

高性能网络数字孪生仿真引擎研究

石鸿伟^{1,2}, 倪中阳², 陆千沂², 黄韬^{2,3}

(1. 东南大学网络空间安全学院, 江苏 南京 211189; 2. 紫金山实验室, 江苏 南京 211111;
3. 北京邮电大学网络与交换技术全国重点实验室, 北京 100876)

摘要: 网络数字孪生技术在提升 IP 承载网仿真测试的运维效率与决策精度方面具有重要作用, 然而, 当前仍面临仿真精度不高与仿真性能不足等问题。基于 PNetLab 仿真平台, 提出了一种高性能网络数字孪生仿真引擎——ePNetLab (extended PNetLab)。首先, 对原有平台进行性能优化与功能扩展, 改进接口响应机制, 设计跨节点通信方案, 提升拓扑构建效率并增强集群化组网能力。其次, 设计并实现基于社区划分算法的拓扑动态构建方法, 有效地降低了大规模仿真场景的构建时间与资源开销。最后, 通过实验评估验证了所提方案的可行性与高效性。实验结果表明, ePNetLab 在拓扑构建效率方面相较于原生 PNetLab 在最优条件下提升了 82.9%; 同时, 所引入的社区划分算法在仿真效率、资源利用率与业务性能等方面较其他算法有较大提升。

关键词: IP 承载网; 数字孪生网络; 仿真引擎; 社区分割算法

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2026.00505

Research on high-performance network digital twin simulation engine

Shi Hongwei^{1,2}, Ni Zhongyang², Lu Ganyi², Huang Tao^{2,3}

1. School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China

2. Purple Mountain Laboratories, Nanjing 211111, China

3. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: Network digital twin technology plays a significant role in improving the maintenance efficiency and decision-making accuracy of IP bearer network simulation and testing. However, it still faces challenges, such as low simulation accuracy and insufficient simulation performance. A high-performance network digital twin simulation engine, extended PNetLab (ePNetLab), based on the PNetLab simulation platform was proposed. Firstly, the original platform was optimized in terms of performance and functionality. The interface response mechanism was improved, a cross-node communication scheme was designed, and the efficiency of topology construction was enhanced, along with the ability to form clustered networks. Secondly, a topology dynamic construction method based on community detection algorithms was designed and implemented, effectively reducing the construction time and resource overhead in the large-scale simulation scenarios. Finally, experimental evaluations were conducted to verify the feasibility and efficiency of the proposed solution. The experimental results show that ePNetLab improves the topology construction efficiency by 82.9% compared with the native PNetLab under the optimal conditions. Meanwhile, the community partitioning algorithm introduced greatly improves simulation efficiency, resource utilization, and business performance compared with the other algorithms.

Key words: IP network, digital twin network, simulation engine, community partitioning algorithm

收稿日期: 2025-05-23; 修回日期: 2025-07-05

通信作者: 石鸿伟, shihongwei@pmlabs.com.cn

基金项目: 国家重点研发计划 (No. 2022YFB2702303)

Foundation Item: The National Key Research and Development Program of China (No. 2022YFB2702303)

0 引言

随着信息技术的飞速发展，IP 承载网作为现代通信网络的核心组成部分，在支撑海量数据传输、保障业务连续性以及实现复杂网络功能方面发挥着至关重要的作用。为了更好地应对网络的动态变化、优化资源配置以及提升运维效率，网络仿真技术应运而生。通过构建虚拟的网络环境，对真实网络的运行状态、性能指标以及潜在问题进行模拟与分析，为网络规划、故障排查以及优化决策提供了有力的技术支持。然而，现有的仿真软件在实际应用中仍存在诸多的不足之处，限制了其在复杂网络环境中的有效应用。

首先，仿真精度不够是当前仿真软件面临的主要问题之一^[1-2]。现有的仿真工具在模拟网络行为时，通常难以精确地反映真实网络中的复杂交互关系、动态变化以及设备特性。在数值仿真层面，模型精度不足问题尤为突出，由于不同厂商设备的内部逻辑、协议实现存在显著的差异，且复杂网络交互的动态特性难以通过简化数学模型完整刻画，导致仿真结果与真实设备行为出现系统性偏差，这种精度上的偏差可能导致仿真结果与实际运行情况存在较大差异，从而影响基于仿真结果的决策准确性。其次，仿真性能不足也是制约现有仿真软件应用的重要因素^[3-4]。随着网络规模的不断扩大和复杂度的增加，传统的仿真工具在处理大规模网络拓扑时通常面临着性能瓶颈，导致仿真构建时间过长、资源消耗过大甚至无法完成仿真任务，在模拟仿真层面，尽管能生成真实网络环境，但其规模扩展性受制于硬件资源瓶颈，大规模拓扑仿真面临计算集群资源占用过高、并行化效率低等问题。

围绕上述网络仿真问题，相关领域展开了诸多研究，现有研究通过精细化建模部分缓解了精度不足的问题，但在多厂商设备协议差异兼容性、动态交互行为刻画等方面仍存在显著的局限；同时，针对性能瓶颈的解决方案虽能扩展仿真规模，却难以平衡计算效率与模型保真度，导致大规模场景中精度与性能难以协同优化。

基于此，首先，本文深入地研究对比了多种通用及专用网络仿真软件，详细总结了现有仿真工具在仿真精度、规模、性能等方面的优劣势，结合 IP 承载网复杂网络环境及实际应用需要，确定以 PNetLab^[5]

为基础构建仿真引擎。其次，对 PNetLab 进行性能优化与功能拓展，分析与改进接口响应参数，补充集群化功能，提升其在拓扑变更处理方面的效率，新增大规模组网能力。接着，设计并实现了基于 PNetLab 的高性能仿真引擎 ePNetLab (extended PNetLab)，引入社区分割算法进行拓扑动态构建，通过自动化分割与资源预分配来优化与缩短大规模仿真环境构建时间。最后，实验验证了 ePNetLab 在拓扑构建效率、仿真效率、资源利用率及业务性能上有显著优势。

PNetLab 平台作为一种先进的模拟仿真技术，凭借独特的虚拟化技术运行网元镜像，无须复杂的协议建模，功能上能够高度对标真实网元，从而天然具备比传统数值仿真更高的精度。尽管 PNetLab 在精度方面表现出色，但其自身仍存在一些亟待改进的缺陷，本文提出了接口响应优化策略，有效地弥补了 PNetLab 在接口处理效率上的短板，通过集群扩展技术，合理配置和管理集群资源，大幅地提升了平台的仿真能力；此外，为了提高 ePNetLab 引擎仿真场景的构建效率与运行效率，引入了社区分割算法。本文从平台选型、接口优化、集群扩展以及算法引入等多个层次出发，系统地对 PNetLab 平台进行了深入研究与改进，为 IP 承载网仿真领域提供了一种更高效、精准的解决方案。

