

网络系统实验平台: 发展现状及展望

杨林瑶^{1,2,3} 韩双双^{1,2,4} 王晓^{1,2,4} 李玉珂^{1,2,3} 王飞跃^{1,2,5}

摘要 本文首先介绍了网络仿真技术的概念、特点和发展现状,在此基础上介绍了当前主流的通用型网络仿真软件的主要特点并对它们进行了对比分析和评价. 为了解决现有网络架构灵活性和服务能力差的问题,我们引入了面向大数据建模、能够对网络整体状态进行预测和实时优化的平行网络架构及其计算实验平台. 该平台融合虚拟的人工网络 and 实际网络的数据并对其进行实验分析和评估,可以发现最优的控制方案. 基于平行网络计算实验方法,本文设计了针对微信朋友圈转发预测的网络优化方法,并通过实验验证了该方法可以有效提升网络的性能.

关键词 仿真软件, 网络仿真, 平行网络, 计算实验, 虚实交互

引用格式 杨林瑶, 韩双双, 王晓, 李玉珂, 王飞跃. 网络系统实验平台: 发展现状及展望. 自动化学报, 2019, 45(9): 1637–1654

DOI 10.16383/j.aas.c180728



开放科学(资源服务)标识码(OSID):

Computational Experiment Platforms for Networks: The State of the Art and Prospect

YANG Lin-Yao^{1,2,3} HAN Shuang-Shuang^{1,2,4} WANG Xiao^{1,2,4} LI Yu-Ke^{1,2,3} WANG Fei-Yue^{1,2,5}

Abstract In this paper, the current network experiment platforms are investigated, introduced and compared. The most popularly used simulator and its main advantages and disadvantages are demonstrated. To solve the problem of lacking flexibility and service capability of traditional network experiment platforms, based on the parallel network architecture, the corresponding computational experiment platform is proposed. The proposed platform is data-oriented, and by using the computational experiments and analysis, an optimized control strategy can be continuously updated and tracked, thus the self optimization of network systems is achieved. In the end, a computational experiment based on retweet analysis of Wechat Moments is proposed, and the effectiveness of this method is evaluated.

Key words Simulators, network simulation, parallel network, computational experiments, virtual-reality interaction

Citation Yang Lin-Yao, Han Shuang-Shuang, Wang Xiao, Li Yu-Ke, Wang Fei-Yue. Computational experiment platforms for networks: the state of the art and prospect. *Acta Automatica Sinica*, 2019, 45(9): 1637–1654

收稿日期 2018-11-05 录用日期 2019-02-03
Manuscript received November 5, 2018; accepted February 3, 2019

国家自然科学基金(61501461, 61702519, 61533019, 91720000), 北京市科委项目(Z181100008918007), 英特尔智能网联汽车大学合作研究中心(ICRI-IACV)资助

Supported by National Natural Science Foundation of China (61501461, 61702519, 61533019, 91720000), Beijing Municipal Science Technology Commission (Z181100008918007) and Intel Collaborative Research Institute on Intelligent and Automated Connected Vehicles (ICRI-IACV)

本文责任编辑 王卓
Recommended by Associate Editor WANG Zhuo

1. 中国科学院自动化研究所复杂系统管理与控制国家重点实验室 北京 100190 2. 青岛智能产业技术研究院 青岛 266000 3. 中国科学院大学 北京 100049 4. 青岛慧拓机器人有限公司 青岛 266000 5. 国防科技大学军事计算实验与平行系统技术研究中心 长沙 410073

1. The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190 2. Qingdao Academy of Intelligent Industries, Qingdao 266000 3. University of Chinese Academy of Sciences, Beijing 100049 4. Vehicle Intelligence Pioneers Inc., Qingdao 266000 5. Research Center of Military Computational Experiments and Parallel Systems Technology, National University of Defense Technology, Changsha 410073

网络技术自诞生以来经历了飞速的发展,当前,计算机网络和通信网络已经融入人们生活的方方面面. 近年来,随着网络商业化应用的进步,新的网络结构不断涌现,新的协议和算法被不断提出和应用. 然而,目前的网络也面临着许多问题,比如缺乏高效的资源管理能力、服务定制能力薄弱、难以提供有效的用户管理^[1-2]等,现有的网络体系结构越来越难以满足未来的发展需要. 在这种情况下,如何高效、低成本地设计有效的新型网络结构、开发新的网络协议和算法,以解决目前网络结构不灵活、网络开发部署成本高等问题,已经成为一个焦点问题^[3].

网络仿真通过软件计算网络中的实体如节点、路由器、交换机、链路等之间的交互来建模网络中的行为,将网络中的行为如收发数据等价为随机发生的离散事件进行排队处理^[4-7],是一种由统计模型驱动的网络规划、设计和分析技术,能够为网络研究提供客观、可靠的依据. 因此,在当前网络规模和结构复杂性急剧上升的情况下,网络仿真技术可以

灵活、精准地设计、模拟网络的结构,以较低的成本进行网络研究,并准确评估算法的性能和效果^[8-11],极大地方便了研究人员进行研究和设计.当前,国内在网络仿真软件的设计和研究成果不多,但国外已经有很多成功的应用软件^[12-39].其中,有适用于多种网络类型的通用仿真软件^[12-35]如开源的 NS-2、NS-3、OMNeT++、J-SIM 等及商业仿真软件 OPNET、QualNet 等,还有一些针对特定网络设计的专用网络仿真软件^[36-39],它们极大地方便了网络研究并得到了广泛的应用.

然而,网络仿真技术还存在准确度过分依赖于模型精度和参数设置,不能自适应地调整模型和参数,紧密耦合的仿真层次设计限制了高层应用的灵活性等缺陷.为了克服传统的网络结构设计的缺陷,提高网络的灵活性,降低网络设计和更新成本,McKeown 教授提出了软件定义网络 (Software-defined network, SDN) 技术^[40].SDN 采用转发与控制分离、网络全局信息对控制可见的设计,解耦了网络的软硬件资源,并将其作为服务开放出去,极大地提高了网络的灵活性和开放性.但是,SDN 仍不能很好地进行动态优化和控制.为了满足网络自适应优化,提高服务质量的需要,未来的网络架构应当进化为基于 ACP (Artificial societies, computational experiments, and parallel execution) 方法 (即人工社会、计算实验和平行执行)^[41-43],面向网络大数据的平行网络架构^[44].

平行网络基于真实网络的数据建模人工网络,通过构建灵活的网络智能体和规则构建大量的人工网络场景^[45-48],生成完备的人工网络场景数据.大规模的人工网络数据结合真实网络的数据,通过各种计算实验^[49-52],给出对真实网络优化运行的指导意见.同时,将优化方案在人工网络和真实网络中平行执行,使网络模型训练和评估在线化、长期化^[53-55],能够持续优化网络系统,提高复杂网络的运行效果.平行网络系统为从整体上描述网络的状态提供了基本思路,并为开展网络的优化运行实验提供了可行的方法.其综合型和主动性的建模方式使计算实验可以全面考虑各种因素对系统的影响,实现对系统整体的模拟^[56-61],而传统的仿真方式只能实现局部行为的模拟.

