

## 轻量级软件定义边缘计算网络拓扑发现方法

周启钊<sup>1</sup>, 石中煜<sup>2</sup>

(1. 成都信息工程大学计算机学院, 四川 成都 610225; 2. 成都职业技术学院, 四川 成都 610041)

**摘要:** 现有软件定义边缘计算网络(SD-ECN)已实现高效、可靠、安全的网络通信, 但面对物联网、人工智能、大数据等不同规模、不同需求的网络环境, 其拓扑架构的弹性和可扩展性仍存在不足。在软件定义网络中, 开放流发现协议(OFDP)通过封装链路层发现协议(LLDP)包, 由控制器发送和接收相关的报文, 并解析生成网络拓扑。边缘设备通过高速链路相互连接形成边缘网络, 并通过一个或多个汇聚节点连接到核心网络。在差异化技术网络环境需求下, 边缘节点的更新率通常较高, 而核心骨干交换机则无须重复更新。在此背景下, 现有软件定义边缘计算网络拓扑发现方法存在冗余链路查找和拓扑重构效率低的问题, 最终导致控制器不堪重负、边缘计算任务卸载机制失效。为此, 提出了一种轻量级软件定义边缘计算网络拓扑发现方法。针对冗余链路查找问题, 通过多级流表规则和LLDP帧优化实现了数据包的动态绑定、动态验证和实时转发; 针对控制器在拓扑发现过程中下发报文数量过多的问题, 提出了一种基于最小顶点覆盖思想的边缘节点和链路发现算法; 针对控制器接收报文数量过多问题, 提出了一种动态单双向边缘链路解析与合并算法。实验结果表明, 与现有网络拓扑发现方法相比, 所提方法在实现拓扑重构过程中解析链路容量和节点间的连接关系功能的基础上实现了轻量级的网络拓扑发现, 有效减少了网络数据包的数量和控制器的负载。

**关键词:** 边缘计算; 软件定义网络; 链路层发现协议; 拓扑重构

**中图分类号:** TN915.08

**文献标志码:** A

**DOI:** 10.11959/j.issn.2096-109x.2025071

## Lightweight software-defined edge computing network topology discovery method

ZHOU Qizhao<sup>1</sup>, SHI Zhongyu<sup>2</sup>

1. College of Computer Science and Technology, Chengdu University of Information Technology, Chengdu 610225, China

2. Chengdu Polytechnic, Chengdu 610041, China

**Abstract:** Software-defined edge computing networks enable efficient, reliable and secure network communication. The topological architecture of software-defined edge computing networks requires flexibility and scalability to adapt to the technological environments including Internet of things, artificial intelligence and big data. Edge computing devices are interconnected through high-speed links, core network and converging nodes. In software-defined networks, the OpenFlow discovery protocol (OFDP) utilizes link layer discovery protocol (LLDP) packets for sending and receiving related messages and generating network topology. The edge node update rate is typically

收稿日期: 2025-03-08; 修回日期: 2025-09-10

通信作者: 周启钊, zhouqz@cuit.edu.cn

基金项目: 重载快捷大功率电力机车全国重点实验室开放课题(QZKFKT2025-06); 山区河流保护与治理全国重点实验室开放课题(SKHL2413)

**Foundation Items:** The Opening Project from the State Key Laboratory of Heavy-Duty and Express High-Power Electric Locomotive(QZKFKT2025-06), The Opening Project from the State Key Laboratory of Hydraulics and Mountain River Engineering (SKHL2413)

high under different scales and requirements while the core backbone convergence switches do not require repeated updates. Existing topology discovery schemes for software-defined edge computing networks consequently exhibit low efficiency in redundant link identification and topology reconstruction, ultimately causing controller overload and affecting edge computing task offloading mechanisms. To resolve these issues, a lightweight topology discovery method for software-defined edge computing networks was proposed. For redundant link discovery challenges, dynamic packet binding, verification and real-time forwarding were implemented through multi-level flow table rules and optimized LLDP frame protocols. For excessive message transmission during topology discovery in SDN controllers, an edge node and link discovery algorithm based on minimum vertex cover concept was developed. Additionally, an MVC-based dynamic single/double-directional edge link resolution and merging algorithm was introduced to alleviate message overload in SDN controllers. Experimental results demonstrated that compared with existing network topology discovery algorithms, the proposed method achieved lightweight network topology discovery while maintaining capabilities for resolving link capacities and inter-node connections during reconstruction. This approach effectively reduces data packet volume within the network and decreases controller workload.

**Keywords:** edge computing, software-defined network, LLDP, topology reconstruction

## 0 引言

随着物联网、人工智能、大数据等技术的快速发展,大量应用场景对低时延、高带宽的计算和传输能力的需求与日俱增,传统的云计算模式已经无法满足需求。边缘计算作为一种新兴的计算模式,通过将计算、存储和网络资源部署在靠近用户或数据源的边缘位置,可以有效减少网络时延,提高数据传输效率<sup>[1]</sup>。传统车载自组织网络(vehicular Ad-hoc network, VANET)通常采用分布式架构实现移动边缘计算(mobile edge computing, MEC)网络,每个车载单元独立进行路由选择,因此仅能实现局部最优。软件定义网络(software defined network, SDN)在路由选择上具有全局视图,在VANET中引入SDN可以从根本上解决路由选择盲目性的问题。软件定义边缘计算网络(software defined edge computing network, SD-ECN)将SDN与MEC相结合,通过对网络控制和计算资源进行虚拟化和集中化管理,实现网络资源的灵活配置和边缘计算资源的高效利用,为提供更快速、可靠和安全的网络服务开拓了全新的视野<sup>[2]</sup>。

SD-ECN主要由边缘计算节点、网络控制器与中心控制器三大组件构成<sup>[3]</sup>。其中,边缘计算节点(如物理服务器、虚拟机或容器)根据中心控制器的指令执行计算任务调度与资源管理,其计算数据的交互与SD-ECN的拓扑结构密切相关。为提高网络性能与可靠性,底层边缘设备通

常按分布式拓扑进行组织,每台边缘设备均集成SDN交换机和边缘计算节点。这些边缘设备通过高速链路相互连接,形成边缘网络<sup>[4]</sup>。在中间层,边缘网络通过一个或多个汇聚节点接入核心网络,该层负责全网的控制和管理,并提供对外的连接能力。为满足差异化技术需求,边缘节点需直接向终端设备提供实时服务,以应对设备频繁移动和业务负载波动;而核心层则专注于全局流量调度,其稳定性直接决定了网络服务质量。

当前,SD-ECN正加速向通信与计算深度融合的方向演进,其核心挑战在于实现动态边缘节点之间的拓扑实时感知与资源协同调度<sup>[5]</sup>。无线算力网络依托通算一体化架构支持多级算力协同,但其有效运行依赖高效的拓扑发现机制。具体而言,VANET中的边缘节点具有极高的网络更新频率(高时变拓扑),而核心骨干交换机的拓扑则相对稳定、更新频率较低(低时变拓扑)<sup>[6]</sup>。受此差异化拓扑变化影响,SD-ECN中被动式流表的生存周期通常较短,车辆间数据传输时新旧流表具有不同的更新需求。相较于传统边缘计算节点需对底层网络执行重复更新的模式,现有SD-ECN架构通过对底层边缘节点进行抽象化处理,有效降低了拓扑层面的更新频率。

为实现高效、可靠且安全的网络通信,SD-ECN的拓扑结构需具备弹性与可扩展性,以适配物联网、人工智能、大数据等不同规模、不同需求的网络应用环境。针对当前低时变拓扑的应用现状,现有研究大多聚焦于特定的应用服务模