1 相关工作

数字孪生网络 (DTN, digital twin network) 是数字孪生技术在网络领域的应用，通过构建网络的虚拟副本，实现对网络状态的实时映射和优化。近年来，DTN 在提升网络运维效率、优化资源配置以及支持复杂网络功能方面展现出显著的应用价值。

1.1 数字孪生网络的发展现状

国际上，国际电联电信标准化部门发布了首篇 DTN 标准 Y.3090 《数字孪生网络——需求和架构》，给出了 DTN 的概念和需求^[6]。互联网工程任务组也开展了“数字孪生网络：概念和参考架构”的立项工作^[7]。美国电气电子工程师学会等国际组织在 DTN 的架构、建模和应用方面进行了深入研究。国际研究还关注了 DTN 在 IP 通信中的应用，提出了基于数字孪生的网络自治和生成式智能架构。实践方面，2024 年，英伟达联合诺基亚和美国东北大学推出了英伟达研究云平台，通过构建适用于通信的 DTN 研究平台，为高效人工智能 (AI, artificial intel-

ligence) 使能的通信网络提供仿真工具。2023年, 爱立信提出了无线接入网数字孪生2.0, 利用大数据计算技术创建能够实时反映无线接入网数据变化的数字孪生, 支持多网络、多层次和多种无线接入技术。

国内, 网络数字孪生技术的研究和应用已经取得了显著进展。中国移动在第三代合作伙伴计划项目(3GPP, Third Generation Partnership Project) 牵头首个网络数字孪生标准, 推动了DTN在国际标准中的应用^[8]。中国联通也发布了《中国联通数字孪生网络白皮书》, 讨论了孪生网络架构和典型应用场景^[9]。华为基于资源、网络、链路对象精细化建模, 构造了具备全域感知、预测及根因分析能力的数字孪生网络, 实现了跨时空网络全局故障感知与分析。中兴通讯提出了基于数字孪生的6G无线网络自治解决方案, 通过数字孪生技术优化无线网络的规划和运维。

同时, DTN的相关研究在学术界也成为热点。如, 利用图神经网络(GNN, graph neural network) 构建网络的数字孪生模型, 包括对网络节点、路由和调度规则的建模, 从而实现对网络时延、抖动和丢包的快速评估^[10-12]; 利用数字孪生技术进行网络生命周期的管理, 从而有效地降低管理成本、提高网络效率^[13]; 利用DTN产生数据从而训练强化学习模型^[14-15]; 以及数字孪生技术在网络切片、边缘计算等领域的应用^[16-19]。

在IP承载网领域, 数字孪生技术的应用实践主要集中在网络规划、优化和故障管理。例如, 通过数字孪生技术构建的虚拟网络模型, 能够实时反映物理网络的状态, 支持网络拓扑的动态调整和资源的灵活分配, 从而提升网络的运维效率和资源利用率。此外, 数字孪生技术还被用于模拟网络故障场景, 提前评估故障对业务的影响, 并优化故障恢复策略^[20]。

1.2 业界网络仿真引擎技术对比

仿真引擎是DTN的核心支撑工具, 通过高效模拟网络拓扑和运行状态可为DTN提供精准的虚拟模型和运行支持, 仿真引擎的功能与性能直接影响仿真的精度和效率。

网络仿真引擎主要存在两种技术路线: 数值化和模拟化。数值化是通过数学模型和计算机算法来模拟网络的行为, 核心在于利用软件工具创建虚拟的网络环境, 从而对网络协议、拓扑结构和流量行

为进行研究和分析^[21]。这种方法的优点是成本低、可控性强、能够灵活地调整参数并进行重复实验, 特别适合于理论研究和大规模网络的性能评估。然而, 由于完全依赖于模型和算法, 数值化在反映真实网络复杂性方面存在局限性, 且对大规模网络的仿真需要较高的计算资源。相比之下, 模拟化通过真实硬件或软件模拟实际网络的运行, 核心在于利用真实设备或虚拟化技术来构建接近实际运行环境的网络^[22]。模拟化的优点是精度高、实时性强, 能够更好地反映真实网络的行为, 特别适合于测试和验证网络设备的实际性能以及进行实时交互实验。然而, 模拟化的缺点是成本较高, 需要网元镜像和高性能硬件支持, 且在扩展性方面存在挑战, 难以模拟大规模网络环境。

针对上述两种技术路线, 本文分别筛选了业界主流仿真软件进行统一分析与比对。数值化方面, OPNET是一款商业化的网络仿真工具, 支持大规模网络的仿真, 能够模拟多种网络设备和协议, 图形化界面易于使用, 擅长网络规划和优化^[23]。OMNeT++是一个开源的离散事件仿真框架, 支持多种编程语言, 图形化界面基于Eclipse, 提供了丰富的工具和扩展模块^[24]。NS3是一款开源的网络仿真工具, 专注于网络协议和算法的研究, 提供了丰富的配套工具, 如动画演示工具和拓扑生成器工具^[25]。模拟化方面, eNSP是由华为提供的网络仿真平台, 专注于华为设备的模拟, 支持华为路由器和交换机的大部分特性, 图形化操作界面易于上手, 支持拓扑创建、修改和保存^[26]。HCL是由H3C公司开发的网络设备模拟软件, 支持H3C设备的模拟, 功能包括设备模拟、虚拟组网和数据传输模拟, 界面简洁, 操作方便^[27]。GNS3是一款开源的网络仿真工具, 支持多种真实设备和虚拟设备的模拟, 支持多种网络协议和设备类型^[28]。PNetLab是一个开源的网络仿真平台, 支持多种网络设备的模拟, 具有良好的可扩展性和灵活性^[29]。

数值类仿真作为业界研究热点, 相关文献已从开发语言、安装耗时、学习曲线、支持协议、优劣势等角度对仿真软件进行了分析对比^[30-31]。本文从IP承载网应用场景角度, 以精度、规模、性能、异构支持、易用性、扩展性和成本共7个方面对上述数值与模拟仿真软件进行横向对比。其中, 精度反映模拟真实网元运行机理的程度, 规模反映可仿真的