作为平行网络技术研究的重要手段,仿真技术的研究和应用十分关键.网络仿真软件数量众多、特点不同^[7],为了使读者深入了解网络仿真技术及其主要平台的原理和主要特点,推动下一代网络分析和计算实验技术的研究,本文主要做了以下工作:

1) 系统地分析了主流的通用型网络仿真平台的发展现状、特点和优缺点及网络仿真技术的缺陷^[62-66]和发展趋势,对各网络仿真软件的主要性

能进行了对比分析和评价;

2) 介绍了平行网络的架构并着重介绍了计算实验的理论、方法、应用^[43-53]和平台架构等相关问题,设计了一个计算实验验证该方法的合理性和可行性,以期对未来网络研究和网络仿真技术研究提供有益的启发和借鉴.

本文的组织结构为:第 1 节概述网络仿真平台的发展现状和研究情况;第 2 节对主要的通用型网络仿真软件进行介绍,并对其优势和主要的缺陷进行了分析;第 3 节分析当前网络仿真技术面临的主要问题和挑战;第 4 节分析网络仿真的发展趋势和展望;第 5 节介绍平行网络的计算实验方法的相关理论和实验案例;第 6 节总结本文的主要内容并对平行网络的发展前景进行了展望.

1 网络系统实验平台概述

计算机网络仿真软件兴起于 20 世纪 80 年代,它的出现很好地弥合了网络设计及其应用之间的鸿沟.网络仿真具有如下的特点:1) 简单易用,可以很方便地进行网络协议的验证与设计^[4];2) 应用广泛,可以用于大规模网络的设计分析,这是其他技术难以比拟的;3) 成本较低,设计和修改比较容易,使用起来灵活方便;4) 具有较高的精度,能够很好地刻画网络的特性,具有良好的统计复用性.

网络仿真对网络中随机发生的离散事件如丢包、延迟等的模拟,因此,网络仿真软件的内核大多是离散事件驱动的.仿真软件的原理框图如图 1 所示.网络仿真软件内部一般都配置有一个离散事件调度器,按照队列组织离散事件.在进行仿真时,每次从队首取出一个离散事件进行处理并将其从队列中删除,并且根据离散事件的结果修改全局变量也就是网络中的状态变量并产生过程统计数据,然后接着取下一个事件,直至队列为空即为仿真结束^[5].仿真时钟用于记录仿真的进程,随处理的离散事件的增加而推进.为了模拟网络中的随机性如数据包在网络中的逗留时间和丢失率等,仿真系统需要引入能够产生随机分布的随机数发生器.随机数发生器一般是用一定的算法实现的伪随机数发生器,本质上不具备真正的随机性,并且在算法遇到重复值时,随机数会发生周期性重复,此时便不能有效模拟网络的随机性.因此,仿真软件模拟的随机事件数量不能超过伪随机数发生器的循环周期,伪随机数发生器的设计至关重要^[6].

网络仿真软件的一般使用流程^[7]为:1) 明确研究对象仿真需求,如仿真的对象、层次、规模等.2) 根据需求创建仿真模型,部署节点、交换机、路由器等仿真实体.3) 创建仿真场景,设计网络的拓扑结构和设定特殊场景参数如移动场景下的速度等.

4) 设定其他网络参数, 如带宽、数据率等. 5) 启动仿真进程, 采集数据. 6) 运用数据分析工具对采集到的结果统计数据数据进行可视化分析.

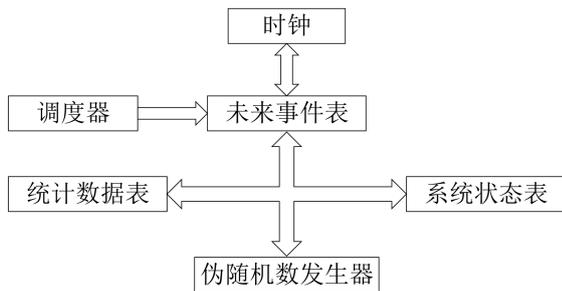


图 1 仿真软件原理框图
Fig. 1 Schematic diagram of simulators

网络仿真技术主要应用在网络性能评估和网络吞吐量规划、预测新协议的性能和进行新协议的开发及快速进行网络部署设计等领域^[8]。经过三十多年的发展, 网络仿真领域已经产生了数量庞大, 特点各异、用途不一的仿真软件簇. 当前的网络仿真软件大多基于分组级仿真设计, 都可以在单机上运行, 但是每一种仿真软件都有或多或少独特的侧重方向, 都有自己的优缺点和适用的领域, 例如物理层的仿真通常选择具有更高物理层精度的 NS-2^[9], 大量节点的网络仿真需要选择具有分布式处理能力的平台, 如果需要较少的成本最好选择开源的软件等, 因此用户在选择仿真工具时应当面向具体应用, 根据仿真的需求来选择合适的仿真软件.

评价仿真软件优劣的主要指标有: 1) 内存效率, 即仿真软件仿真一定数量的节点需要的计算机内存^[10], 所需内存越少, 内存效率越高; 2) 仿真能力, 即软件在满足精度要求的条件下所能够仿真的最大节点数量; 3) 速度, 即仿真软件仿真一定节点数量时需要的运行时间, 特别是对于大型网络的仿真时间; 4) 灵活性, 即仿真模型设计和修改的方便程度; 5) 可扩展性, 即对仿真软件进行扩展或修改应用到新的领域所需要的工作量, 越容易添加新模块和功能的仿真软件, 其可扩展性越高; 6) 易用性, 即软件的用户界面和仿真设计、分析辅助工具和其他用户支持等^[11].

网络仿真平台按照其适用领域可以分为通用仿真平台^[12-35]和专用仿真平台^[36-39], 通用的仿真平台可以用于绝大多数类型网络及网络的不同层次, 主流的通用网络仿真软件主要有 NS-2、NS-3、GlomoSim (Global mobile information system simulator) 等, 而专用的仿真平台是专为某一类型的网络如无线传感网 (wireless sensor networks, WSN)^[36]、点对点 (Peer-to-Peer, P2P) 网络^[37]、车载自组织网络^[39] (Vehicular ad-hoc networks,

VANET) 等设计的, 只有有限的使用范围. 表 1 列出了通用型网络仿真平台和一些专用于特定类型网络的仿真软件.

