^{-\frac{\left(\frac{\psi_{\mathrm{drop}}-\overline{\alpha}}{1-\overline{\alpha}}\right)^2 u_i}{2}}\leqslant\lambda \tag{17}$$

因此有

$$\psi_{\mathrm{drop}}=\overline{\alpha}+\sqrt{\frac{2\ln(1/\lambda)}{(1-\overline{\alpha})^{d+2}N}} \tag{18}$$

其中，d 是路径中节点 R_d 与源节点的距离。对于误报率可得同样的结果，证毕。

5.2 安全性分析

本节讨论 PAO-PFV 机制的安全性。如第 3 节所述，恶意节点篡改、丢弃、注入及重放等攻击可等价划分为注入与丢弃攻击 2 类。

数据注入。恶意节点 R_i 注入虚假数据，必将会使得短消息验证码 MAC_{i+1} 以概率 $\rho=1-(1/2^{\eta})$ 验证失败，并增加 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)$ 计数值，异常检测算法可有效定位。

数据丢弃。恶意节点 R_i 丢弃有效的数据分组，将使其与邻居节点 R_{i+1} 计数值 $\mathrm{Cnt}_{i+1}^{\mathrm{suc}}(t)$ 与 R_i 计数值 $\mathrm{Cnt}_i^{\mathrm{suc}}(t)$ 存在显著差异，节点 R_i 报告其统计值为 $\mathrm{Cnt}_i^{\mathrm{suc}}(t)$，由于丢弃了部分数据，则 $\mathrm{Cnt}_i^{\mathrm{suc}}(t)-\mathrm{Cnt}_{i+1}^{\mathrm{suc}}(t)$ 差值将使其超过了测阈值（R_i 丢弃数据使得 $\mathrm{Cnt}_{i+1}^{\mathrm{suc}}(t)$ 减小），检测算法将定位 R_i 与 R_{i+1} 之间的链路为异常链路。

现有的各典型方案提供的安全功能如表 2 所示。

5.3 算法复杂度分析

PAO-PFV 中数据流路径向量和共享密钥是流的首个分组到来时动态建立的，在新数据流入网后，第一跳源交换机 R_0 需要根据路径向量通过 L（L 为路径长度）次对运算建立与各交换机之间的共享密钥，各后继交换机 R_i 仅需要通过一次对运算建立与第一跳交换机 R_0 的共享密钥，并进行一次短消息认证码验证，在后续的通信中，由于暂存了共享密钥，R_0 仅需 L 次摘要运算；而 R_i 仅需一次认证码验证计算；控制器 C 通过一次扫描实现异常检测，其算法复杂度为 $\Theta(L)$。表 3 为相似方案在单次数据传输中各节点的算法复杂度，其中 P 表示一次对运算，M 表示一次摘要运算，E 表示加解密运算，N 表示属性签名中的属性元个数。

5.4 分组头开销

本节讨论现有典型机制的数据分组头开销情况。基于属性基的 ACFlow 机制需要修改用户终端网络协议栈[26]，嵌入属性标签，其分组头开销随路径增加维持为一固定值，假设签名信息为 1 024 bit，分组头开销达 144 B。SDNsec[25]固定分组头开销为 22 B，路径长度增加 1，分组头长度将增加 8 B 用于加密下发流表规则，分组头总长度为(22+8L) B。OPT[21]路径长度增加 1，分组头长度将增加 16 B，

表 2 典型方案安全功能分析

方案	机密性	完整性	源认证	异常定位	抗注入攻击	抗丢弃攻击	路径一致性
文献[21]	×	√	√	×	√	×	√
文献[23]	×	√	√	√	√	√	√
文献[24]	×	×	×	√	√	√	√
文献[25]	×	√	×	×	√	×	√
文献[26]	×	√	√	×	√	×	√
PAO-PFV	×	√	√	√	√	√	√

注：√表示支持，×表示不支持。

为(52+16*L*) B。图 7 为 PAO-PFV、SDNsec、ACFlow、OPT 方案分组头在负载为 600 B，路径长度为 2、4、6、8、10 时，数据分组头所占开销曲线变化。