网元数量，性能反映构建仿真场景的效率，异构支持反映对多厂商多型号网元的支持能力，易用性反映仿真软件的操作复杂度，扩展性体现功能扩展、新型网元扩展、应用程序接口（API, application program interface）开放及与其他系统的集成能力，成本体现使用软件的必要花费。主流仿真软件对比见表1，其中，PNetLab在精度、异构、易用性、扩展性与成本方面优势突出，在规模和性能方面不足可进行针对性优化，相较于其他仿真软件，更适用于IP承载网复杂场景、高精度、大规模的仿真要求，因此，本文将其作为基础，构建高性能网络仿真引擎 ePNetLab。

表1 主流仿真软件对比

| 仿真软件 | 精度 | 规模 | 性能 | 异构支持 | 易用性 | 扩展性 | 成本 |
|---------|----|----|----|------|-----|-----|----|
| OPNET | 中 | 优 | 优 | 优 | 中 | 劣 | 劣 |
| OMNeT++ | 劣 | 优 | 中 | 劣 | 劣 | 中 | 优 |
| NS3 | 劣 | 优 | 中 | 劣 | 劣 | 中 | 优 |
| eNSP | 优 | 劣 | 劣 | 劣 | 优 | 劣 | 优 |
| HCL | 优 | 劣 | 劣 | 劣 | 优 | 劣 | 优 |
| GNS3 | 优 | 中 | 中 | 优 | 优 | 中 | 优 |
| PNetLab | 优 | 中 | 中 | 优 | 优 | 中 | 优 |

2 设计与实现

为构建一个灵活、高效且可靠的网络仿真引擎 ePNetLab，系统设计遵循模块化、高性能、高扩展原则。模块化方面，ePNetLab采用分层架构，服务网格层与虚拟化层分离业务逻辑与基础设施，支持

模块化扩展与替换，将系统拆分为功能独立、接口清晰的子模块，实现各模块的解耦与独立演进。高性能方面，ePNetLab通过社区分割算法动态划分网络拓扑，减少跨节点通信开销，同时优化底层平台，补充批量化处理能力，提升拓扑变更处理效率。高扩展方面，引擎具备弹性架构与开放接口，支持动态扩缩容，可根据仿真规模自动调整计算资源，API允许第三方系统无缝集成。

2.1 仿真引擎总体架构

依据上述设计原则，网络仿真引擎 ePNetLab 总体架构如图1所示，从底向上分为4个主要层次：基础设施层、虚拟化层、控制层和接入层，其中，控制层与接入层以微服务运行在服务网格环境中。

基础设施层是网络仿真引擎总体架构的根基，由硬件和操作系统两大核心组件构成。硬件部分包括物理服务器、网络设备、存储设备及计算资源，为上层虚拟化提供物理资源支撑。操作系统基于这些硬件部署，采用高稳定性、可扩展性的Linux发行版或专用服务器系统，负责管理硬件资源、提供基础服务接口以及支持虚拟化技术。该层为整个仿真引擎提供物理资源池化能力，使上层的虚拟化层能够通过资源抽象和动态分配高效运行。

虚拟化层即 PNetLab 仿真平台，是网络仿真引擎的核心功能载体，位于基础设施层之上，由运行环境、虚拟网元、虚拟交换机及虚拟链路等关键组件构成。核心内容是通过底层硬件资源的抽象化与池化，构建可动态编排的仿真资源池。具体而

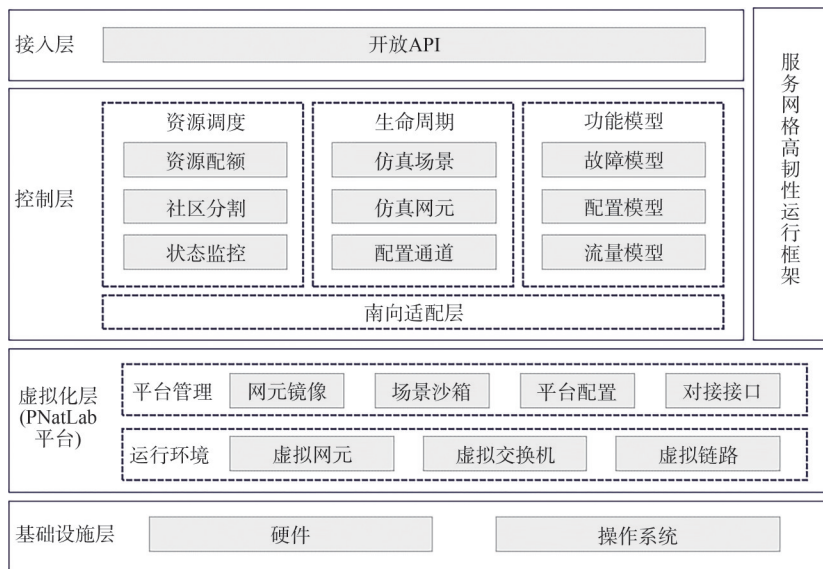


图1 网络仿真引擎 ePNetLab 总体架构

言，虚拟网元通过虚拟化技术将物理计算资源转化为可独立运行的仿真节点，模拟真实网络设备的功能；虚拟交换机基于软件定义网络技术实现逻辑转发规则的灵活配置；虚拟链路通过虚拟接口构建节点间的逻辑连接，形成可定制的拓扑结构。虚拟化层为上层控制层提供高度灵活、可扩展的仿真环境，一是通过资源隔离与动态分配，支持多租户场景中的并行仿真实验，避免资源冲突；二是借助虚拟化技术的快速部署能力，实现仿真场景的灵活构建与销毁，提升仿真效率；三是通过软件化定义网络元素，能够精准模拟单体网元功能与复杂网络行为，为功能验证、韧性测试等推演过程提供操作对象。

控制层是网络仿真引擎的中枢管理模块，位于接入层与虚拟化层之间，包括资源调度、生命周期管理、功能模型及南向适配层4个部分，控制层通过协调虚拟化层的资源与基础设施层的物理能力，实现仿真环境的全流程管控与仿真要素的全生命周期编排。具体方面，资源调度基于系统实时资源利用率及社区分割算法，对仿真环境中的资源进行合理分配和调度，确保仿真任务能够获得所需的计算、存储和网络资源；生命周期管理关注网元与场景的编排调度，从创建、配置到运行和销毁，支撑仿真环境按照预定的流程和条件执行，同时为虚拟网元提供管理通道，提供网元级别的监控与管理；功能模型提供了一系列模型来模拟网络行为和配置，包括故障模型、配置模型和流量模型。这些模型使仿真环境能够模拟各种网络状况，如设备故障、链路中断和不同的流量模式，以测试和验证网络的性能和韧性；南向适配层作为控制层与虚拟化层之间的桥梁，负责将控制层的指令和配置信息准确地传递和应用给虚拟化层，实现对仿真环境的精确控制。控制层通过这些功能模块，实现了对仿真