数量庞大的网络仿真软件为研究人员和工程师挑选合适的工具带来了困难, 也催生了很多针对仿真软件对比的研究工作, 这些工作大多按照一定的标准对不同的平台进行对比, 或者针对特定的项目或场景进行性能比较. 如文献 [9] 对 NS-2、J-SIM、GlomoSim、OMNeT++、QualNet 和 OPNET 的特点进行了概述, 文献 [12] 对 J-SIM、OMNeT++、NS-2、OPNET 的特性进行了比较, 文献 [21] 对 GlomoSim、NS-2、OMNeT++、J-SIM 进行了全面的分析, 文献 [66] 则对 J-SIM、OMNeT++、NS-2 就其安装简易度、说明文档、可视化和数据分析能力进行了比较. 还有的工作通过实验对不同平台针对特定问题的仿真能力进行了比较, 如文献 [67] 就仿真准确度对 NS-2、OPNET 进行了对比, 文献 [68] 则进一步比较了 OPNET、NS-2、GlomoSim 的准确度. 不同论文研究的软件对象不尽相同, 侧重点也不完全一样. 为了给网络研究人员选择恰当的网络仿真软件提供可以依据的参考, 本文对主流的通用型网络仿真平台进行了对比分析和评价, 全面涉及使用最广泛的仿真软件, 同时, 针对目前通信网络数据面和控制面分离的趋势, 对两种 SDN 常用的仿真软件进行了功能介绍与分析.

表 1 常用仿真软件分类

Table 1 Common simulators classification

| 通用型 | 无线传感网 | P2P 网络 | 车载自组网 |
|------------|---------------|------------|---------------|
| NS-2 | SensorSim | QueryCycle | GrooveNet |
| NS-3 | Castalia | Narses | TraNS |
| GlomoSim | VisualSense | 3LS | NCTUns |
| OMNeT++ | Viptos | NeuroGrid | MOVE |
| OPNET | Sidh | Peersim | VANET MobiSim |
| JiST/SWANS | Prowler | P2PSim | Veins |
| GTNetS | NsrIsensorsim | SimP2 | - |
| SSFNet | - | - | - |

2 基础实验平台介绍

本节主要对 NS-2、NS-3、GlomoSim 等主流的通用型网络仿真软件及 Mininet 和 EstiNet 这两种重要的 SDN 仿真软件的背景、框架、仿真设计等进行了介绍, 并对其优缺点进行了分析, 最后小结部分对它们进行了比较分析.

2.1 NS-2

NS (Network Simulator) 起源于 1989 年美国军方的 Real Network Simulator 项目, 是最早的网

络仿真器之一, NS-2 是 NS 比较成熟的版本. NS-2 是一种面向对象的网络仿真器, 本质上是一个离散事件模拟器, 是目前学术界使用最广泛的网络仿真软件. NS-2 的软件架构如图 2 所示, 它基于 C++ 和 Otcl (Object tcl) 两种编程语言开发. 其中, Otcl 可以快速进行修改, 用于构成解释器, 适于编写仿真模型和进行仿真参数配置; 而 C++ 程序具有很高的执行效率, 主要构成编译器负责执行仿真进程, 用户可以基于 C++ 编写自己的库或协议, 并利用 Make 模块添加到 NS-2 的库文件中. 解释器和编译器之间通过 TclCL (Tcl with classes) 接口进行交互, TclCL 可以将 Otcl 脚本解释为 C++ 代码执行, 并基于模型配置脚本生成对应的编译层变量^[12]. NS-2 内封装了许多功能模块, 包括事件调度器、队列、节点、链路、传输控制协议/因特网互联协议 (Transmission Control Protocol/Internet Protocol, TCP/IP) 等, 用于表示仿真中的网络元素并控制仿真进程. NS-2 也是离散事件驱动的仿真器, 事件调度器负责激活仿真队列中的事件, 并通过虚拟的仿真时钟记录仿真时间. NS-2 选择了网络动画演示工具 (Network animator, NAM) 和 Xgraph 作为数据可视化工具, 用户可以在网络配置脚本中添加 Trace 信息设置语句, 仿真时就会生成相应的基于文本的 Trace 文件, 用户可以使用 Xgraph 将数据曲线画出, 也可以用 NAM 将仿真过程以动画展示出来.

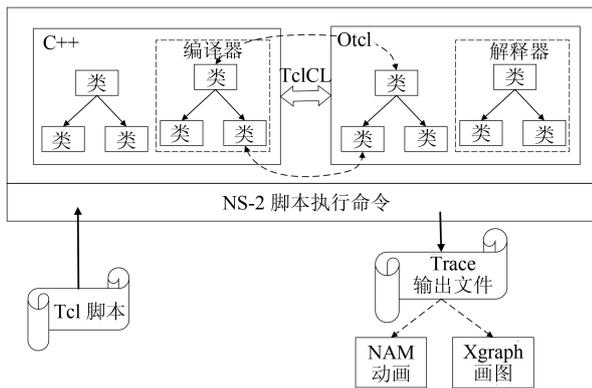


图 2 NS-2 双语言软件结构

Fig. 2 Two language structure of NS-2

NS-2 支持包括 TCP、路由、广播协议在内的广泛的协议仿真, 目前已经形成了十分完善的协议库, 对正在使用的大多数协议都可以直接调用库进行仿真. 它还支持有线、无线、局域网、卫星网络等多种网络类型的仿真, 其仿真结果具有较高的可信度. NS-2 支持简化的开放式系统互联 (Open System interconnect, OSI) 网络分层模型, 省略掉了会话层和表示层^[13]. NS-2 可以与真实的网络进行交互, 它可以与真实网络进行连接并捕捉网络中的数据包,

同时也可以向外界的网络发送数据. NS-2 相比于其他软件具有较好的移动性支持, 可以更精准地建模移动网络模型, 对网络底层仿真具有较高精度.

但是, 由于 NS-2 的两种编程语言的复杂架构, 创建一个新的模型或协议是比较困难的, 导致它的可扩展性和灵活性比较差, 同时, 它本身没有 GUI 支持, 使用比较繁琐. 另外, NS-2 对一些硬件设备的支持不尽如人意, 例如对于天线的支持不够, 只支持双向天线和无向天线. 但是, 得益于其活跃的用户社区, NS-2 拥有很全面的说明文档, 很方便的应用程序编程接口 (Application programming interface, API), 及大量的扩展工具可以供用户添加使用.

2.2 NS-3

为了简化 NS-2 的框架, 降低复杂度, NS-3 研究社区的开发者在 NS-2 基础上开发了 NS-3. NS-3 始于 2006 年, 它与 NS-2 最大的不同也是优势在于其核心代码和功能模块完全基于 C++ 开发, 并且大部分模块提供可选的 Python 扩展接口, 极大地降低了用户的编程复杂度. NS-3 架构由一系列基于 C++ 类开发的功能模块组合而成, 各模块可以实现网络的不同功能如移动模型、路由协议等, 模块之间同时存在一定的调用和依赖关系. NS-3 的基本模块及其组织结构如图 3 所示, 其中, Core 模块是 NS-3 的核心模块, 包含事件调度、随机变量、回调 (Callback, NS-3 中简化函数调用的机制)、日志 (Logging, 用于输出内部模块执行过程和调试信息) 等核心功能^[14], Network、Mobility 等模块用于实现网络数据分组、节点移动模型等相应的网络功能, Helper 包含模块的 Helper 类代码, Test 则包含模块的测试代码, Waf 是 NS-3 的代码编译工具, 所有的仿真代码都由它编译运行. NS-3 的仿真模型具有清晰的逻辑和较高的可读性, 模型的组成及其顺序一般包括: 选择相应的模块、配置 Logging 和 Trace、设定命令行参数、创建节点、安装网络设备、安装协议栈、设定移动性模型、安装应用层协议、启动仿真等.