式, 采用基于全网广播的探测机制 (如 OpenFlow 全端口扫描), 依赖频繁的链路层发现协议 (link layer discovery protocol, LLDP) 帧泛洪完成链路发现。此类方法在计算资源分布不均衡的场景下, 缺乏对边缘协作流程的层次化设计与集群拓扑规划<sup>[7]</sup>, 易造成网络负载显著增加。与此同时, 基于集中式控制器轮询的方法 (如控制器周期探测), 存在单点计算压力过大、拓扑重构时延显著的问题。当 SD-ECN 承载的应用类型增多时, 重复的链路信息获取与更新操作需依赖控制器的 packet\_out 消息完成, 不仅导致网络传输效率低下, 还会出现链路发现时间随拓扑深度递增的现象, 同时该类方法难以与原生的开放流发现协议 (openflow discovery protocol, OFDP) 模式实现有效兼容。综上, 现有 SD-ECN 拓扑发现方法存在链路查找冗余度高、拓扑重构效率低的缺陷, 容易导致控制器负载过载及边缘计算任务卸载机制失效<sup>[8]</sup>。鉴于此, 本文提出一种优化的拓扑发现方法, 旨在解决现有机制中上述亟待突破的关键问题。

本文的主要贡献如下。

1) 提出了基于 LLDP 帧优化的多级流表拓扑发现模式, 实现了 SD-ECN 数据包的动态绑定、动态验证与实时转发, 在链路伪造场景下保障了边缘节点的安全接入, 为边缘计算拓扑重构及任务卸载的服务质量提供了重要保障。

2) 提出了基于最小顶点覆盖思想的边缘节点链路发现算法和动态单双向边缘链路解析与合并算法, 将控制器作为拓扑重构的请求枢纽, 提供 SD-ECN 运行中高可用的动态一致性模型, 有效减少网络报文的数量及控制器的负载。

3) 仿真分析表明, 本文所提的轻量级 SD-ECN 拓扑发现方法使控制器交互数据包数量平均减少 70% 以上, 绑定模式在理论上提升了抵御链路伪造攻击的能力, 在提高拓扑发现效率的同时保持了安全标准。

## 1 相关工作

近年来, 国内外研究聚焦于拓扑发现机制的动态响应能力优化。SDN 控制器通过定期向所有交换机的活跃端口下发 LLDP 报文的方式获取全局拓扑视图。SD-ECN 继承了 SDN 控制器的灵活重

构网络拓扑的能力。SD-ECN 在重建网络拓扑时, 需要对指定网络的活跃设备进行检查并获取其基本信息。控制器采用 packet\_in 与 packet\_out 机制收发 LLDP 报文, 通过灵活配置与动态调整网络资源, 实现网络拓扑的快速重构与扩展<sup>[9]</sup>。在边缘计算场景中, 针对单域业务, 边缘网络可以根据业务带宽需求主动触发网络拓扑重构, 调整链路容量与节点连接关系, 从而实现网络资源的按需分配与利用率提升<sup>[10]</sup>。对于边云协同、多云计算等跨域业务, 则可以通过多域协调器对相关域内拓扑进行重构, 从而降低承载新跨域业务的开销。

现有集中式 SDN 控制器在资源充足时, 能够在任意两个节点之间建立任意粒度的 IP 链路, 实现链路容量与节点连接关系的实时灵活调整。OFDP 是一种链路发现机制, 其原理是: SDN 控制器下发 LLDP 探测报文, 再收集交换机返回的报文, 解析报文中的设备与端口信息, 从而实现网络链路的自动化发现与拓扑绘制<sup>[11]</sup>。在交换机间存在双向链路的场景下, 当进行拓扑重建时, OFDP 的发包数量为所有端口数之和, 收包数量为链路数的两倍。OFDPv2<sup>[12]</sup>对 OFDP 进行了改进, 将每个端口发送一个 LLDP 报文改为每台交换机发送一个 LLDP 报文, 并由核心交换机复制转发至各边缘节点, 从而减少了发包数量。然而, 在 SD-ECN 中进行拓扑重构, 要求边缘节点必须具备较高的更新频率。OFDP 和 OFDPv2 因重复获取边缘链路信息而效率受限, 同时也缺乏对各类拓扑伪造攻击的有效防护。高效安全链路发现 (ESLD)<sup>[13]</sup>算法对 OFDPv2 进行了优化, 采用 HashMac 加密与端口过滤技术对所有的 LLDP 报文进行加密, 避免 LLDP 报文被发送给主机, 从而显著提升了安全性。在 SD-ECN 中利用该方法进行拓扑重构, 发包数降至与实际交换机数量一致, 收包数仍为链路数的两倍, 但加解密过程为控制器带来了额外开销。为保持拓扑重建效率, 高效安全链路发现 (ESLD) 需在一定程度上牺牲边缘服务的卸载效率。LADP<sup>[14]</sup>算法的发包方式与 OFDPv2 相同, 但设计了新的 LLDP 帧结构, 并通过在帧中设置随机数来防范广播风暴<sup>[15]</sup>。OFDPv3<sup>[16]</sup>对 OFDPv2 进行了优化, 将每台交换机发送一个 LLDP 报文改为全网仅发送一个 LLDP

报文，并采用组播方式进行全链路发现。当SD-ECN中的应用类型较多时，上述方法的链路发现时间随拓扑深度增加而增长；同时，这些方法无法有效防护拓扑伪造攻击。

表1对上述方法进行了对比，从中可以看出，现有方法多针对特定服务模式进行优化。SD-ECN拓扑重构的应用模型通常属于计算密集型，VANET链路故障可能导致控制器重新路由，产生大量packet\_in/packet\_out交互消息，此时控制器需动态计算数据流的重路由路径、选择备用路由并更新流表项。现有技术尚无法完全避免数据重路由与拓扑重构之间的资源竞争<sup>[17]</sup>。当SD-ECN应用类型增多时，重复获取链路信息会导致网络效率下降，更新消息依赖控制器packet\_out消息的重复拉取，链路发现时间随拓扑深度的增加而递增，同时缺乏与原生OFDP模式的有效兼容。

表1 现有方法对比

| 方法                          | 优势环境  | 重构效率 | 加密校验 | 负载 |
|-----------------------------|-------|------|------|----|
| OFDP/OFDPv2 <sup>[12]</sup> | 高时变拓扑 | 较低   | ×    | 高  |
| ESLD <sup>[13]</sup>        | 较稳定拓扑 | 较高   | √    | 一般 |
| LADP <sup>[14]</sup>        | 高时变拓扑 | 一般   | ×    | 较高 |
| OFDPv3 <sup>[16]</sup>      | 高时变拓扑 | 一般   | ×    | 较高 |
| 本文方法                        | 较稳定拓扑 | 高    | √    | 较低 |

## 2 系统模型

本文提出了一种基于LLDP帧优化、链路信

息绑定表和最小顶点覆盖（minimum vertex cover, MVC）思想的轻量级SD-ECN拓扑发现方法，其系统架构如图1所示。

该方法的拓扑重构机制基于控制器和交换机的交互模式实现，通过构造LLDP帧与链路信息绑定表，使OpenFlow虚拟交换机（OVS）将这些帧传输至单跳相邻交换机。通过接收LLDP帧与链路信息绑定表，每台交换机都可以识别其直接连接的交换机。当相邻交换机连接或断开时，交换机会向控制器实时报告VANET拓扑变更。在进行信息绑定表识别时，利用轻量级的双向链路解析和链路合并提供动态多级流表控制报文转发功能。该方法在实现SD-ECN拓扑重构中解析链路容量与节点间的连接关系功能的同时，有效减少了网络中数据包数量及控制器负载。