环境的全面管理和控制，为上层应用提供了一个稳定、灵活、可扩展的仿真引擎。

接入层位于网络仿真引擎架构的最顶层，提供开放的API，使外部应用和服务能够方便地接入和利用仿真引擎的功能，通过开放API的方式，允许用户自定义仿真任务、访问仿真数据、控制仿真流程等，实现与仿真环境的交互。

服务网络高韧性运行框架为仿真引擎提供了稳定性和可靠性保障，通过在控制层与接入层中实施微服务架构，利用服务网格技术增强服务间的通信、监控和管理，提供故障恢复、负载均衡、动态服务发现等关键功能，实现仿真控制面在面对各种网络故障和异常情况时仍能持续运行。

2.2 PNetLab 平台优化

PNetLab 本身作为社区维护的网络仿真工具，虽然功能丰富完备，但在特定性能与功能上仍存在欠缺，如拓扑管理类接口设计上均采用全量拓扑信息一次性返回机制，拓扑规模的增长带来接口性能劣化的问题，同时，社区版本不支持集群部署，仿真网元数量上受到制约，难以支撑IP承载网灵活多变的仿真规模。本文提出了一系列针对性的优化措施，旨在提升整体性能和扩展能力，满足高性能网络仿真引擎的要求。

1) 接口性能优化

针对仿真过程中涉及的关键API进行了改进，提高系统的整体响应速度和数据处理效率，PNetLab 接口优化清单见表2，优化措施核心在于减少API响应的数据量，减轻服务器负载并加快调用方响应。具体来说，对于添加、编辑、删除节点、网络和链路等操作，原生的接口设计会在操作完成后返回全量节点、网络和链路信息，这在数据量大时会导致响应时间延长，优化后，接口仅返回与操作

表2 PNetLab 接口优化清单

| 接口 | 功能 | 优化前 | 优化后 |
|---|---------|----------------|------------|
| /api/labs/session/nodes/add | 添加节点 | 返回全量节点、网络、链路信息 | 仅返回本节点信息 |
| /api/labs/session/nodes/edit | 编辑节点 | 返回全量节点、网络、链路信息 | 仅返回本节点信息 |
| /api/labs/session/nodes/delete | 删除节点 | 返回全量节点、网络、链路信息 | 不返回节点信息 |
| /api/labs/session/networks/add | 添加网络 | 返回全量节点、网络、链路信息 | 仅返回本网络信息 |
| /api/labs/session/networks/p2p | 添加链路 | 返回全量节点、网络、链路信息 | 仅返回本链路信息 |
| /api/labs/session/networks/delete | 删除网络/链路 | 返回全量节点、网络、链路信息 | 不返回网络/链路信息 |
| /api/labs/session/interfaces/setQuality | 链路质量 | 返回全量节点、网络、链路信息 | 仅返回本链路信息 |
| /api/labs/session/interfaces/setSuspend | 链路中断/恢复 | 返回全量节点、网络、链路信息 | 仅返回本链路信息 |
| /api/labs/session/multi_cfg_detail | 查看节点配置 | 返回全量节点配置信息 | 仅返回本节点配置信息 |

直接相关的单个节点、网络或链路信息，显著地减少了数据传输量。

2) 集群扩展优化

基于 PNetLab 平台使用的网络资源要素，结合跨节点通信基本原理，通过扩展开放网络编排能力，实现两个或多个 PNetLab 节点资源池化，由上层控制层进行统一管理调度。具体来说，PNetLab 节点中的网络资源包括虚拟网元接口及对应的网络隧道访问点 (TAP, tunneling traffic access point)、网桥和宿主机网卡等，在进行集群组网时，每个节点预留一个跨节点通信网卡，使用二层交换机将通信网卡互连，同时，各节点预留一个跨节点通信网桥，通信网卡默认绑定到网桥，在构建跨节点链路时，须将链路两端的虚拟网元接口对应的宿主机空间中的 TAP 口同时绑定到该网桥上，通过交换机中转，在两个网元间形成二层链路。

本集群扩展功能在实现上包括两方面，一方面由 PNetLab 平台开放网络资源原子化管理能力，另一方面由控制层对节点的网络资源进行调度编排，共同实现虚拟网元的跨节点组网功能。ePNetLab 集群网络原理如图 2 所示，以 3 台 PNetLab 节点集群为例，仿真引擎包括控制层节点、交换机和虚拟化节点 3 部分。交换机连接了控制面节点和各个虚拟化节点，实现数据的高效传输和交换，交换机共承载 3 类网络数据平面。一类是虚拟网元管理网络平面 (对应 eth0 网卡)，提供虚拟化节点中仿真网元的管理通道，由控制层节点通过 NETCONF、SNMP 等协议对网元进行配置、状态或性能等数据采集，为仿

真推演提供数据支撑。另一类是跨节点链路网络平面 (对应 eth1 网卡)，当一条链路上的两台网元节点分布在不同虚拟化节点上时，触发创建该网络平面，一条链路对应一个链路网络平面。如果链路两端节点属于同一虚拟化节点，则通过虚拟网桥来实现互联。最后一类是虚拟化节点管理网络平面 (对应 eth2 网卡)，控制层节点通过该平面与 PNetLab 节点通信，调用其北向 API，实现网元的生命周期管理。

2.3 微服务及接口设计

基于仿真引擎总体架构设计，针对部署于服务网络高韧性运行框架中的控制层，面向资源调度、生命周期管理、功能模型、虚拟化层适配几大功能，围绕服务的独立性、可扩展性，定义相关微服务与接口，支撑复杂的网络仿真和任务调度管理。

当前，Kubernetes^[32]结合 Istio^[33]是云原生应用下服务网络的最佳实践，可支撑构建高度可扩展、可观察、安全和易于管理的云原生网络应用，这种组合已成为许多云原生技术栈的核心组件。Kubernetes 是一个开源的容器编排引擎，支持自动化部署、扩展和管理容器化应用程序，通过强大的容器管理平台，帮助开发者轻松地构建、部署和管理应用程序。Istio 是一个开源的服务网络框架，用于连接、管理和保护微服务架构内的服务，支持与 Kubernetes 进行深度集成，借助 Istio 细致的流量控制功能，Kubernetes 可全面增强对微服务的治理能力。服务网络运行架构如图 3 所示，核心技术在于向每个服务实例中注入代理组件，这一代理会拦截并管理服务间的