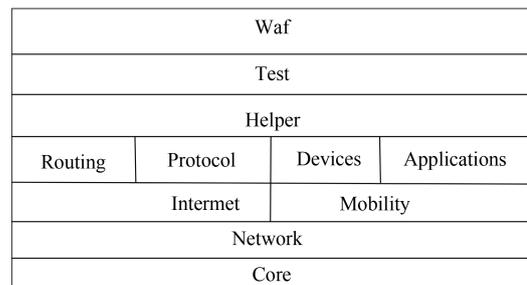


图 3 NS-3 模块体系结构

Fig. 3 Module architecture of NS-3

NS-3 仿真层次类似于 OSI 网络体系结构, 它将数据按照设备层、网络层、传输层、API 套接字和应用层的顺序封装, 由类模拟数据按照 TCP/IP 协议传输的过程, 具有良好的灵活性和可扩展性. NS-3 同样是事件驱动的仿真器, 事件由节点触发, 仿真内核中的事件处理器按照对应的仿真时间从队列中取出相应的事件进行处理, 数据包的转发由指针的交互完成, 仿真进程占用内存资源较少且运行速度较快. NS-3 的数据传输过程与物理网络的真实模型类似, 因而其仿真结果具有较高的可信度. NS-3 的结果数据包括网络场景数据和统计数据, 场景数据可以通过离线的 NetAnim 或在线的 PyViz 可视化工具展示^[10], 从而使用户直观观测到仿真过程, 而统计数据可以利用 NS-3 的 Tracing 系统动态跟踪仿真生成的 Pcap 统计文件, 并可利用 Wireshark 等工具分析输出结果. NS-3 还支持分布式处理和并行仿真, 仿真速度最高可以提升三十倍.

NS-3 目前并不完善, 是正在发展中的仿真器, 它只有少量的现有协议的模型和库, 实现像 NS-2 那样完善的协议库还需要较长的时间和社区用户的共同努力. 但是 NS-3 发展正强劲, NS-2 已经停止更新, 而 NS-3 还在不断维护更新. 作为 NS-2 的升级, NS-3 还将在功能上进行扩展, 例如提供面向仿真过程透明的统一对外编程接口等.

表 2 列出了一些 NS-2 与 NS-3 之间的比较.

表 2 NS-2 和 NS-3 的比较
Table 2 Comparison of NS-2 and NS-3

| NS-2 | NS-3 |
|-------------------|-------------|
| 开源 | 开源 |
| 用户自定义模型困难 | 用户自定义模型容易 |
| 大量的用户贡献工具 | 少量的用户贡献工具 |
| 双语言编写 (C++, Otel) | 单语言编写 (C++) |
| 编程能力要求高 | 编程能力要求适中 |
| 现有模型支持大量的协议 | 现有模型支持少量的协议 |

2.3 GlomoSim

GlomoSim 是加利福尼亚大学洛杉矶分校并行计算实验室开发的一个基于库的并行仿真软件, 主要用于仿真无线网络. GlomoSim 采用模块化的设计思想, 很好地支持了 OSI 协议分层. 它基于 Parsec 语言 (一种基于 C 的仿真语言, 用于实现离散事件仿真模型的顺序和并行执行)^[15] 编写了一系列库, 每一个库专门用于仿真某一层的某个协议^[16], 每个库有一个 API 与其他库进行交互, 用于调用其他层的协议, 实现多协议联合仿真, 新的协议需要使用 Parsec 编写并添加到库里. 这种设计提高了不同层的模型和协议的独立性, 用户可以很方便地

修改和替换某一层的库^[17], 有效提升了它的可扩展性. GlomoSim 可以运行在共享内存或分布式处理器上, 通过一些同步协议进行协作, 可以实现并行仿真和分布式处理, 因此能够被用于大型网络的仿真. GlomoSim 引入了网格的概念, 在网格中, 一个简单的实体可以仿真系统中的若干节点, 当网络中增加新的节点时, 仿真的实体数目可以保持不变, 仿真前所需规划的仅仅是该系统应该采用多少个实体来表示. 因此, GlomoSim 可以支持上百万节点. 同时, 它可以提供节点和层次聚类, 这是很多仿真器无法实现的^[18]. 但是, GlomoSim 缺少 TCP 应用, 并且不支持图形用户界面 (Graphical user interface, GUI) 界面.

GlomoSim 在 2000 年发行 2.0 版本后, 便不再进行更新, 现在推出的是商业化的版本 QualNet. QualNet 是用 C++ 编写的应用于无线、有线及混合动态网络的快速、精确的开发、仿真系统, 主要是对移动通信网络仿真进行了优化处理, 在仿真速度上得到了很大的提升, 同时通过对无线信道和射频技术的强化建模也保证了较高的仿真精度. QualNet 基于普林斯顿共享内存计算机应用程序库 (The Princeton application repository for shared-memory computers, PARSEC) 并行仿真内核, 每个节点都独立进行运算, 这也和现实相符合, 允许用户在真正的并行仿真环境当中优化仿真性能. 因此, 仿真速度快是 QualNet 的突出优势, 对小规模同等复杂度的网络模型仿真, QualNet 仿真速度是其他仿真器的几倍, 对于大规模网络, QualNet 仿真速度是其他仿真器的十几倍到几十倍, 如果使用并行仿真模式, 能达到比其他仿真器快千倍的速度.

2.4 OMNeT++

OMNeT++ 是一个免费的多协议离散事件仿真软件, 它不是专为网络系统设计的, 原则上可以用于任何相互传递信息的组件系统的仿真. 该软件已经有很多网络方面的扩展应用, 例如通信网络之间的流量建模、协议仿真、排队网络和复杂网络等^[19]. OMNeT++ 的基本组件是模块, 其结构是一种嵌套层次结构, 模块可以是原子的也可以是其他子模块组成的, 模块之间可以进行通信. OMNeT++ 基于面向对象的设计思想, 采用 C++ 作为编程语言, 新的组件和模块的编写主要通过调用 C++ 类库中的仿真内核、随机数发生器、数据采集器等实现, 因而具有较高的灵活性和可扩展性.

尽管 OMNeT++ 灵活的嵌套模块化结构难以理解, 它的使用却是相对容易的. OMNeT++ 在配置网络模型时不一定要使用 C++ 编程, 它支持更易于掌握的网络描述语言 (Network description, NED), 只有最低层次的模块需要用 C++ 编写^[19].