表 3　　相似方案计算复杂度

方案	源节点/终端	中间节点	目的节点/终端
文献[21]	LM	2M	LM
文献[25]	LE	M+E	M+E
文献[26]	NP	—	NP
PAO-PFV	LM	M	M

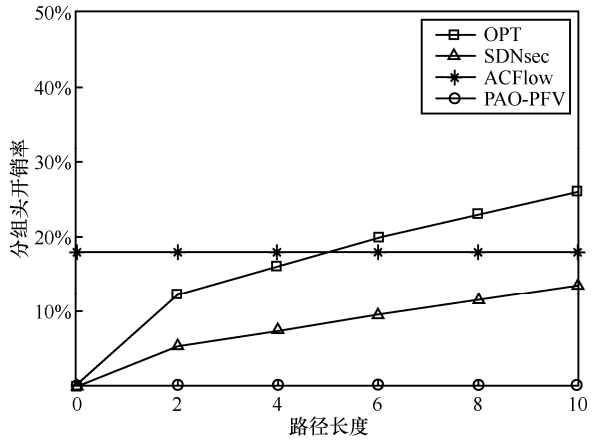


图 7　分组头开销随路径长度变化

由图 7 可知，ACFlow 分组头所占开销约占数据分组的 17%；随着路径长度的增加，SDNsec 分组头开销约占 8%～14%，OPT 分组头开销约 10%～26%；PAO-PFV 通过端址重载，无任何分组头开销，明显优于 SDNsec 与 ACFlow 和 OPT 等机制。

6　实验及评估

本节搭建了一个简单的模拟网络环境，通过扩展相关仿真组件对方案攻击检测的有效性进行评估，最后评估网络传输时延、网络吞吐率等性能。

6.1　实验设置

实验使用联想 Lenovo E580 作为运行平台，配置为 Inter(R) Core(TM)i7-8550 CPU、1.8 GHz，8 GB 内存，操作系统运行 64 位 Ubuntu14.06，使用 Mininet 模拟网络环境，可编程 P4 软件交换机、扩展的交换机行为模型（BMV2, behavioral-model version 2）[16] 以及基于 P4Runtime 接口实现的控制器相关组件，模拟网络采用分布式拓扑，由 30 个虚拟 P4 交换机及若干虚拟主机终端组成。

实验采用 PBC[29]库实现双线性对运算，一次对运算和一次消息认证码计算时间分别为 0.60 ms 及 10 μs 左右。基于对 P4、BMV2 模型的扩展，以下将对方案攻击检测有效性和网络性能进行评估。

6.2　攻击检测评估

实验选择一条路径长 L=8 的简单路径 $\mathrm{PATH}=(R_0,R_1,\cdots,R_L)$，恶意节点 R_4 实施数据丢弃和注入攻击，本实验用于检测在 5.1 节中理论设定的误报与漏报率 λ 条件下，异常定位恶意链路真实的误报率与漏报率及算法有效性。

定义恶意节点的一次攻击是在某一单独的随机时间间隔 t_i 内或若干连续的间隔内实施的注入、丢弃或二者的组合，设恶意节点攻击次数为 M，将控制器检测到一次攻击并有效定位异常称为一次命中，令未检测到的恶意攻击次数为 S、检测到攻击但不能实现异常定位的次数为 N，则漏报率（FN, false negative）可表示为 $\mathrm{FN}=(S+N)/M$；控制器检测到恶意攻击但定位错误的次数为 P、无恶意攻击但引发错误定位异常的次数为 T，则误报率（FP, false positive）可表示为 $\mathrm{FP}=(P+T)/(M+T)$。实验 1～实验 4 中选取理论误报与漏报率 $\lambda=1\%$，其他 λ 设定值实验与此类似。

实验 1。本实验共进行 5 次，每次节点 R_4 分别以 injectrate=0.1%、0.5%、1%、2%、4%的概率在随机选择的时间间隔内向路径中的后继节点注入虚假数据，MAC 长度 $\eta=2$ bit。通过 iperf 以持续时间 t=1 500 s 向 R_0 发送数据，随机的时间间隔 t_i 设定为 $100\ \mathrm{ms}\leqslant t_i\leqslant 200\ \mathrm{ms}$，控制器统计节点转发信息实施异常检测，结果如图 8 所示。