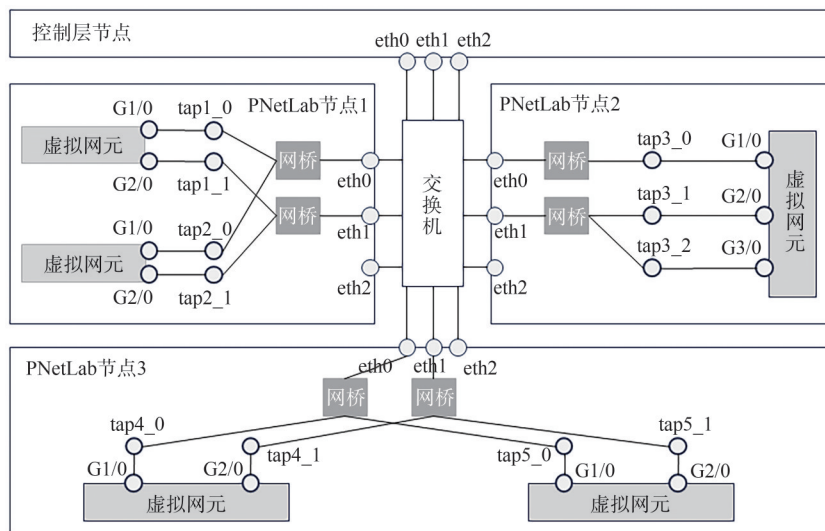


图 2 ePNetLab 集群网络原理

通信流量，进而实现流量路由、负载均衡、故障恢复、监控、日志记录和安全等功能。服务网格由控制平面和数据平面组成，其中，控制平面负责配置和管理数据平面的代理，协调不同服务之间的通信策略、路由规则和监控信息等；数据平面通过 Sidecar 代理将通信流量劫持到服务网格中进行处理，保障了服务间通信的可靠性和灵活性。

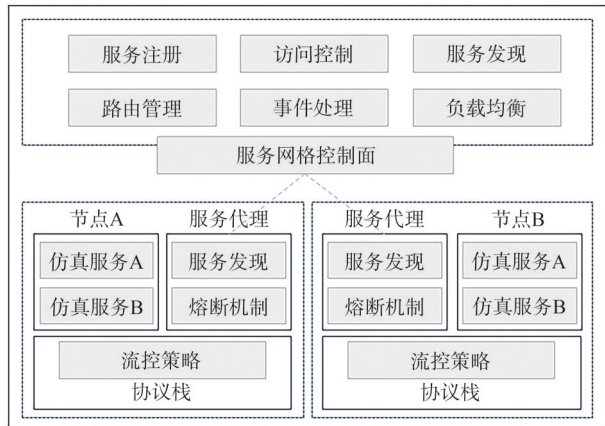


图3 服务网络运行架构

微服务方面，控制层拆分为7个微服务，ePNet-Lab 仿真引擎微服务及接口设计如图4所示，其中，适配微服务负责与PNetLab 节点交互，对接PNetLab 北向接口实现平台的配置、管理和监控；集群管理微服务负责 PNetLab 集群的管理，包括集群的配置、扩缩容等；资源调度微服务负责环境资源的配

给，基于社区分割算法在 PNetLab 节点间分割拓扑；集群监控微服务负责监控 PNetLab 节点的运行状态，包括节点健康状态、资源利用率等；网元配置微服务负责为虚拟网元提供配置管理通道，包括管理 IP 地址分配、通道如 NETCONF、SNMP 配置；场景管理微服务负责仿真场景的创建、执行、修改与删除；功能模型微服务为仿真过程提供功能模型，包括故障模型、配置模型和流量模型。基于流程设计，上述微服务通过遵循表现层状态转移 (REST, representational state transfer) 架构规范的 API 灵活地进行服务间调用，实现高效的相互协作和资源共享，同时，每个服务可单独部署和扩展，提升整个系统的稳定性与稳健性。

针对上述微服务中的资源管理服务，本文首次将格文-纽曼 (GN, Girvan-Newman) 算法引入大规模网络拓扑分割分配中，基于虚拟化的网络仿真场景中，虚拟网元之间通常存在频繁的通信需求，如果这些虚拟网元被随机分配到不同的物理节点，跨节点通信可能会导致显著的时延和带宽消耗。虚拟化集群环境中的虚拟网元节点合理分配是实现高效资源利用、降低通信开销和提升服务性能的有效手段。

GN 算法是一种经典的层次化社区分割算法，核心思想是通过逐步移除网络中具有高边介数的边，将网络分解为逐渐细化的社区结构。其数学理论支撑主要基于图论、最短路径统计和模块度优化^[34-35]。

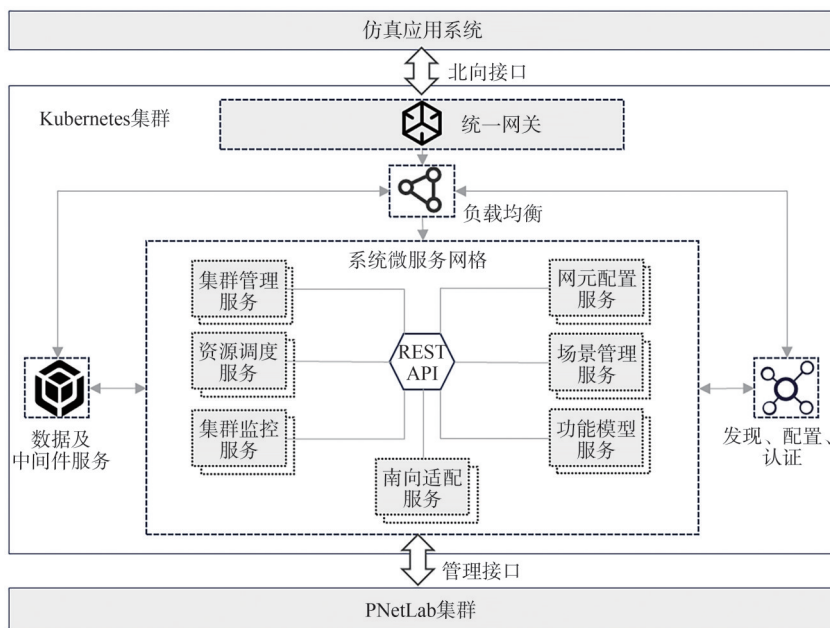


图4 ePNetLab 仿真引擎微服务及接口设计

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j) \quad (1)$$

模块度是衡量网络社区划分质量的核心指标，定义如式(1)，其中， A_{ij} 表示邻接矩阵元素，若节点*i*与*j*相连则为1，否则为0； k_i, k_j 表示节点*i*和*j*的度数（连接的边数）； m 为网络总边数，值为 $(\sum_i K_i)/2$ ， K_i 为节点*i*的度数； $\delta(c_i, c_j)$ 为指示函数， c_i 和 c_j 分别表示节点*i*和*j*的社区标识，若节点*i*和*j*属于同一社区则为1，否则为0。 Q 的取值范围为 $[-0.5, 1.0]$ ，值越大表示社区内连接越紧密、社区间连接越稀疏，GN算法在每轮移除边后，计算当前划分的模块度，选择 Q 最大的划分作为最优结果。