同时, OMNeT++ 有良好的 GUI 支持, 可以很方便地在图形化界面下调试参数、修改模型和观察结果, 而且, 它还拥有十分全面的用户手册、API 说明文档以及一些教程, 用户可以得到良好的支持. OMNeT++ 通过信息传递接口 (Message passing interface, MPI) 和并行虚拟机 (Parallel virtual machine version 3, PVM3) 实现了并行仿真功能, 在设计时就考虑了扩展性, 因此很适合开发和修改, 它的程序兼容性很好, 其他软件编写的仿真程序经过少许修改就可以应用到 OMNeT++ 中, 它的程序也可以移植到任意一台没有安装 OMNeT++ 仿真环境的主机上运行. 但是 OMNeT++ 的协议没有很好地按照 OSI 模型组织, 同时对移动性建模的支持也比较差.

2.5 OPNET

OPNET 最早由麻省理工学院于 1986 年推出, 现在已经发展成为一个广泛使用的商业离散事件仿真器, 但是研究人员可以获得一个免费的开放版使用. OPNET 是基于 C++ 编写的, 其商业应用最为广泛^[20], 近几年被第三方权威机构 (如 Network World 等) 评选为“世界级网络仿真软件”第一名^[21]. OPNET 支持大量的无线网络技术, 如 IEEE 802. 11、IEEE 802.16、3G、超宽带、蓝牙和卫星网络等, 它有很多有线和无线网络协议模型, 用户可以直接从这些预定义的模型中选择协议构建自己的模型, 易于使用. OPNET 仿真内核采用层次化的模拟方式, 其协议模型按照业务层、TCP 层、IP 层、IP 封装层、地址解析协议 (Address resolution protocol, ARP) 层、介质访问控制 (Media access control, MAC) 层、物理层的层次模型设计, 符合 OSI 分层模型的设计^[22]; 其网络仿真模型与网络、设备、协议的实际网络层次结构对应, 分为进程层、节点层、网络层三层. 最底层为进程模型, 用状态机来描述协议, 并可使用 Proto-C 语言编码节点模型中处理机及队列的行为. 进程层模型的行为由状态转移图刻画, 进程按照状态转移图的转移条件在阻塞和活动两个状态之间循环, 并根据执行代码执行相应的动作. 第二层为节点模型, 可以反映设备的特性, 由相应的协议模型构成, 每个节点模块用于实现真实网络节点设备的一个或几个功能, 多个模块组合在一起实现节点的完整功能. 最上层为网络模型, 包含子网、节点和通信链路等, 用于封装网络设备、通信链路和业务配置等^[23].

OPNET 的仿真模型都可以看作由多个相互连通的子系统组合成的分布式系统, 子系统之间由不同的消息传递机制进行交互, 包括基于包的通信、应用接口控制信息通信和基于通信链路的通信等^[23]. 其中, 基于包的信息传递机制可以模拟实际网络中

包的流动及包的封装和拆解, 适用于同一节点内部不同模块之间的通信; 接口控制信息是用户自定义的与事件相关的数据列表, 可以用到各种有关事件调度的场景; 基于通信链路的消息传递用一系列管道去模拟物理链路中不同节点的通信, 包括点对点链路、总线链路和无线链路.

OPNET 将统计结果分为矢量统计量和标量统计量, 以便于对仿真结果进行分析处理. 其中, 矢量统计量分为本地统计量和全局统计量, 分别针对单个模块和整个网络模型的统计信息进行收集, 并向用户提供四种不同的采集模式, 即 All value (收集所有值)、Sample (采样收集)、Bucket (取平均)、Glitch removal (使平滑), 增强了多角度分析结果的能力^[24]. 标量统计量一般由用户手动设置, 可以针对某个参数设置不同的数值水平, 同时执行多个场景仿真, 从而根据标量统计量同时进行多个参数水平的测试, 极大地提高了研究效率.

OPNET 既支持多个程序并行执行, 又支持分布式仿真, 同时它的系统在环机制可以与真实网络环境中的软硬件进行交互. 作为一个商业软件, OPNET 提供了全面的用户手册, 并提供良好的图形编程界面, 模型创建、仿真执行、结果分析等都可以在 GUI 下很方便地进行, 这些都极大地降低了开发的难度. 相比于 NS-2、GlomoSim 等仿真软件, OPNET 支持对特殊的硬件设备如收发器和天线等的仿真. OPNET 的伪随机数发生器的循环长度比较短, 对大规模网络的仿真能力较弱.

2.6 J-SIM

J-SIM 是俄亥俄州立大学的一个研究小组用 Java 开发的通用仿真平台, 它的结构是一种灵活的基于组件松耦合的软件架构. 在 J-SIM 中, 节点、连接、协议等都是组件, 大的组件可以由几个内部组件组成. 组件之间可以通过它们的端口相互通信, 并且支持一对一、一对多和多对多的通信模式. 组件的行为由一个契约定义, 契约规定了组件应该对它的每个端口接收到数据做出什么反应^[25], 组件在运行时可以自发处理数据, 因此, 可以单独对组件进行设计、运行、测试, 而不需要过分关注整体系统, 十分易于进行扩展. J-SIM 库的设计基于 Java 类, 并且基于图论的思想, 仿真模型直接用图来表示^[26], 不同的节点之间以单端方式连接. J-SIM 运行时由 Java 虚拟机调度线程执行, 因此, J-SIM 可以跨操作系统平台运行. J-SIM 仿真是一种查询驱动的仿真, 当用户提出一个仿真查询, 系统会在保存以前模型结果的数据库中查找信息, 如果查到所需的结果, 就直接返回给用户, 否则系统就实例化这个模型, 执行它然后将结果返回给用户并将结果保存到数据库中.

J-SIM 支持网络分层, 并且可以仿真任意层数

的网络, 它的自治组件结构使得它可以很方便地实现不同层的节点之间的连接, 因此还可以进行交叉层的仿真. J-SIM 提供图形化的用户界面, 用户直接画图拖放节点来定义网络模型, 大幅减少代码量. 模型还可以很好地重利用, 只需要将建好的模型当作子图添加即可, 也便于理解. J-SIM 还提供接口可以与其他脚本语言进行整合, 新发布的版本还提供了一个 Tcl 的 Java 命令语言 (Java command language, Jacl) 解释器^[26], 因此, 一些其他语言编写的工具如 NAM 也可以整合到 J-SIM 中.

J-SIM 提供一些小型网络的示例和说明, 但是缺少大型仿真的指导文档, 用户进行复杂网络开发难以得到良好的支持. 另外, J-SIM 和 OPNET 一样, 其随机数发生器是 Java 自带的, 随机数的循环长度比较小, 因此对于大规模网络的仿真能力较差.