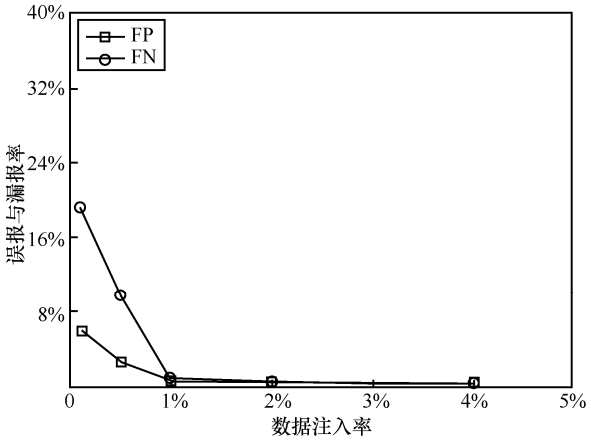


图 8　注入攻击检测

从图 8 可以看出，节点以极小概率实施数据注入攻击时，定位算法未能及时捕获攻击行为，漏报

率较高，这是由于计数值 $\mathrm{Cnt}_i^{\mathrm{fal}}(t)$小于检测阈值，控制器检测到攻击但不能定位；$\mathrm{Cnt}_i^{\mathrm{fal}}(t)$在达到检测阈值时依然存在小概率定位错误引发误报；注入数据率增大，漏报误报率随之降低，注入数据率在 2%与 4%时，误报与漏报率低于预设的 $\lambda=1\%$，约为 0.5%～0.8%。

实验 2。节点 R_4 同样以 injectrate=0.1%、0.5%、1%、2%、4%的概率向路径中的后继节点注入虚假数据，检测 MAC 长度分别为 $\eta=3$ bit 和 $\eta=4$ bit 时检测算法的漏报率。通过 iperf 以持续时间 t=1 500 s 向 R_0 发送数据，实验结果如图 9 所示。

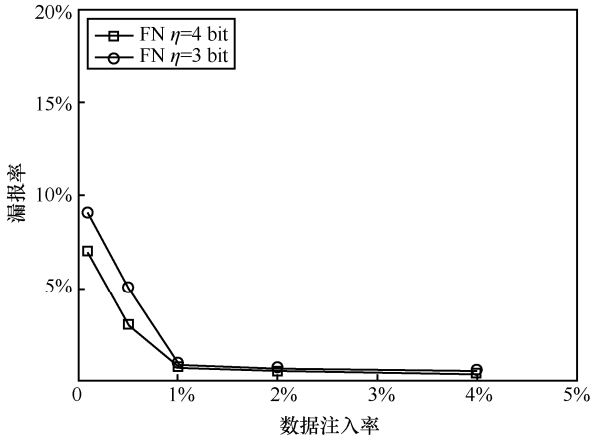


图 9　不同 MAC 位长注入漏报率

由图 9 可以看出，MAC 位长度增加，检测定位算法的漏报率也随之降低。对给定的 λ，MAC 长度 η 增大，注入的虚假数据通过验证的概率减小，则验证失败计数概率 ρ 值增大，检测阈值减小，此时漏报率将减小，当 injectrate=2%时，漏报率约为 0.4%。

实验 3。该实验共进行 4 次，节点 R_4 每次分别以数据分组丢失率 droprate=0.5%、1%、2%、4%的概率在随机选择的时间间隔内丢弃数据分组，网络自然数据分组丢失率 $\alpha=0.4\%\sim0.8\%$，平均数据分组丢失率 $\bar{\alpha}\approx0.6\%$，通过 iperf 以持续时间 t=1 500 s 向 R_0 发送数据，实验结果取平均值，如图 10 所示。