GN算法的数学步骤遵循一种层次化分解逻辑，流程可概括为3个阶段：初始化、迭代移除边和终止条件判定。首先，算法初始化时计算网络中所有边的边介数，量化每条边在网络最短路径中的关键性。其次，迭代阶段移除当前边介数最高的边，切断连接不同社区的“桥梁”，网络结构变化后重新计算剩余边的介数以更新全局状态，此过程循环执行，逐步将网络分裂为内聚的社区。最后，算法通过两类终止条件结束运行：一是极端情况下所有边均被移除，网络退化为孤立节点；二是根据模块度判定当前社区划分的质量，当模块度达到局部最大值时，认为此时划分的社区结构在“组内紧密、组间稀疏”的数学意义下达到最优，从而终止迭代。通过GN算法识别网络拓扑中具有高内聚性和低耦合性的节点子集，将拓扑内网元划分为多个逻辑组，并以逻辑组为单元集中分配虚拟化资源。

接口方面，控制层对外接口分为北向接口和管理接口，北向接口是面向仿真应用系统的接口，允许外部系统或用户与仿真引擎进行交互，用于接收来自用户或应用程序的请求，触发仿真引擎内部的相关操作。管理接口是面向PNetLab集群的适配接口，用于集群原子化功能的调用与控制，实现资源的优化配置、任务的调度执行以及集群状态的监控。

3 实验与分析

为验证ePNetLab仿真引擎在IP承载网数字孪生场景中的性能表现，本文围绕接口优化效率与资源分配算法性能开展实验验证。实验设计分为两部分：一是通过对比PNetLab优化前后的节点与链路

创建效率，量化大规模拓扑构建的性能提升效果；二是对不同节点规模的异构网络拓扑进行层次化社区分割实验，通过对比多种分配策略，验证GN算法在资源分配优化与通信开销控制中的有效性。实验从拓扑构建效率与算法性能突破双重维度验证了ePNetLab仿真引擎的技术优势，为IP承载网的智能资源调度与高可靠组网提供了理论与实验支撑。

实验环境包括4台硬件服务器，每节点128核、256 GB内存、2 TB资源配给，其中，3台部署优化后的PNetLab仿真软件，原生版本为4.2.10，1台部署仿真控制面微服务，K8S版本为1.19.9，4节点共同组成仿真引擎ePNetLab。

3.1 PNetLab接口性能优化验证

本实验分别从节点与链路创建耗时角度评估PNetLab平台接口优化情况。节点方面，依次创建1至100个网元，分别记录优化前后的单节点创建耗时情况，PNetLab平台优化前后仿真网元创建时间对比如图5所示。结果表明，优化前耗时呈显著线性增长趋势，单节点耗时从129 ms递增至1 270 ms，耗时增幅达884.5%，表明规模扩展导致性能严重劣化。优化后首节点因初始化流程优化耗时增加11.3%，但后续节点耗时稳定在183~229 ms区间，与规模无显著相关性。具体场景对比显示，10节点总耗时从234 ms降至198 ms，效率提升15.4%，50节点总耗时从678 ms优化至207 ms，效率提升69.5%，100节点总耗时从1 270 ms锐减至217 ms，效率提升达82.9%。

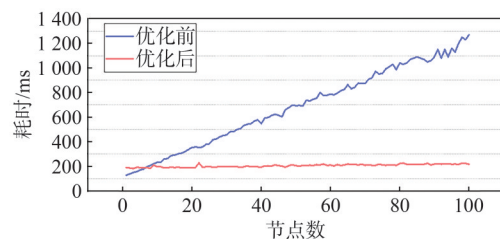


图5 PNetLab平台优化前后仿真网元创建时间对比

链路方面，在100节点规模网元的基础上，依次创建1至200条链路，分别记录优化前后单链路创建的耗时情况，PNetLab平台优化前后仿真链路创建时间对比如图6所示。结果表明，优化前链路创建耗时呈规模敏感特征，单链路耗时从1 890 ms波动增长至2 110 ms，耗时增幅11.6%，表明系统存在资源竞争导致的累积时延。优化后耗时整体下降62.3%~68.7%，首链路耗时降低68.7%，200链路时

耗时稳定在 843 ms，且规模相关性显著减弱。关键指标对比显示，50 链路场景中总耗时从优化前均值 1 887 ms 降至优化后均值 671 ms，降幅 64.5%，100 链路时从 1 988 ms 优化至 739 ms，降幅 62.8%，200 链路时从 2 110 ms 优化至 843 ms，降幅 60.0%。总体而言，接口优化使链路创建效率平均提升 2.8 倍。

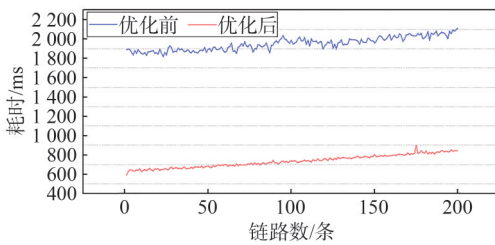


图6 PNetLab平台优化前后仿真链路创建时间对比

实验证明，PNetLab 优化为 ePNetLab 仿真引擎后有效地解除了规模与性能的正相关性，消除了原有系统随规模扩展的性能劣化问题，有效地支撑了大规模网络仿真拓扑的动态构建需求。

3.2 基于GN算法的社区分割性能验证

本实验验证了GN算法在优化大规模拓扑构建与降低通信开销中的有效性，通过对比GN算法与Louvain算法^[36]、随机分配策略的性能差异，实验聚焦3个核心目标，一是社区划分的质量，量化模块度与跨社区通信占比，评估算法对网络内聚性的提升效果；二是仿真效率与资源利用率，分析仿真构建时间及物理资源消耗，验证算法对系统负载的优化能力；三是业务影响，测试故障恢复时间，衡量算法对网络可靠性的实际增益。