2.7 JiST/SWANS

JiST (Java in simulation time) 是一种运行在 Java 虚拟机上的网络仿真平台, 它是新一代离散事件仿真器的原型, 即基于虚拟机的仿真^[27]. SWANS (Scalable Wireless ad Hoc Network Simulator) 是一个在 JiST 平台之上搭建的无线网络仿真器, 它由独立的软件组件构成, 可以通过组件结合搭建出无线网络模型, 具有较高的灵活性. JiST 的框架由四部分组成: 编译器、虚拟机、字节代码重写器和基于语言的仿真时间内核. JiST 的仿真执行过程为: 首先对模型代码进行编译, 然后在加载时动态重写代码, 最后由仿真时间内核生成仿真时间执行语义, 控制虚拟机执行仿真可执行代码^[28].

JiST/SWANS 最大的特点就是非常高效, 相比于其他类型的仿真平台无论在速度还是内存消耗上都有优势, 表 3 给出了一些节点数下 JiST/SWANS 与 GlomoSim 和 NS-2 的对比.

表 3 JiST/SWANS 与 GlomoSim 和 NS-2 的性能对比
Table 3 Comparison of JiST/SWANS, GlomoSim and NS-2

| 节点数 | 仿真器 | 耗时 | 内存消耗 |
|-----------|----------|---------|-----------|
| 500 | SWANS | 43 s | 1 101 KB |
| | GlomoSim | 82 s | 5 759 KB |
| | NS-2 | 7136 s | 5 8761 KB |
| 5 000 | SWANS | 430 s | 5 284 KB |
| | GlomoSim | 6191 s | 27 570 KB |
| 50 000 | SWANS | 4 377 s | 49 262 KB |
| 1 000 000 | SWANS | - | 933 MB |

JiST 还具有较高的灵活性, 其仿真程序不需要用专门的仿真语言编写, 也不需要调用各种仿真函数, 而是可以用 Java 直接定义, 它将 Java 虚拟机直接转换为仿真平台, 可以在普通的 Java 编译

器上编译, 因此具有良好的跨平台性和可移植性. JiST/SWANS 支持移动性模型, 并且协议按照 OSI 标准组织, 对分层协议的支持较好.

2.8 GTNetS

GTNetS (Georgia Tech Network Simulator) 是美国乔治亚工学院基于 C++ 编写的面向对象的分布式高效网络仿真平台, 它具有良好的可扩展性, 主要面向大型网络设计, 支持超过 100 000 个节点. 它通过降低待处理的离散事件规模、优化内存管理、减小日志文件的大小这三方面的优化有效地提高了大量节点时的仿真效率. 基于分布式和并行处理设计, GTNetS 可以有效地提高数据的处理能力和仿真速度. GTNetS 的每一个协议栈都属于一个独立的 C++ 类, 并且协议分层是按照现实中的协议分层设计的^[29]. GTNetS 支持图形化显示, 并且支持对节点和链路的禁用和启用.

GTNetS 是一个全特性的网络模拟环境, 其设计理念是创造一个完全近似于实际网络的模拟环境, 使状态仿真的结果能够更准确地反映真实网络运行状态并且提高其代码的可移植性, 使程序可以很方便地移植到真实的网络设备中去. GTNetS 严格按照网络协议分层设计, 其不同层的协议栈之间有明确的界限. GTNetS 中的数据包包含一类数据单元协议数据单元 (Protocol data unit objects, PDUs) 用以定义数据包的传输协议, 这些数据单元会随着数据包在协议栈中的上下传输被添加或删除. GTNetS 中的节点设计与实际网络中的终端节点如电脑、网卡等类似, 一个节点可以有多个网络接口, 每个接口又可以绑定一个 IP 地址或与一个子网相连接^[30]. GTNetS 的传输层实体也模拟真实网络场景构建, 其采用类似 TCP 的方式与端口绑定, 通过端口与其他传输层实体进行交互.

2.9 SSFNet

SSFNet (Scalable simulation framework network models) 是一个用于互联网协议和网络结构仿真的基于 Java SSF (Simple seam framework) 的组件集合, 主要用于仿真 IP 层以上的网络模型, 具有良好的可扩展性, 物理层和链路层可以在单独的组件中进行仿真. SSFNet 模型是自我配置的, 也就是每一个 SSFNet 类实例可以通过查询配置数据库自动进行配置. 网络配置文件采用领域建模语言 (Domain modeling language, DML) 格式, 用来配置一个完整的网络模型, 并且在一个可扩展的 DML 数据库包的帮助下实例化一个仿真^[31]. 用于构建虚拟互联网模型的主类分成两个衍生的框架: SSF.OS (包括 ProtocolGraph、ProtocolSession、ProtocolMessage and PacketEvent 三个主类) 用

于建模主机和系统组件, SSF.NET (包括 Net、Host and Router、NIC 和 Link 四个主类) 用于建模网络的连通性、创建节点和链接配置. SSFNet 仿真的运行过程为: 首先 SSFNet 中的 Net.Net 对象查询 DML 库, 将模型加载到运行配置数据库对象 Configuration 中, 然后 Net.Net 对象将实例化所有实体如节点、协议等, 最后调用初始化方法进行初始化, 仿真程序便开始运行, 并输出仿真的过程参数.

SSFNet 在设计时格外重视内存占用, 因此资源消耗比较小, 同时, 具有有效的线程调度程序和内存到内存的消息传递机制, 可以实现并行化处理, 因此支持较多的节点数量. 但是, SSFNet 缺少对用户的扩展工具支持, 也无法很方便地对结果进行分析^[32], 这些对用户的使用造成了障碍.

2.10 Mininet

Mininet 是 2010 年 Stanford 大学基于 Linux 操作系统开发的虚拟网络仿真平台, 是目前主流的 SDN 仿真平台. Mininet 可以对基于 OpenFlow、OpenVswitch 模型的各种协议进行开发验证, 且代码可以很好地迁移到硬件设备中^[33]. Mininet 利用 Linux 内核提供的轻量级虚拟机制如进程和命名空间来创建网络模型并实现网络节点之间的数据转发和交互, 它可以通过命令行实现复杂的网络拓扑结构的构建和修改, 且提供 Python API 定制各种工具, 支持多人协同研发.

Mininet 具有良好的可扩展性, 它的轻量级结构使得它在一台主机上就可以轻松模拟具有上千个节点的网络, 并且可以验证、测试一个包含主机、链路、交换机等的完整网络体系. Mininet 还具有很好的可移植性, 由于它完全基于 Linux 内核开发, 它的程序可以移植到其他 Linux 主机中, 其他 Linux 程序也可以在 Mininet 环境下运行. 但是, Mininet 的中央处理器 (Central processing unit, CPU) 周期由虚拟主机、虚拟交换机和控制器共享, CPU 调度器不能准确地控制调度顺序, 因此, Mininet 的仿真结果准确度不够高且难以重现^[34].