图 10 中，在恶意数据分组丢失率较低的情况下，定位检测的误报率与漏报率较高。在检测时间 t 较短时，恶意数据分组丢失数较少，未能达到检测阈值，将发生漏报，而由于受自然数据分组丢失率 α 的动态变化的影响使错误定位异常也将引发误报。随着节点恶意数据分组丢失率的增大，误报与漏报率明显减少，这与理论设定的 λ 值相近，约为 0.6%。

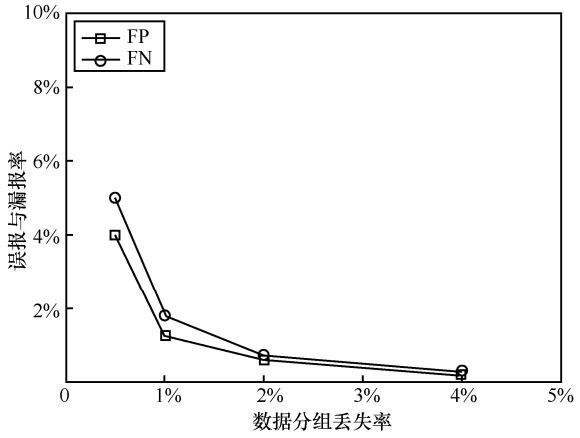


图 10　丢弃攻击检测

实验 4。组合攻击实验，共进行 3 次，恶意节点 R_4 分别以 injectrate=2%、droprate=2%，injectrate=4%、droprate=4%以及 injectrate=2%、droprate=4%在随机选取的间隔内实施组合攻击。$\eta=3$ bit，平均数据分组丢失率 $\bar{\alpha}\approx0.6\%$。iperf 以持续时间 t=1 500 s 向 R_0 发送数据，结果取平均值，如图 11 所示，定位检测算法可以有效实施定位，误报率与漏报率为 0.2%～0.5%。

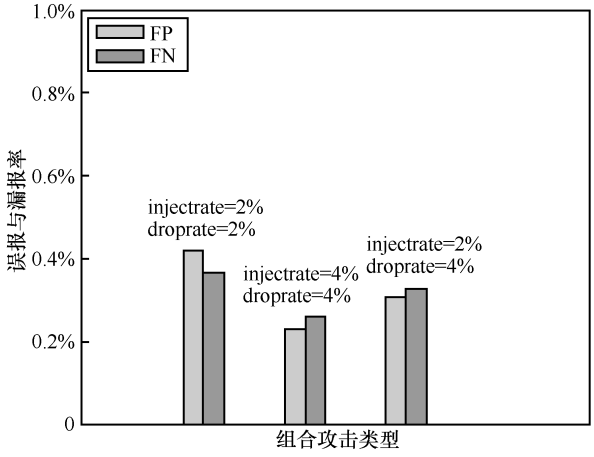


图 11　组合攻击检测

基于以上分析可知，受检测阈值影响，异常检测结果存在漏报与误报的可能，检测算法不可能同时避免这 2 类误差。恶意节点以极小概率实施注入与丢弃攻击时，异常检测漏报率较高，以较大概率实施攻击时，漏报率减小。在恶意节点以极小概率实施攻击时，可通过累加间隔 t_i 统计信息，获取足够的验证信息可以减少误报；在恶意节点以较大概率实施攻击时，

可以适当减少间隔 t_i 统计信息，减小漏报，以快速定位异常链路。

6.3　网络性能评估

本节对网络性能进行评估，需要说明的是，该仿真实验是在单机平台上的模拟网络环境，与真实网络环境在性能上存在很大差异，评价参数包括往返时延（RTT, round trip time）和数据吞吐率。实验测量了运行 PAO-PFV 协议以及未运行 PAO-PFV（UN-PAO-PFV）也即运行基本的 SDN 网络协议的网络性能，并对同类方案 OPT[21]、SDNsec[25]和 ACflow[26]的性能进行了评估。

实验 5。PAO-PFV 中不需要修改终端协议栈，其往返时延 RTT 的测量可通过 PING 数据分组来测量。选取长度为 2、4、6、8、10 的路径进行测试，实验结果分别如图 12 和图 13 所示。