如图2所示，轻量级SD-ECN拓扑发现模块包含数据平面和控制平面两部分。数据平面主要包括LLDP发送模块、LLDP接收模块和物理端口，负责数据的收发和处理。控制平面则通过基于SDN的LLDP流表管理模块进行统一管理，并实现与拓扑数据库（SDN控制器）的交互。基于SDN的LLDP流表管理模块作为整个系统的核心，负责监听packet\_in消息以获取网络中的链路信息。这些信息将被用于链路发现与拓扑保护应用，进而实现SD-ECN拓扑重构应用。LLDP发送和接收模块作为数据平面与控制平面之间的桥梁，通过物理端口发送和接收LLDP数据包，并

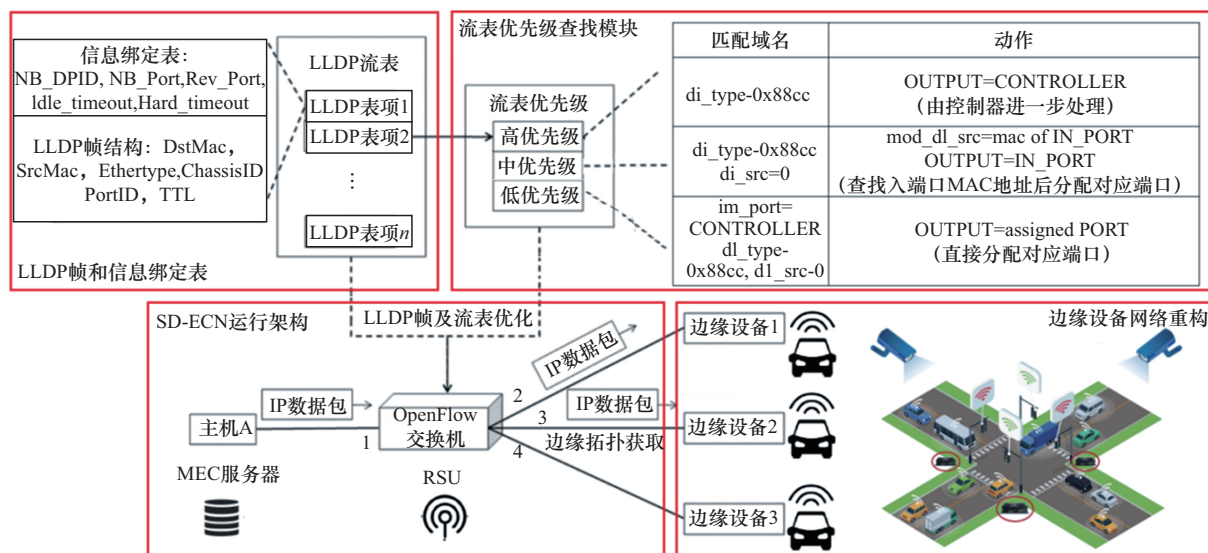


图1 轻量级SD-ECN拓扑发现方法系统架构

将其上报给基于 SDN 的 LLDP 流表管理模块进行处理。通过这些模块之间的紧密配合,系统能够实现高效的链路发现与拓扑管理,从而提供连贯、稳定且灵活的网络控制与管理能力。下文将分别描述图 2 中的模块设计和拓扑重构方法的实现细节。

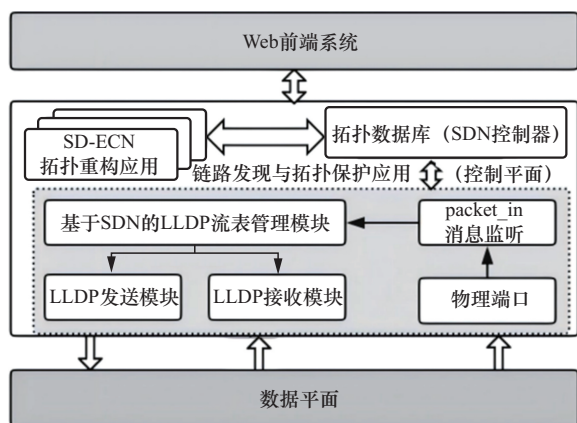


图2 轻量级SD-ECN拓扑发现模块

## 2.1 LLDP帧和信息绑定表

SD-ECN 拓扑重构的链路发现与拓扑保护应用模块基于 IEEE 802.1AB 构建。该模块主要负责从原始 LLDP 数据中提取信息,进而生成用于高效拓扑重构的 LLDP 帧和信息绑定表。LLDP 帧结构见表 2, 信息绑定表的结构见表 3。由于 OpenFlow 依赖数据路径 ID (DPID) 识别交换机, 将发送方的 DPID 分配给 Chassis\_ID 字段。在启用内置的 LLDP 解析模块后, 默认情况下每台交换机周期性地从每个活跃端口向直连交换机发送一个 LLDP 帧, 发送 LLDP 帧的时间间隔可以根据网络策略进行调整。

表2 LLDP 帧结构

| 源 MAC 地址 | 目的 MAC 地址  | 以太网类型     | 机箱 ID     |
|----------|------------|-----------|-----------|
| DstMac   | SrcMac     | Ethertype | ChassisID |
| 端口 ID    | TTL (生存时间) | 流表优先级     | 信息绑定表     |
| PortID   | TTL        | Priority  | 具体见表 3    |

信息绑定表用于记录拓扑重构中已识别的相邻交换机。为优化源地址验证效率, 本文方法基于 SDN 动态源地址验证的基本范式对 LLDP 帧进行改造, 将绑定关系转化为 OVS 可执行的流表

规则。通过将验证流程从 SDN 控制器分离并转移到多台交换机上, 借助交换机硬件的强大并行处理能力统筹完成源地址验证工作, 有效减轻控制器计算负载。该信息绑定表的每个条目包含 5 个字段, 即源交换机路径 ID (NB\_DPID)、源交换机端口 (NB\_Port)、接收端口 (Rev\_Port)、表项动态时延 (Idle\_timeout) 和表项静态时延 (Hard\_timeout)。每台相邻交换机的信息都被存储在一个流表项条目中。当交换机接收到相邻交换机发来的 LLDP 帧时, 它会实时调整对应表项的生存时间、表项动态时延和表项静态时延。该模块的实际处理周期包含链路绑定周期与解绑周期两个阶段, 其时长由 Idle\_timeout 与 Hard\_timeout 字段协同决定。

表3 信息绑定表

| 字段                    | 大小   |
|-----------------------|------|
| 源交换机路径 ID (NB_DPID)   | 8 字节 |
| 源交换机端口 (NB_Port)      | 2 字节 |
| 接收端口 (Rev_Port)       | 2 字节 |
| 表项动态时延 (Idle_timeout) | 2 字节 |
| 表项静态时延 (Hard_timeout) | 2 字节 |

在整合 LLDP 帧与信息绑定表的基础上, 本文方法结合端口信息绑定表与链路信息表的特征, 构建高、中、低 3 种等级的流表 (Priority) 并部署到对应的交换机上, 同时生成目标 LLDP 报文并下发给骨干交换机。当交换机收到 LLDP 数据包, 它会按照从高到低的优先级顺序匹配本地流表, 如图 3 所示。高优先级流表负责处理携带数字签名的 LLDP 帧, 通过匹配 “MAC 地址+端口号+时间戳” 动态绑定三元组, 精准筛选合法数据包; 中优先级流表负责处理已通过初步动态验证但尚未完成链路状态确认的 LLDP 数据包; 低优先级流表负责处理未绑定动态标识的常规 LLDP 探测包, 执行基本的转发规则。通过优先级递减的三级处理机制, 本文方法可以实现快速过滤非法数据包 (高优先级层直接拦截伪造帧)、渐进式状态验证 (中优先级层完成时效性校验) 和资源隔离保护。交换机借助预装的三级流表完成 LLDP 数据包的转发与预处理后, 相关数据包将被上传至控制器, 由控制器进行解析。