2.11 EstiNet

EstiNet 是台湾 SDN 解决方案提供商思锐科技 (EstiNet) 研发的商用 OpenFlow 网络仿真器, 能够模拟数千个 OpenFlow 1.0 和 1.3 交换机. 它具有良好的用户界面并提供可视化工具, 能够对 OpenFlow 控制器发送的数据包进行分析, 并且在网络故障时可以快速地进行分析. EstiNet 利用一种称为内核重进入 (Kernel re-entering)^[35] 的方法使用隧道网络接口来捕捉从 IP 层向下发送的数据包并将它们送入仿真内核. 在仿真内核中, 每一个虚拟主机都有自己的协议栈, 协议栈包含 MAC 层、物理层和其

他 IP 层之下的协议层, 这些协议层可以仿真时延、带宽等, 然后, 将包发回给目的主机的物理层.

EstiNet 的仿真过程如图 4 所示: 首先虚拟主机 1 的应用程序发送的数据包依次经过各层协议被隧道 1 抓取, 隧道 1 按照先进先出的顺序将数据送入仿真内核, 仿真内核通过仿真协议栈各层依次对数据进行分析, 然后仿真内核再依次经过各层对数据进行处理, 最后再通过仿真内核的接口将数据依次通过物理层到传输层传送到目的主机 2. 所有 Linux 程序不经修改即可作为应用程序加入到 EstiNet 的仿真中, 并且使用 Linux 内核构建的真实 TCP/IP 协议栈来传送数据, 因此 EstiNet 具有良好的扩展性和准确性. 相比于 Mininet, EstiNet 的仿真结果可以复现且更加精确, 同时, 在大规模网络仿真上 EstiNet 的性能更加优越.

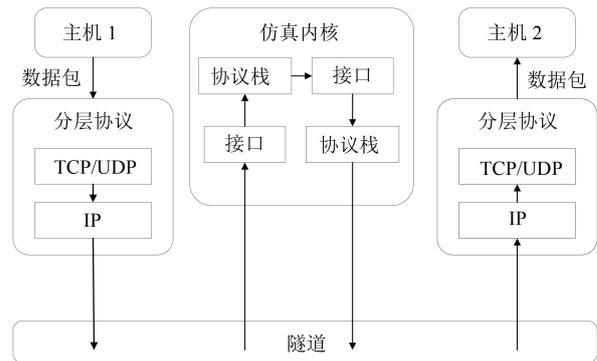


图 4 EstiNet 仿真过程

Fig. 4 EstiNet simulation process

2.12 总结、分析与评价

网络仿真是网络科学研究和通信网络协议开发必不可少的重要手段, 网络仿真平台具有数量庞大、适用范围不一的特点, 本章对比较常用的通用网络仿真软件进行了介绍.

其中, NS-2 是应用最广泛的仿真软件, 尽管它的双语言架构难以掌握, 但是它拥有完善的协议库, 因此广受青睐. NS-3 是 NS-2 的升级, 由双语言架构简化为单语言架构, 更加灵活且易于掌握, 但是目前协议库尚不完善. GlomoSim 的最大特点是通过分布式计算和并行处理技术实现了大规模网络的仿真运算, 能够支持百万级的节点数量, 其缺点在于缺少 TCP 支持且缺乏良好的用户界面. QualNet 是基于 GlomoSim 推出的商业化的仿真软件, 其最大的优势在于运算迅速, 但是它不开源, 成本高. OM-Net++ 也是一个很活跃的仿真平台, 被学术界广泛使用, 它的嵌套层次结构可以提供非常灵活的仿真支持, 易于修改和开发, 但是缺少对层次协议和移动性模型的支持. OPNET 是最常用的商业网络仿真软件, 拥有十分完善的图形界面和协议库, 但是它

的可扩展性较差, 难以进行二次开发. J-SIM 是一款基于 Java 开发的仿真平台, 它基于图形化思想设计, 拥有很好的 GUI, 允许画图设计网络, 大大简化了代码设计, 但是它的说明文档不充分, 使用范围不如上述的平台广. JiST/SWANS 是一个基于 Java 虚拟机的仿真平台, 它可以直接用 Java 编写仿真模型, 并且在 Java 虚拟机上运行, 因此比较灵活方便. GTNetS 的设计理念是创建一个完全近似于真实网络的模拟环境, 代码可以在真实网络和虚拟环境之间很方便地移植. SSFNet 也是基于 Java 开发的, 它的主要优点是内存消耗比较小并且节点可以自动配置, 但是它的功能较弱, 只能仿真 IP 层以上协议.

SDN 是未来重要的发展方向之一, 经典的网络仿真平台一般对 SDN 的支持不够强大, 例如 NS-3 只能支持 0.89 版本的 OpenFlow 协议, 而最新的 OpenFlow 协议已经升级到 1.5 以上了, 所以针对 SDN 网络一般使用专门的 SDN 仿真平台, 其中最具有代表性的是 Mininet 和 EstiNet. Mininet 是一个开源的 SDN 网络仿真平台, Mininet 运行在 Linux 内核上, 采用基于流程的思想设计, 可以用 Python 编写仿真程序, 是目前使用最广泛, 最活跃的 SDN 仿真软件, 但是 Mininet 的仿真结果不能复现且精确度差一些. EstiNet 是商业化的 SDN 软件, 兼具仿真和调试的功能, 基于内核重进入的设计思想, 使用隧道对数据包进行分析仿真, 因此精度较高且结果易复现.

大量研究对不同网络仿真软件的性能进行了实验对比, 基于文献 [66–70] 的实验结果并结合作者对各平台的实验验证, 表 4 对文中所介绍的仿真软件的特性进行了分析评价 (使用 * 表示对某一评价指标的优劣, * 越多表示其对应指标越优).

3 现存问题与挑战

网络仿真技术作为当前网络研究和设计的主流

技术, 取得了较大的发展和进步, 多种多样的仿真平台为网络结构和协议设计和分析提供了有力的工具. 但是, 网络复杂度和用户需求的进一步提升对网络仿真技术提出了更高的要求, 网络仿真技术还面临着许多问题和挑战.

首先, 随着网络规模和复杂度的不断提升, 大规模网络计算和处理技术需要进一步应用到网络仿真中^[62–63]. 目前提高网络仿真软件速度的方法主要有: 1) 并行和分布式处理技术, 并行计算指的是同时处理多条指令的技术, 这主要通过共享内存的方式实现, GlomoSim 已经有所实现, 主要用于有线网络的模拟上, 而分布式处理技术是将任务分配到多个处理器上同时进行处理, 可以有效地提高软件的可扩展性; 2) 分段仿真技术^[71], 它通过消除冗余计算的方式提高计算效率、节约计算资源; 3) 混合模拟技术, 它通过将解析模型和离散模型相结合的方式加速仿真器的运行^[64]. 但是多数仿真软件尚未有效利用这些技术, 同时, 近年来兴起的图形处理器 (Graphics processing unit, GPU) 加速计算^[72] 等技术被证明可以有效提升计算能力, 而网络仿真对这些技术的应用比较迟滞.