图 12 是路径长度 L=6 时，最初开始的 6 个数据分组的往返时延，运行基本协议与运行 PAO-PFV 协议往返时延 RTT 曲线对比。从图 12 可以看出，新数据流的第一个测试分组比后续的数据分组往返时延多了约 18 ms，这是由于第一个数据分组在数据传输的过程中需要计算与路径向量上的相关节点的共享密钥，在随后的测试中，二者的往返时延差约为 1 ms。

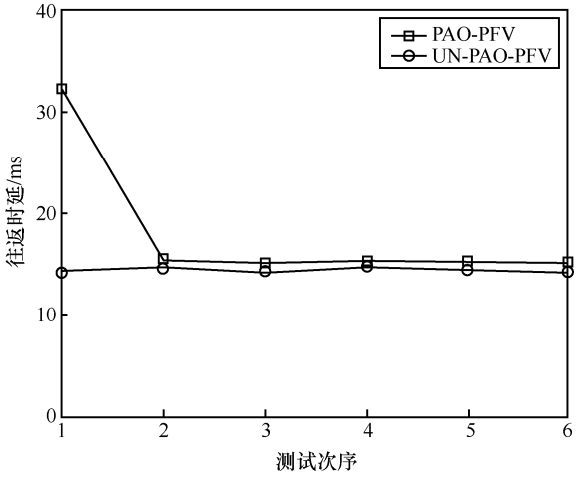


图 12　路径长度 L=6 时，最初开始的 6 个数据分组的往返时延

图 13 是不同路径长度下类似方案 OPT、SDNsec、ACflow 与 UN-PAO-PFV 和 PAO-PFV 的往返时延，路径长度为 8 时 PAO-PFV 的 RTT 约为 23 ms，UN-PAO-PFV 的 RTT 为 21 ms，PAO-PFV 引入了不超过 10%的单向网络传输时延，OPT 与 SDNsec 相当，为 24～25 ms，高于 PAO-PFV，引入了 14%～15%的时延，而 ACflow 因分组的验证都需经过对运算，时延最大，引入了约 20%的时延。

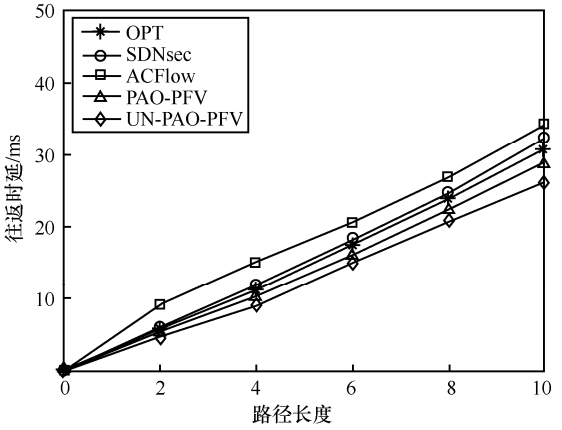


图 13　不同路径长度下不同方案的往返时延

实验 6。通过 iperf 向网络中注入数据，测试数据吞吐率。与往返时延测试类似，实验对 PAO-PFV 协议、UN-PAO-PFV 以及 OPT、SDNsec 和 ACflow 进行了测试。

图 14 和图 15 分别表示路径长度为 4 和 8、终端之间数据吞吐率在负载为 300～1 200 B 时的变化情况。可以看到，PAO-PFV 协议的终端间的数据吞吐率与 UN-PAO-PFV 的数据吞吐率十分接近，可达到基本协议的 93%以上，PAO-PFV 中通过端址重载实现包转发验证所带来的通信开销损失小于 7%～8%，而 OPT 与 SDNsec 通信开销损失在 11%～12%，ACflow 开销损失最大，占 16%～17%。实验 5 和实验 6 结果表明，PAO-PFV 机制在安全性提高的前提下，并未以大量牺牲网络性能为代价。