Louvain算法与GN算法相似，是社区发现领域中另一个经典且具有代表性的算法，两者的设计理念有显著差异。GN算法是一种“自上而下”的算法，它从整个网络开始，逐步移除连接不同社区的

边，最终将网络分裂成一个个社区；Louvain算法是一种“自下而上”的算法，它从每个节点作为一个独立社区开始，逐步合并节点或社区，以最大化整个网络的模块度增益。两者都以模块度作为衡量社区划分质量的核心指标，通过迭代的方式逐步优化网络的社区结构，最终达到模块度最大化的目标。

实验操作针对50、150、500节点规模的异构网络拓扑，分别采用GN算法、Louvain算法及随机分配策略进行社区划分与资源映射。实验过程中，通过ePNetLab控制层日志实时采集模块度(Q值)、跨物理节点通信流量占比及仿真构建时间，模拟随机5%链路中断并记录开放式最短路径优先(OSPF, open shortest path first)协议收敛时间，同时监测CPU峰值利用率。所有实验重复5次，剔除异常值后计算均值，不同节点规模下拓扑划分策略效果对比见表3。

实验数据表明，GN算法在DTN仿真中展现出显著的性能优势，在50~500节点规模下，GN算法的模块度显著高于Louvain算法和随机分配策略，社区划分具备更高的内聚性；跨社区通信占比随规模扩大仅从50节点的25.4%增至500节点的40.2%，较随机分配降低48.9%，有效地缓解了物理节点间的带宽竞争。仿真效率方面，GN算法的构建时间虽略高于Louvain算法，但资源利用率表现更优，且耗时增速接近。业务性能方面，GN算法的故障恢复时间较Louvain算法与随机分配策略均有显著降低，表明其社区划分机制有效地抑制了故障传播时在跨节点链路上的开销。综上，GN算法通过层次化社区分割来优化资源调度，在仿真效率、资源利用率及业务性能上均具有较好的表现，为大规模网络拓扑的动态构建与实时运维提供了关键的技术支撑。

表3 不同节点规模下拓扑划分策略效果对比

| 节点规模 | 策略 | 模块度(Q) | 跨节点通信占比 | 构建时间/s | CPU峰值利用率 | 收敛时间/ms |
|------|---------|--------|---------|--------|----------|---------|
| 50 | GN | 0.72 | 25.4% | 124 | 64% | 105 |
| | Louvain | 0.69 | 27.1% | 112 | 69% | 123 |
| | 随机分配 | 0.35 | 46.8% | 155 | 75% | 178 |
| 150 | GN | 0.65 | 30.5% | 325 | 75% | 166 |
| | Louvain | 0.56 | 39.2% | 309 | 87% | 198 |
| | 随机分配 | 0.31 | 63.1% | 432 | 92% | 345 |
| 500 | GN | 0.51 | 40.2% | 743 | 90% | 343 |
| | Louvain | 0.42 | 49.3% | 725 | 96% | 398 |
| | 随机分配 | 0.18 | 75.8% | 872 | 98% | 584 |

4 结束语

本文针对网络数字孪生技术在 IP 承载网仿真试验中的应用,提出了一种高性能的仿真引擎 ePNetLab,该引擎基于 PNetLab 进行了深入的改进与优化。通过对 PNetLab 的性能优化与功能拓展,改进接口处理逻辑,提升了拓扑变更处理效率,基于虚拟化技术,设计并实现了集群化组网能力。在优化后的 PNetLab 基础上,设计了高性能仿真引擎 ePNetLab,引入了基于社区分割算法的拓扑动态构建方法,通过自动化网络分区与资源分配,缩短了大规模仿真环境的构建时间与应用开销。这些研究成果不仅为 IP 承载网的运维与决策提供了更可靠的技术支持,也验证了 ePNetLab 在高性能网络数字孪生仿真领域的应用价值,为后续的研究和应用奠定基础。

尽管仿真引擎的研究已经取得了一定进展,但仍存在许多挑战和改进空间。未来,期望通过引入更先进的算法和技术,进一步提升 ePNetLab 仿真引擎的性能和可靠性,包括利用人工智能技术优化资源调度和网络管理,以及通过机器学习和数据分析提高仿真精度,同时,将致力于增强引擎的安全性,确保仿真环境能够抵御潜在的网络攻击和威胁。通过这些努力,希望 ePNetLab 在网络规划、优化和管理方面的功能更加全面与实用。

参考文献:

- [1] Wu Y W, Zhang K, Zhang Y. Digital twin networks: a survey[J]. *IEEE Internet of Things Journal*, 2021, 8(18): 13789-13804.
- [2] Ali Madni M A, Iranmanesh S, Raad R. DTN and Non-DTN routing protocols for Inter-CubeSat communications: a comprehensive survey[J]. *Electronics*, 2020, 9(3): 482.
- [3] Jia W J, Wang W, Zhang Z Z. From simple digital twin to complex digital twin Part I: a novel modeling method for multi-scale and multi-scenario digital twin[J]. *Advanced Engineering Informatics*, 2022, 53: 101706.
- [4] Dai Y Y, Zhang K, Maharjan S, et al. Deep reinforcement learning for stochastic computation offloading in digital twin networks[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(7): 4968-4977.
- [5] Basile F, Carbone C, Chiacchio P. PNetLab: a tool for the simulation, analysis and control of discrete event systems based on petri nets[J]. *IFAC Proceedings Volumes*, 2004, 37(18): 213-218.
- [6] ITU-T. Digital twin network-Requirements and architecture: ITU-T Y.3090-2022[S]. Geneva: ITU, 2022.
- [7] IETF. Digital Twin Network: Concepts and Reference Architecture: draft-irtf-nmrg-network-digital-twin-arch-04[S/OL]. [Draft]. 2022-07-11 [2025-07-19].
- [8] 孙滔,周铖,段晓东,等.数字孪生网络(DTN):概念、架构及关键技术[J]. *自动化学报*, 2021, 47(3): 569-582.
Sun T, Zhou C, Duan X D, et al. Digital twin network (DTN): concepts, architecture, and key technologies[J]. *Acta Automatica Sinica*, 2021, 47(3): 569-582.
- [9] 中国移动研究院.数字孪生网络(DTN)白皮书[R/OL].北京:中国移动通信集团有限公司,2021 [2025-07-19].
CMRI (China Mobile Research Institute). White Paper on Digital Twin Network (DTN) [R/OL]. Beijing: China Mobile Communications Group Co., Ltd., 2021 [2025-07-19].
- [10] Ferriol-Galmés M, Suárez-Varela J, Paillissé J, et al. Building a digital twin for network optimization using graph neural networks[J]. *Computer Networks*, 2022, 217: 109329.
- [11] Ferriol-Galmés M, Cheng X L, Shi X, et al. FlowDT: a flow-aware digital twin for computer networks[C]//Proceedings of the ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE Press, 2022: 8907-8911.
- [12] Ferriol-Galmés M, Rusek K, Suárez-Varela J, et al. RouteNet-erlang: a graph neural network for network performance evaluation[C]//Proceedings of the IEEE INFOCOM 2022 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2022: 2018-2027.
- [13] Sanz Rodrigo M, Rivera D, Moreno J I, et al. Digital twins for 5G networks: a modeling and deployment methodology[J]. *IEEE Access*, 2023, 11: 38112-38126.
- [14] Vilà I, Sallent O, Pérez-Romero J. On the design of a network digital twin for the radio access network in 5G and beyond[J]. *Sensors*, 2023, 23(3): 1197.
- [15] Zhang Z F, Liu Y C, Peng Z Y, et al. Digital twin-assisted data-driven optimization for reliable edge caching in wireless networks[J]. *IEEE Journal on Selected Areas in Communications*, 2024, 42(11): 3306-3320.
- [16] Wang K, Du H T, Su L. Digital twin network based network slice security provision[C]//Proceedings of the 2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI). Piscataway: IEEE Press, 2023: 1-6.
- [17] Wang H Z, Wu Y L, Min G Y, et al. A graph neural network-based digital twin for network slicing management[J]. *IEEE Transactions on Industrial Informatics*, 2022, 18(2): 1367-1376.
- [18] Hao Y X, Wang J X, Huo D K, et al. Digital twin-assisted URLLC-enabled task offloading in mobile edge network via robust combinatorial optimization[J]. *IEEE Journal on Selected Areas in Communications*, 2023, 41(10): 3022-3033.
- [19] Aliyu I, Arigi A, Oh S, et al. Towards a partial computation offloading in in-networking computing-assisted MEC: a digital twin approach[C]//Proceedings of the NOMS 2024-2024 IEEE Network Operations and Management Symposium. Piscataway: IEEE Press, 2024: 1-9.

- [20] Cui Y, Liu J C, Yu M L, et al. Network digital twin[J]. IEEE Network, 2024, 38(1): 5-6.
- [21] Hui L B, Wang M W, Zhang L, et al. Digital twin for networking: a data-driven performance modeling perspective[J]. IEEE Network, 2023, 37(3): 202-209.
- [22] Villa D, Tehrani-Moayyed M, Robinson C P, et al. Colosseum as a digital twin: bridging real-world experimentation and wireless network emulation[J]. IEEE Transactions on Mobile Computing, 2024, 23(10): 9150-9166.
- [23] Chen M, Miao Y M, Humar I. Introduction to OPNET network simulation[M]//OPNET IoT Simulation. Singapore: Springer Singapore, 2019: 77-153.
- [24] Xian X D, Shi W R, Huang H. Comparison of OMNET++ and other simulator for WSN simulation[C]//Proceedings of the 2008 3rd IEEE Conference on Industrial Electronics and Applications. Piscataway: IEEE Press, 2008: 1439-1443.
- [25] Campanile L, Gribaudo M, Iacono M, et al. Computer network simulation with ns-3: a systematic literature review[J]. Electronics, 2020, 9(2): 272.
- [26] Chen J, Zheng X, Zhang L, et al. WLAN simulation experiment based on ENSP[C]//Proceedings of the Journal of Physics: Conference Series. IOP Publishing, 2019, 1325(1): 012046.
- [27] 邱鹏, 霍瑛, 蒋悦. 基于H3C Cloud Lab的企业网络设计与仿真[J]. 实验室研究与探索, 2020, 39(10): 118-125.
Qiu P, Huo Y, Jiang Y. Design and simulation of enterprise networks based on H3C cloud lab[J]. Research and Exploration in Laboratory, 2020, 39(10): 118-125.
- [28] Korniyenko B, Galata L, Ladieva L. Research of information protection system of corporate network based on GNS3[C]//Proceedings of the 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). Piscataway: IEEE Press, 2020: 244-248.
- [29] Ajiardiawan G, Suryanto Y. Performance analysis of segment routing on MPLS L3VPN using PNETLAB[C]//Proceedings of the 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST). Piscataway: IEEE Press, 2021: 185-190.
- [30] Arvind T. A comparative study of various network simulation tools[J]. International Journal of Computer Science & Engineering Technology (IJCSET), 2016, 7(8): 374-378.
- [31] Siraj S, Gupta A, Badgajar R. Network simulation tools survey [J]. International Journal of Advanced Research in Computer and Communication Engineering, 2012, 1(4): 199-206.
- [32] Carrión C. Kubernetes scheduling: taxonomy, ongoing issues and challenges[J]. ACM Computing Surveys, 2022, 55(7): 1-37.

- [33] Nutalapati P. Service mesh in Kubernetes: implementing istio for enhanced observability and security[J]. Journal of Scientific and Engineering Research, 2021, 8(11): 200-206.
- [34] Girvan M, Newman M E J. Community structure in social and biological networks[J]. PNAS, 2002, 99(12): 7821-7826.
- [35] Newman M E J, Girvan M. Finding and evaluating community structure in networks[J]. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics, 2004, 69(2): 026113.
- [36] Ghosh S, Halappanavar M, Tumeo A, et al. Distributed louvain algorithm for graph community detection[C]//Proceedings of the 2018 IEEE international Parallel and Distributed Processing Symposium (IPDPS). IEEE Press, 2018: 885-895..

[作者简介]



石鸿伟(1982-), 男, 东南大学网络空间安全学院博士生、紫金山实验室高级工程师, 主要研究方向为未来网络体系架构、软件定义网络、网络智能控制。



倪中阳(1987-), 男, 博士, 紫金山实验室研究员, 主要研究方向为网络管控、数字孪生。



陆干沂(1991-), 男, 博士, 紫金山实验室研究员, 主要研究方向为网络编排、数字孪生。



黄韬(1980-), 男, 博士, 北京邮电大学网络与交换技术全国重点实验室教授、博士生导师, 主要研究方向为路由与交换、软件定义网络、内容分发网络、确定性网络、算力网络。