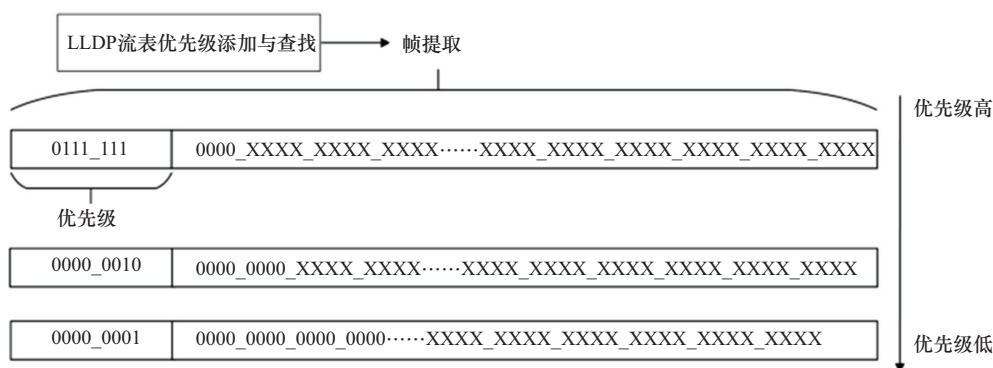


图3 LLDP流表按优先级进行处理

## 2.2 LLDP发送及接收模块

本文提出了一种基于LLDP的OpenFlow网络拓扑发现算法，该算法功能通过LLDP发送模块与LLDP接收模块协同实现。

LLDP发送模块的处理流程如图4所示。该模块首先利用OpenFlow默认的ovs-ofctl命令查询交换机的MAC地址、DPID和端口ID，随后将收集到的信息按照本文设计的LLDP帧格式填充到LLDP帧的各个字段中，完成封装后通过OVS的ovs\_vport\_send函数发送至对应端口。LLDP流表的超时行为由Idle\_timeout与Hard\_timeout字段共同控制。

LLDP接收模块的处理流程如图5所示。该模块首先处于等待状态，准备接收LLDP包。当接收到LLDP包时，该模块首先检查LLDP包中的以太网字段，完成LLDP包的有效性验证。在进行流表

匹配时，若未找到匹配条目，则新增相应LLDP流表项；若条目已存在，则将该条目的Idle\_timeout值更新为当前LLDP帧的TTL值。所有端口的维护操作通过OVS的ovs\_port\_modify函数实现，并依托LLDP流表中的Rev\_Port字段完成匹配条目的增、删、改、查操作。当流表超时或检测到端口关闭时，系统自动删除LLDP流表中对应的失效条目，从而确保表项数据的准确性。最终，LLDP流表中的条目数量将与交换机实际连接的相邻交换机数量保持一致。在链路解析过程中，若新接入的交换机接收到首个LLDP帧，该帧将被封装于packet\_in消息中，并通过相应端口转发至控制器。转发前，系统会检查当前队列中的数据包信息、更新计数器，并记录packet\_in队列状态，以便控制器对新接入交换机进行链路发现规则的配置与

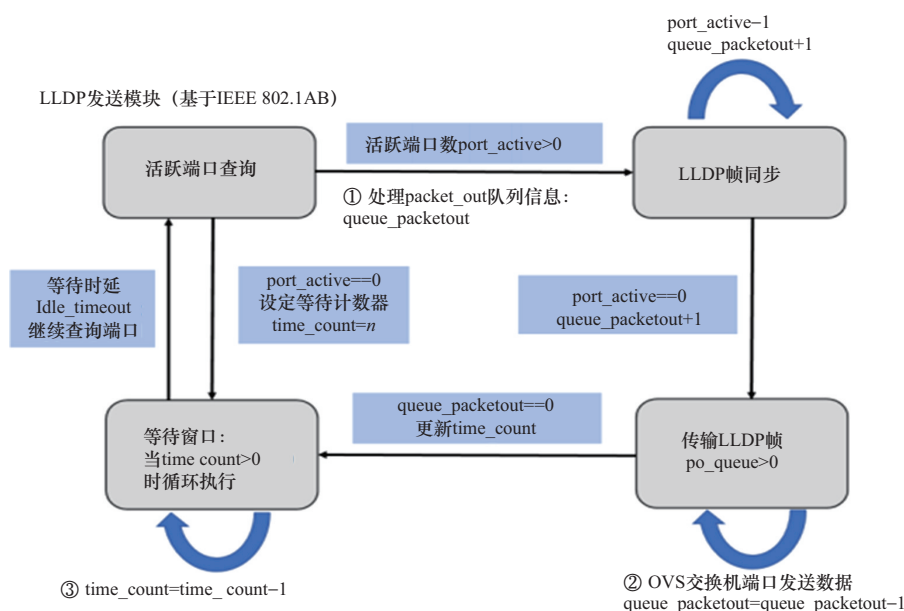


图4 LLDP发送模块的处理流程

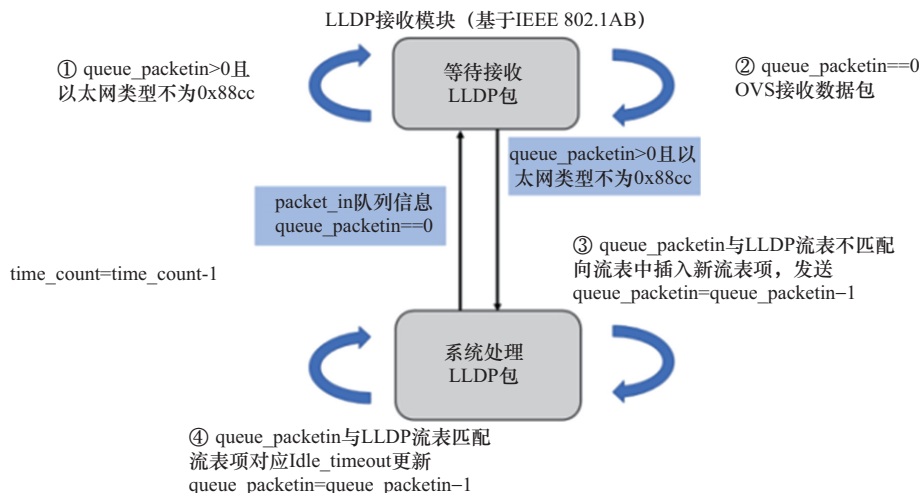


图5 LLDP接收模块的处理流程

管理。该机制确保新设备加入网络时能够被及时识别，进而提升网络的灵活性与可管理性。

## 2.3 基于最小顶点覆盖算法思想的链路解析

### 2.3.1 拓扑重构方法框架

针对SD-ECN当前低时变拓扑的现状，为优化OVS拓扑链接实时更新效率，本文提出一种基于最小顶点覆盖算法思想的双向链路解析和链路合并方法，其框架如图6所示。控制器建立交换机MAC地址与端口ID的映射关系，并对端口类型进行标识，为后续的骨干交换机选取及多级流表构造提供数据支撑；同时，根据端口状态变化、链路解析结果及超时情况同步更新链路信息表。控制器构造LLDP帧及信息绑定表，并依据当前的网络拓扑信息，采用静态或动态MVC算法选出骨干交换机集合。遴选出骨干交换机集合后，该框架结合端口信息表和链路信息表，为每台骨干交换机构造并下发相应的LLDP报文。为保障处理效率，控制器将在流表中插入高优先级规则，确保packet\_in消息能被即时处理，从而缩短整体发现时延。报文下发后，交换机依据预置的流表规则（如图3所示）对LLDP数据包进行转发处理。控制器收到packet\_in消息后，依次通过源MAC地址确定链路源端口、通过packet\_in来源确定目的端口，并通过比对ChassisID与上传packet\_in消息的交换机的标识，判断链路为单向或双向。解析完成后，控制器执行链路合并逻辑，并更新链路信息表与端口绑定信息表。每一轮链路发现结束后，控制器根据拓扑变