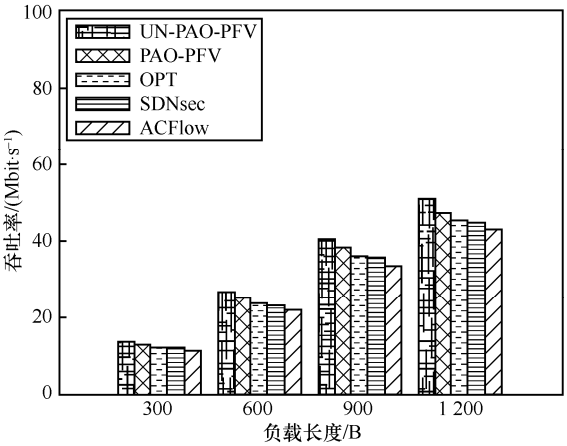


图 14　路径长度为 4 时不同负载的数据吞吐率

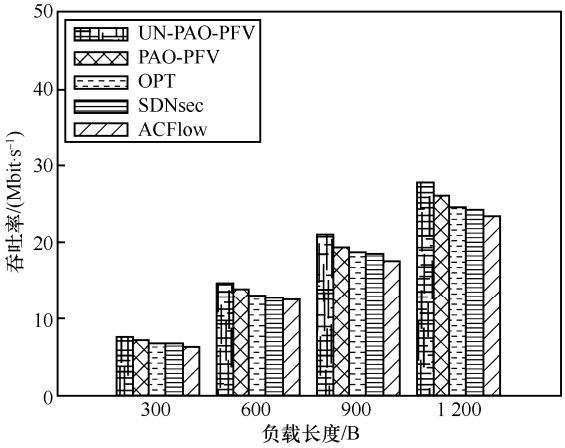


图 15 路径长度为 8 时不同负载的数据吞吐率

7 结束语

SDN 中现有的转发验证机制大多通过附加固定或可变长度的分组验证字段或标签实现转发验证，引入了较大的通信开销与运算开销，降低了网络性能。利用 SDN 转发控制分离与网络可编程的特点，本文提出了基于端址重载的 SDN 分组转发验证机制 PAO-PFV。PAO-PFV 中源交换机重构分组头端口和地址信息实现端址重载，各后继交换机基于重载的端址信息实现分组转发验证，控制器统计流路径中节点在随机时间间隔内接收验证正确与错误的分组计数信息并定位异常；为有效定位异常，理论分析了恶意注入与丢弃攻击的异常检测阈值；最后，通过扩展 P4 软件交换机实现了所提机制。通过重载端址，PAO-PFV 最大的优点在于无任何额外的数据分组头开销，可有效检测恶意的注入与丢弃攻击；分析与实验结果表明，该机制优于同类方案，以引入不超过 10%的网络传输时延和低于 8%的吞吐率损失实现了高效转发验证，下一步的工作将基于 P4 硬件交换机评价其在真实环境中的性能。

参考文献：

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

[2] NUNES B A A, MENDONCA M, NGUYEN X N, et al. A survey of software-defined networking: past, present, and future of programmable networks[J]. IEEE Communications Surveys & Tutorials, 2014, 16(3): 1617-1634.

[3] 王蒙蒙, 刘建伟, 陈杰, 等. 软件定义网络: 安全模型、机制及研究进展[J]. 软件学报, 2016, 27(4): 969-992.
WANG M M, LIU J W, CHEN J, et al. Software defined networking: security model, threats and mechanism[J]. Journal of Software, 2016, 27(4): 969-992.

[4] SINGH D, SHIV A, CHAMOLI S K. Software defined networking (SDN) challenges, issues and solution[J]. International Journal of Engineering and Computer Science, 2019, 7(1): 884-889.

[5] GUDE N, KOPONEN T, PETTIT J, et al. Nox[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(3): 105-110.

[6] PORRAS P, CHEUNG S, FONG M, et al. Securing the software-defined network control layer[C]// Network and Distributed System Security Symposium. Piscataway: IEEE Press, 2015: 1-15.

[7] SHIN S, SONG Y, LEE T, et al. Rosemary: a robust, secure, and high-performance network operating system[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2014: 78-89.

[8] HONG S, XU L, WANG H, et al. Poisoning network visibility in software-defined networks: new attacks and countermeasures[C]// Network and Distributed System Security Symposium. Piscataway: IEEE Press, 2015: 1-15.

[9] ANDERSEN D G, BALAKRISHNAN H, FEAMSTER N, et al. Accountable Internet protocol (AIP)[C]//Proceedings of the ACM SIGCOMM 2008 conference on Data communication. New York: ACM Press, 2008: 1-8.

[10] PAPPAS C, REISCHUK R M, PERRIG A. FAIR: forwarding accountability for Internet reputability[C]//2015 IEEE 23rd International Conference on Network Protocols. Piscataway: IEEE Press, 2015: 189-200.

[11] ZHANG X, ZHOU Z, HSIAO H C, et al. ShortMac: efficient data-plane fault localization[C]//Network and Distributed System Security Symposium. Piscataway: IEEE Press, 2012: 2-12.

[12] MIZRAK A T, CHENG Y C, MARZULLO K, et al. Fatih: detecting and isolating malicious routers[C]//2005 International Conference on Dependable Systems and Networks. Piscataway: IEEE Press, 2005: 538-547.

[13] LIU K J, DENG J, VARSHNEY P K, et al. An acknowledgment-based approach for the detection of routing misbehavior in MANETs[J]. IEEE Transactions on Mobile Computing, 2007, 6(5): 536-550.

[14] ZHANG X, JAIN A, PERRIG A. Packet-dropping adversary identification for data plane security[C]//Proceedings of the 2008 ACM CoNEXT Conference. New York: ACM Press, 2008: 24.

[15] PADMANABHAN V N, SIMON D R. Secure traceroute to detect faulty or malicious routing[J]. ACM SIGCOMM Computer Communication Review, 2003, 33(1): 77-82.

[16] BOSSHART P, DALY D, GIBB G, et al. P4: programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87-95.

[17] YAO G, BI J, XIAO P Y. Source address validation solution with OpenFlow/NOX architecture[C]//2011 19th IEEE International Conference on Network Protocols. Piscataway: IEEE Press, 2011: 7-12.

[18] CASADO M, FREEDMAN M J, PETTIT J, et al. Ethane[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(4): 1-12.

[19] BALLARD J R, RAE I, AKELLA A. Extensible and scalable network monitoring using OpenSAFE[C]//Internet Network Management Con-

ference on Research on Enterprise Networking. Berkeley: USENIX Association, 2010: 1-5.

[20] WUNDSAM A, LEVIN D, SEETHARAMAN S, et al. OFRewind: enabling record and replay troubleshooting for networks[C]//Usenix Conference on Usenix Technical Conference. Berkeley: USENIX Association, 2011: 1-6.

[21] 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.

[22] 周启钊, 于俊清, 李冬. SDN 环境下 SAVI 动态配置技术研究[J]. 通信学报, 2018, 39(S1): 235-243.
ZHOU Q Z, YU J Q, LI D. Dynamic source address validation in software defined network[J]. Journal on Communications, 2018, 39(S1): 235-243.

[23] 王首一, 李琦, 张云. 轻量级的软件定义网络数据分组转发验证[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.

[24] DHAWAN M, PODDAR R, MAHAJAN K, et al. SPHINX: detecting security attacks in software-defined networks[C]//Proceedings 2015 Network and Distributed System Security Symposium. Piscataway: IEEE Press, 2015: 1-15.

[25] SASAKI T, PAPPAS C, LEE T, et al. SDNsec: forwarding accountability for the SDN data plane[C]//2016 25th International Conference on Computer Communication and Networks. Piscataway: IEEE Press, 2016: 1-10.

[26] 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.

[27] HESS F. Efficient identity based signature schemes based on pairings[C]//Selected Areas in Cryptography. 2003: 310-324.

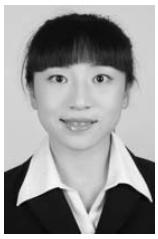
[28] HAGERUP T, RÜB C. A guided tour of chernoff bounds[J]. Information Processing Letters, 1990, 33(6): 305-308.

[29] LYNN B. On the implementation of pairing-based cryptosystems[D]. Stanford: Stanford University, 2007.

[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。



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。