化情况，决定是否重新计算骨干交换机集合以启动下一轮发现。此外，当相邻交换机断开连接时，交换机会主动封装携带DPID的通知消息，通过packet\_in消息即时上报给控制器。至此，该双向链路解析和链路合并方法完成拓扑重构闭环。该方法在信息绑定表识别阶段，通过轻量级双向链路解析与链路合并策略，提供动态多级流表控制报文转发能力，既实现了SD-ECN拓扑重构过程中链路容量和节点连接关系解析功能，又有效减少了网络冗余数据包的数量和控制器的负载。

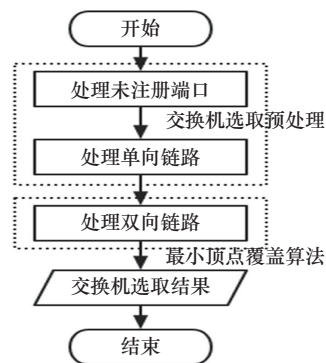


图6 基于最小顶点覆盖算法思想的双向链路解析和链路合并方法框架

### 2.3.2 链路解析与合并

在SD-ECN拓扑重构过程中，边缘节点及交换机被抽象为无向连通图。其中，交换机作为点，链路作为边，忽略主机。交换机通过LLDP报文与流表转发控制机制发现其所有相邻的链路。被选中发送LLDP包的交换机称为骨干交换

机。骨干交换机的选取问题可以转化为图的最小顶点覆盖问题。在大规模网络中,当拓扑局部发生变化时,采用静态算法计算MVC开销较大,且可能导致新覆盖集与原覆盖集产生较大的差别。为此,本文采用增量式MVC算法,利用变更前后的覆盖集求解新覆盖集,实现提升计算速度、减小变更开销、缩小变更范围的优化目标。在经典的贪心算法基础上,本文算法针对多个节点具有相同最大邻接度的情况进行了改进:将这些节点构成新的子图,进行二次贪心计算,选取邻接度最小的节点。与经典的贪心算法相比,本文算法的运行速度更快,且最小顶点覆盖集规模减少5%~10%。

SDN中存在双向和单向两种链路,控制器对两种链路的解析过程如图7所示。如图7(a)所示,交换机S1上传LLDP包,其中的Chassis ID

字段与S1自身标识一致,判定为双向链路。如图7(b)所示,S2上传LLDP包,其中的Chassis ID字段与S2标识不一致,判定为单向链路。该单向/双向链路发现机制利用了流表对LLDP包字段的重写能力。当交换机收到LLDP包时,会将其端口MAC地址写入SrcMac字段。控制器解析时,可以通过SrcMac字段推断出链路的源交换机,并通过分析给控制器发送包的交换机,确定链路的目的交换机,从而有效地解析出完整的链路信息。该方法显著提高了链路发现的效率,增强了网络管理的准确性与可靠性。

### 2.3.3 算法复杂度及安全性分析

本文假设在SD-ECN拓扑中,数据层由SDN交换机和主机设备组成,任意两台交换机之间均存在双向通信链路,即网络拓扑为连通图,且交

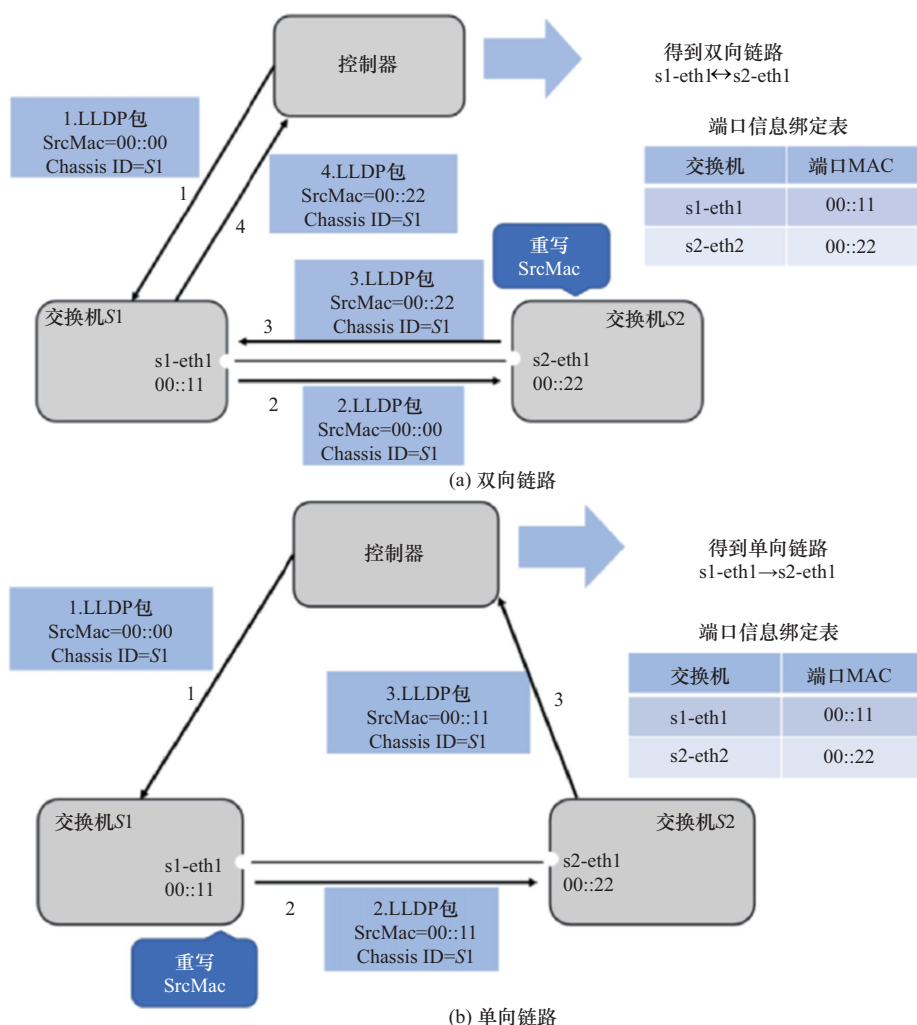


图7 链路解析示意

交换机的活跃端口连接的是主机或者双向链路。本文设交换机的数量为 $S$ ，主机设备数量为 $H$ ，链路数量为 $L$ 。研究人员基于图论模型计算出 OFDP/OFDPv2、ESLD、LADP 和 OFDPv3 方法的 packet\_in 与 packet\_out 包需求数量，结果见表 4。研究人员基于表 4 的数据推导出上述方法的时间与空间复杂度，结果见表 5。以本文方法为例，时间复杂度由选择最大邻接度交换机  $O(S^2)$ 、构造端口信息绑定表  $O(S)$ 、标记链路  $O(L)$  构成，故其总的时间复杂度为  $O(S^2+S+L)$ 。空间复杂度由存储端口表  $O(2L+H)$  和存储链路信息表  $O(L)$  构成，故其总的空间复杂度为  $O(3L+H)$ 。当 VANET 链路发生故障时，控制器因重新路由而产生大量交互消息，在此场景下本文方法的时间复杂度与 LADP 相当，而空间复杂度则低于其他方法，理论上可缓解控制器消息负载过高的问题。

表 4 现有拓扑发现方法 LLDP 报文需求理论数量

| 方法          | packet_in 数量      | packet_out 数量    |
|-------------|-------------------|------------------|
| OFDP/OFDPv2 | $L \times 2$      | $H + L \times 2$ |
| ESLD        | $(L, L \times 2)$ | $(1, S-1)$       |
| LADP        | $L \times 2$      | 1                |
| OFDPv3      | $L \times 2$      | $S$              |
| 本文方法        | $L$               | $(1, S-1)$       |

表 5 现有拓扑发现方法的时间与空间复杂度

| 方法          | 最大时间复杂度       | 最大空间复杂度    |
|-------------|---------------|------------|
| OFDP/OFDPv2 | $O(S^2+S+4L)$ | $O(6L+2H)$ |
| ESLD        | $O(S^2+S+2L)$ | $O(4L+H)$  |
| LADP        | $O(S^2+2L)$   | $O(4L+H)$  |
| OFDPv3      | $O(S^2+S+2L)$ | $O(5L+H)$  |
| 本文方法        | $O(S^2+S+L)$  | $O(3L+H)$  |

本文提出的 SD-ECN 拓扑重构机制在控制器下发与接收报文的过程中均具备轻量级特性，解决了现有研究方法在 SD-ECN 应用类型多样化场景下链路信息获取效率低的问题。下发报文阶段采用最小顶点覆盖算法选取交换机，并通过流表控制转发，避免重复转发而产生冗余报文。接收报文阶段针对单向/双向链路设计了新的链路解析规则，每条链路仅需解析一个报文，利用轻量级双向链路解析与链路合并机制提供动态的多级

流表控制报文转发功能，实现了交换机侧 LLDP 帧模型的轻量级合成。

本文所提轻量级拓扑发现方法在提升 SD-ECN 重构效率的同时，通过 LLDP 帧优化的完整性保护机制增强系统安全性。在 LLDP 帧优化过程中，借助多级流表规则实现数据包与边缘设备身份的动态绑定。通过动态绑定每个条目的 5 个字段，从原理上防范伪造帧注入攻击。基于时间戳的 LLDP 帧动态验证机制（超时自动丢弃）可规避重放攻击风险，确保拓扑信息的时效性与可信性。针对当前 SD-ECN 低时变拓扑的特点，该轻量级机制适用于动态业务需求场景，解析报文数量显著减少，能够实现高效且安全的拓扑重构。本文方法还可选择部分交换机下发 LLDP 报文，每条单向/双向链路仅解析一个 LLDP 报文，通过端口过滤减少主机接收 LLDP 帧；同时对部分 LLDP 报文进行动态加密，防止恶意伪造的拓扑重构信息。SD-ECN 拓扑发现算法实现了稳定拓扑背景下高效的链路发现。

### 3 实验与结果分析

#### 3.1 实验设置

为了评估本文方法的性能，研究人员在搭建的模拟 SD-ECN 环境中，将其与现有的代表性拓扑发现方法（见表 1）进行了对比。研究人员在服务器与 VMware（Ubuntu 16.04.2 系统）上构建 SDN 环境，服务器配置信息见表 6。实验采用 4 种典型拓扑结构（TOP 1~TOP 4），具体信息见表 7。

表 6 服务器配置信息

| 属性   | 配置信息                                                                                          |
|------|-----------------------------------------------------------------------------------------------|
| CPU  | Intel Xeon i7-6500U 2.5 GHz                                                                   |
| 内存   | 8 GB                                                                                          |
| 硬盘   | NVMe SAMSUNG MZVLV256                                                                         |
| 操作系统 | Windows 10                                                                                    |
| 软件版本 | JDK 1.8.0/Python 3.5.2/Floodlight 1.4/ VMware Workstation 14/Mininet 2.3.0/Open vSwitch 2.3.0 |

为评估本文算法在复杂动态拓扑场景下的有效性，研究人员设计了一种基于概率的状态空间动态扰动机制，模拟移动设备接入边缘服务器时所引发的拓扑重构（如车联网中 RSU 的动态连接）。本文搭建的模拟 SD-ECN 主要由边缘服务器与移动设备（mobile device, MD）组成。设  $m$  台边

缘服务器的集合为  $\mathcal{M} = \{1, 2, \dots, M\}$ 。MD 在 MEC 覆盖范围内自由移动，并随机计算密集型任务。系统采用长度为  $T$  的离散时间模型<sup>[18]</sup>，时间集合表示为  $\mathcal{T} = \{1, 2, \dots, T\}$ 。该机制会依据预设的概率值，对智能体接收到的状态空间中的任务数据施加符合伯努利分布的随机扰动，从而满足 SD-ECN 实际应用场景的真实性与复杂性要求。

表7 实验拓扑结构信息

| 名称    | 拓扑结构 | 交换机数量/台 | 主机数量/台 | 链路数量/条 | 骨干交换机数量/台 |
|-------|------|---------|--------|--------|-----------|
| TOP 1 | 星形   | 85      | 420    | 420    | 17        |
| TOP 2 | 树形   | 63      | 188    | 62     | 21        |
| TOP 3 | 线型   | 100     | 100    | 99     | 50        |
| TOP 4 | Mesh | 45      | 90     | 210    | 15        |

如图8所示，在该SD-ECN仿真环境中，部署了4台位置均匀分布的边缘服务器，构成一个MEC系统。同时，在系统的覆盖范围内设置了100个MD，MD的位置和运动轨迹在系统中随机生成。假设MD生成的计算密集型任务的数据量在[0.5, 10] MB区间内均匀分布，完成任务所需的CPU周期在[0.4, 8] GHz区间内均匀分布。为简化模型，假设边缘服务器的可用计算资源在[5, 20] GHz/s区间内均匀分布<sup>[19]</sup>。在SD-ECN动态拓扑收敛测试中，节点失效率设为1%~20%。

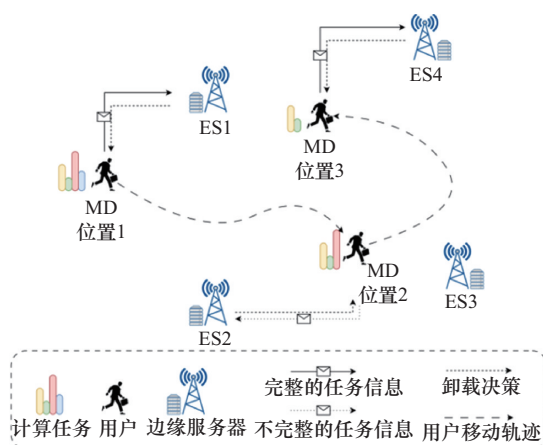


图8 SD-ECN仿真环境

### 3.2 通信开销对比

消息数量是指SD-ECN在边缘拓扑发现的单个周期内，控制器收发的LLDP包总数。根据OpenFlow协议，该类LLDP包数量可折算为

packet\_out与packet\_in消息的数量之和。在SD-ECN中，控制器在拓扑发现过程中处理的LLDP包数量直接影响高动态业务配置的处理效率。本实验按固定的拓扑发现周期进行轮询统计，在控制器的物理网卡端口上抓取每个周期进出控制器的OpenFlow消息数量，得到单个拓扑发现周期内需处理的LLDP包数量。

该实验默认在网络链路稳定、无丢包等条件下进行，拓扑发现周期为5 s，统计得到的各方法在160个周期（800 s）内所处理的消息数量如图9所示。针对本文方法在不同拓扑发现周期中LLDP包的校验模式差异，研究人员将其划分为链路绑定信息周期（en）与解绑周期（pla）两个阶段。其中，链路绑定信息周期对应前3个拓扑发现周期（累计15 s），解绑周期对应前3个拓扑发现周期之后的所有周期（785 s）。对两个周期的消息数量进行平均处理后，其所需处理的消息数量略高于解绑周期消息数量。

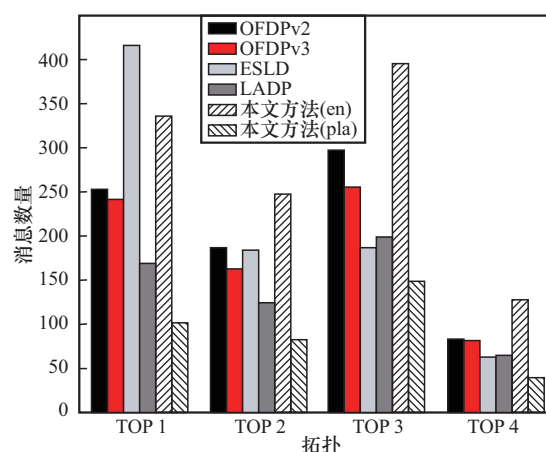


图9 各方法所处理的消息数量

拓扑发现过程所需处理的控制消息数量直接影响SD-ECN拓扑重构的流量开销。在现有OpenFlow机制下，携带定量LLDP包的packet\_out与packet\_in消息大小已知且固定，结合LLDP包数量与大小可以计算出当前拓扑发现周期的通信开销。本文统计了160个周期（800 s）所处理的控制消息数量。前15 s为链路绑定信息周期，之后的785 s为解绑周期。如图10所示，在15 s的链路绑定周期，本文方法采用校验式报文下发模式，其通信开销与OFDP系列方法相

当, 但高于 ESLD 与 LADP 方法。经过多个拓扑发现周期, 本文方法的平均综合所需处理的控制消息数量略高于解绑周期所需处理的控制消息数量, 并显著低于其他方法, 说明其通信开销小, 且所有消息均携带链路绑定信息与时间戳, 相较于采用明文传输模式的其他方法更具安全性与完整性, 符合理论分析预期 (即本文方法在实现 SD-ECN 拓扑重构中的链路容量与节点连接关系解析功能的同时, 有效减少了网络中数据包数量与控制器的负载)。

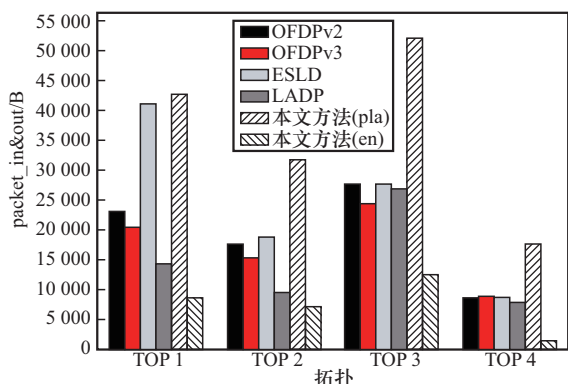


图10 控制消息数量对比

针对图9所示结果进行理想效率增益的综合分析 (具体见表8), 本文方法在多种典型拓扑下均表现出显著优势。在拓扑1和拓扑2中, 其理想效率增益分别达到77.6%与69.1%, 明显高于OFDP/OFDPv2、ESLD及LADP等对比方案。即使在拓扑3与拓扑4等更复杂的场景中, 本文方法的效率增益(61.2%、50.6%)仍保持领先, 稳定优于其他方法。这进一步验证了本文方法的轻量级机制在降低通信开销及方面的有效性。

表8 理想效率增益对比

| 名称  | OFDP/OFDPv2 | OFDPv3 | ESLD  | LADP  |
|-----|-------------|--------|-------|-------|
| 拓扑1 | 61.8%       | 77.6%  | 55.4% | 40.7% |
| 拓扑2 | 71.3%       | 69.1%  | 57.8% | 41.8% |
| 拓扑3 | 67.4%       | 61.2%  | 62.7% | 39.2% |
| 拓扑4 | 52.6%       | 50.6%  | 57.3% | 38.6% |

### 3.3 CPU时间开销对比

SD-ECN资源由控制器集中管理与调配, 而CPU的时间开销与拓扑发现过程中控制器的负载承受能力息息相关, CPU时间开销是衡量拓扑发

现方法效率的关键指标之一。

各方法前3个周期的CPU时间开销对比如图11所示。本文方法在第1个拓扑发现周期需要进行决策, 并下发、接收和处理携带链路绑定信息的LLDP包, 因此CPU时间开销相对较大。本文方法在第1个周期获得核心拓扑链路信息后, 选取骨干交换机集合, 并向每台骨干交换机下发不携带链路绑定信息的LLDP报文进行拓扑发现。若拓扑结构未发生显著变化, 则后续周期无须重复进行决策, 可复用上一周期的LLDP报文。随着周期的推进, 本文方法控制器下发与处理的消息数量显著减少, 进而大幅减少了拓扑发现的CPU时间开销。各方法的完整多周期CPU时间开销对比如图12所示, 可以看出, 本文方法在后续周期内的累计CPU时间开销小于现有方法。此外, 图13展示了各方法在4种拓扑场景下的CPU时间开销对比情况, 可以看出, 本文方法在4种拓扑场景下的CPU时间开销均小于其他方法。

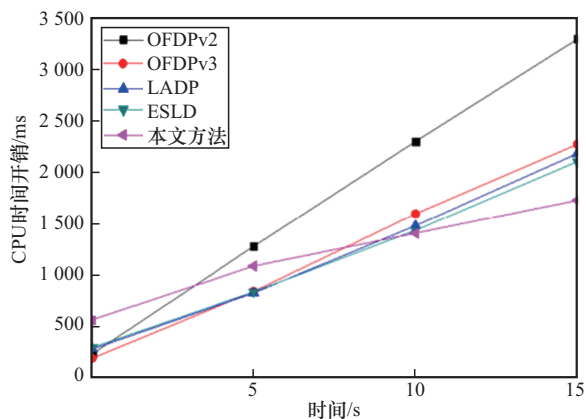


图11 各方法前3个周期的CPU时间开销对比

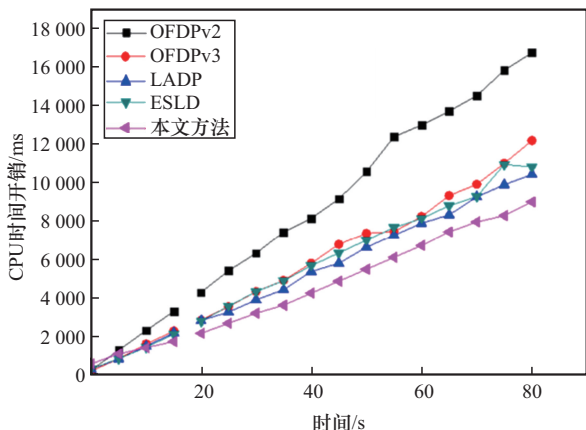


图12 各方法完整多周期的CPU时间开销对比

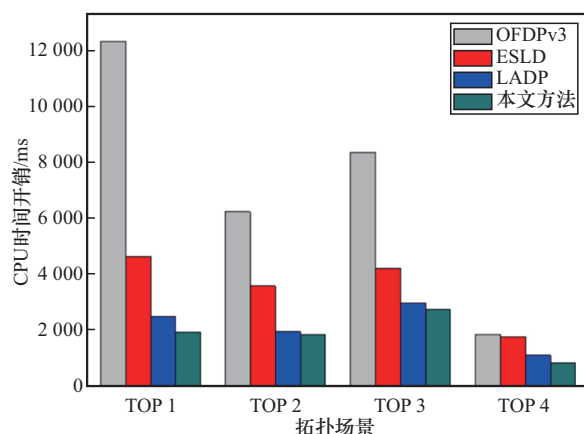


图13 各方法在4种拓扑场景下的CPU时间开销对比

### 3.4 在动态场景下的鲁棒性对比

针对SD-ECN边缘节点或链路失效的动态场景，图14对比了各方法的失效响应收敛时间。OFDPv2的收敛时间随节点失效率的提升呈指数级增长，这一现象是其协议头冗余与状态机迭代机制的缺陷导致的；相比之下，OFDPv3通过引入轻量化拓扑发现机制，在同一失效率区间内收敛时间增幅明显减小。LADP的收敛时间增速随失效率增加呈线性增长，ESLD通过特征空间对齐技术有效抑制了节点失效引发的拓扑波动，尤其是在15%~20%高失效率区间表现出稳定的低时延特性。本文方法在收敛时间方面明显优于OFDPv2与OFDPv3，与LADP和ESLD的差异不显著。本文方法的收敛时间增速较平缓（失效率每提升5%，收敛时间仅增加12.7 ms），证明其在SD-ECN边缘节点失效的动态场景下具有较好的鲁棒性。

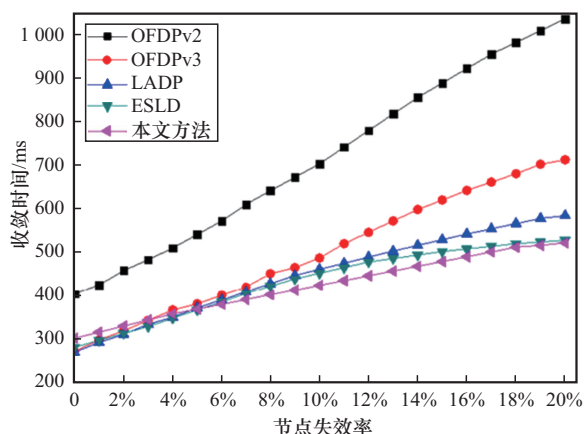


图14 失效响应收敛时间对比

## 4 结束语

在SDN集中控制架构下，高效的拓扑重构是提升SD-ECN服务质量与应用灵活部署的前提。针对现有SD-ECN拓扑发现方法存在冗余链路查找与拓扑重构效率低的问题，本文提出一种基于LLDP帧优化与最小顶点覆盖思想的轻量级SDN拓扑发现方法。控制器通过拓扑发现的基础服务获取网络视图，利用轻量级双向链路解析和链路合并机制，提供动态多级流表控制报文转发功能，实现交换机侧LLDP帧模型的轻量级合成。该方法在实现SD-ECN拓扑重构中链路容量与节点间连接关系解析功能的前提下，有效减少了网络中消息的数量与控制器的负载，并能良好地兼容原生OFDP模式。

## 参考文献：

- [1] VIJAYASEKARAN G, DURAI PANDIAN M. An efficient clustering and deep learning based resource scheduling for edge computing to integrate cloud-IoT[J]. *Wireless Personal Communications*, 2022, 124(3): 2029-2044.
- [2] LI C L, YU Z Q, LI X Y, et al. Low-latency AP handover protocol and heterogeneous resource scheduling in SDN-enabled edge computing[J]. *Wireless Networks*, 2023, 29(5): 2171-2187.
- [3] HUANG X L, SHI P, LIU Y F, et al. Towards trusted and efficient SDN topology discovery: a lightweight topology verification scheme[J]. *Computer Networks*, 2020, 170: 107119.
- [4] KONG D Z, SHEN Y, CHEN X, et al. Combination attacks and defenses on SDN topology discovery[J]. *IEEE/ACM Transactions on Networking*, 2023, 31(2): 904-919.
- [5] HE T Z, TOOSI A N, BUYYA R. Efficient large-scale multiple migration planning and scheduling in SDN-enabled edge computing[J]. *IEEE Transactions on Mobile Computing*, 2024, 23(6): 6667-6680.
- [6] WU J J, DONG F, LEUNG H, et al. Topology-aware federated learning in edge computing: a comprehensive survey[J]. *ACM Computing Surveys*, 2024, 56(10): 1-41.
- [7] LYU Z W, LI P F, WEI Z C, et al. A multi-edge jointly offloading method considering group cooperation topology features in edge computing networks[J]. *Peer-to-Peer Networking and Applications*, 2024, 17(6): 3507-3525.
- [8] 郑正, 徐明伟, 李琦, 等. SDN网络拓扑污染攻击防御机制研究[J]. *计算机研究与发展*, 2018, 55(1): 207-215.  
ZHENG Z, XU M W, LI Q, et al. Defending against SDN network topology poisoning attacks[J]. *Journal of Computer Research and Development*, 2018, 55(1): 207-215.
- [9] 肖军弼, 孟祥泽, 田爱宝, 等. SDN网络中基于拓扑分域的故障恢复方法[J]. *计算机工程与设计*, 2021, 42(6): 1501-1508.  
XIAO J B, MENG X Z, TIAN A B, et al. Failover method based on

- topology subarea in SDN network[J]. Computer Engineering and Design, 2021, 42(6): 1501-1508.
- [10] DAPPER E SILVA T, EMYGDIO DE MELO C F, CUMINO P, et al. STFANET: SDN-based topology management for flying ad hoc network[J]. IEEE Access, 2019, 7: 173499-173514.
- [11] NAIN A, SHEIKH S, SHAHID M, et al. Resource optimization in edge and SDN-based edge computing: a comprehensive study[J]. Cluster Computing, 2024, 27(5): 5517-5545.
- [12] PAKZAD F, PORTMANN M, TAN W L, et al. Efficient topology discovery in OpenFlow-based software defined networks[J]. Computer Communications, 2016, 77: 52-61.
- [13] ZHAO X, YAO L, WU G W. ESLD: an efficient and secure link discovery scheme for software-defined networking[J]. International Journal of Communication Systems, 2018, 31(10): e3552.
- [14] JIA Y Z, XU L, YANG Y W, et al. Lightweight automatic discovery protocol for OpenFlow-based software defined networking[J]. IEEE Communications Letters, 2020, 24(2): 312-315.
- [15] CHANG Y C, LIN H T, CHU H M, et al. Efficient topology discovery for software-defined networks[J]. IEEE Transactions on Network and Service Management, 2021, 18(2): 1375-1388.
- [16] 李冬, 于俊清, 谷永普, 等. 高效安全的软件定义网络拓扑发现协议[J]. 网络与信息安全学报, 2023, 9(6): 20-33.
- LI D, YU J Q, GU Y P, et al. Efficient and safe software defined network topology discovery protocol[J]. Chinese Journal of Network and Information Security, 2023, 9(6): 20-33.
- [17] 刘亚群, 邢长友, 高雅卓, 等. TopoObfu: 一种对抗网络侦察的网络拓扑混淆机制[J]. 计算机科学, 2021, 48(10): 278-285.
- LIU Y Q, XING C Y, GAO Y Z, et al. TopoObfu: a network topology obfuscation mechanism to defense network reconnaissance[J]. Computer Science, 2021, 48(10): 278-285.
- [18] 李国燕, 薛翔, 刘毅, 等. 改进 TD3 的 SDN 车联网边缘计算卸载策略[J]. 计算机集成制造系统, 2023, 29(5): 1627-1634.
- LI G Y, XUE X, LIU Y, et al. Improved TD3 edge computing offloading strategy for software defined networking Internet of vehicles[J]. Computer Integrated Manufacturing Systems, 2023, 29(5): 1627-1634.
- [19] WANG L Y, ZHANG G L. Deep reinforcement learning based joint partial computation offloading and resource allocation in mobility-aware MEC system[J]. China Communications, 2022, 19(8): 85-99.

#### [作者简介]



周启钊 (1991-), 男, 湖南长沙人, 博士, 成都信息工程大学计算机学院副教授, 主要研究方向为软件定义网络、边缘计算、联邦学习和目标检测。



石中煜 (1995-), 女, 青海西宁人, 博士, 成都职业技术学院讲师, 主要研究方向为神经网络、数字孪生和智慧医疗。