其次, 智能分析和优化方法在网络仿真软件中的应用不足. 网络仿真软件的数据分析可以分为两种: 一种被称为在线序列分析, 是指在仿真的同时计算误差, 只有当误差充分小的时候仿真停止, 这种方法可以获得较为精确的仿真结果; 另一种方法称为离线序列分析, 为了获得误差较小的结果, 离散序列分析可能要用统计独立的伪随机数重复仿真过程, 直到获得可靠的结果. 当前, 网络仿真软件只能对仿真数据进行简单的处理和展示, 对模型的分析 and 优化主要依靠人工完成, 而缺乏更有效的分析方法. 少有网络仿真软件能够提供完善的机器学习等分析工具, 无法对网络数据进行深度的分析从而为用户优

表 4 仿真软件性能对比

Table 4 Performance comparison of simulators

| 软件名称 | 编程语言 | 平台 | 开源 | 内存消耗 | 速度 | 灵活性 | 可扩展性 | 易用性 |
|------------|-----------|-------|-----|------|-----|-----|------|-----|
| NS-2 | C++, Otcl | 跨平台 | 开源 | * | * | * | ** | * |
| NS-3 | C++ | 跨平台 | 开源 | *** | *** | ** | *** | *** |
| GlomoSim | Parsec | Linux | 开源 | ** | ** | * | *** | ** |
| QualNet | Parsec | 跨平台 | 不开源 | ** | ** | * | * | *** |
| OMNeT++ | C++ | 跨平台 | 开源 | * | ** | ** | *** | ** |
| OPNET | C++ | 跨平台 | 不开源 | ** | * | ** | * | *** |
| J-SIM | Java | 跨平台 | 开源 | ** | ** | *** | *** | ** |
| JiST/SWANS | Java | 跨平台 | 开源 | *** | *** | *** | ** | ** |
| GTNetS | C++ | Linux | 开源 | — | — | ** | ** | * |
| SSFNet | Java | 跨平台 | 开源 | — | — | * | ** | ** |
| Mininet | Python | Linux | 开源 | ** | ** | *** | *** | ** |
| EstiNet | — | Linux | 不开源 | *** | *** | ** | * | ** |

化网络设计提供更有有效的支持。

再次, 仿真软件与真实设备的交互不足, 既难以利用实际网络的数据支持更逼真的仿真模型设计, 又无法加速仿真算法应用到实际网络设备中^[65]。当前, 多数网络仿真软件没有提供可与实际网络设备进行交互的接口, 仿真模型无法同时在网络设备上验证, 也无法接受实际网络的反馈。同时, 将仿真的模型和算法应用到实际网络上也比较困难。

现有的仿真平台已越来越难以满足规模日益增长的网路的需要, 同时也无法对现实网络进行在线的评估和优化, 无法高效地帮助提升实际网络的性能。为此, 必须对网络仿真技术进行改进升级, 不断提升其模拟大规模网络系统的能力。

4 发展趋势与展望

随着技术不断发展, 网络仿真软件也在不断改进, 并呈现出许多新的发展趋势。

1) 随着并行计算和网格计算等技术的兴起和发展, 越来越多的仿真软件开发了分布式处理和并行仿真机制, 能够利用更多的分布式计算资源加强运算能力, 不断提升仿真速度, 例如, 文献 [73–76] 等都基于并行和分布式计算技术对 NS-2 进行了提升, 实现了对大规模网络的快速仿真, 后期推出的仿真软件如 NS-3 等也都应用了分布式和网格计算等技术强化运算能力。2) 模块化的架构设计被不断强化, 且基于 OSI 的层次化设计被更多网络仿真软件采用以提高网络模型设计的逻辑性和可读性。新开发的网络仿真器的模型基本按照简化的 OSI 协议分层组织, 仿真过程中的数据包以类似的层次结果传输和处理, 使网络仿真与真实的物理网络过程更相似。3) 专用的仿真语言逐渐被摒弃, C++、Java、Python 等通用编程语言被完全用于软件的开发和模型设计, 极大地降低了用户的学习和使用难度。这一方面得益于计算能力的提升, 仿真效率不再依赖于特定的编程语言; 另一方面, 庞大的开源库使得通用编程语言也可以轻易实现网络系统的建模。4) 用户支持不断加强, 当前仍在更新的仿真软件大多提供全面的用户文档, 同时, GUI 和结果分析及展示工具越来越完善, 支持越来越多的开源工具和算法辅助仿真数据的分析。5) 对网络优化分析的支持逐步加强, 与机器学习等算法的结合越来越广泛, 开始提供机器学习等优化分析的扩展工具, 使网络设计更加高效, 例如, NS-3 甚至发布了 ns3-gym-App^[77] 以提供强化学习工具。

为有效提升网络实验系统的性能, 克服传统仿真无法自适应优化网络等缺陷, 我们在以上发展趋势的基础上引入了基于平行网络的计算实验平台, 从而为网络设计和优化技术的发展提供新的思路。

5 平行网络计算实验平台

传统的网络架构面临着灵活性不足、开发应用慢、运维要求高等一系列问题, SDN 技术将控制层和数据层分离, 解除了它们之间的紧密耦合, 数据层仅负责转发和接收数据, 控制层掌握全局的控制信息, 提供面向用户的可编程接口, 提高了网络的可编程性、灵活性和开放性。但是, SDN 只能基于当前网络信息和既定策略控制网络, 无法充分评估不同网络策略的效果, 同时, 它也无法针对网络攻击等稀疏场景预先给出最优策略。为此, 未来的网络将是基于人工网络大数据建模, 能够全面优化网络的平行网络系统。

5.1 平行网络概念与架构

平行网络基于 ACP 方法提出, 并融合了 SDN 的思想, 所谓 ACP, 是人工网络、计算实验、平行执行三者的有机结合^[1]。人工网络融合了 SDN 转发与控制相分离的思想^[41], 将底层资源封装起来, 提供开放的可编程接口并可基于多智能体方法^[42]对复杂网络系统进行建模, 进一步提升了网络的灵活性; 计算实验可以基于一定的策略和规则生成完备的人工网络场景^[43], 进而基于大量且完备的人工网络数据和实际网络数据计算和优化网络, 产生对各种网络场景的最优控制策略; 平行执行通过人工网络与实际网络之间的动态交互引导实际网络逼近更优的人工网络, 从而达到对网络的集中控制、整体优化^[44]。平行网络不仅使网络更加开放, 而且可以充分利用完备的人工网络大数据, 对各种可能的网络场景下的参数和控制策略预先进行学习、训练和评估^[45], 从而能够为网络提供更全面的解决方案, 提升了网络的自适应优化能力。平行网络及其计算解析架构如图 5 所示, 总体而言, 平行网络的 ACP 由三部曲组成。

1) 人工网络是与实际网络相对应的虚拟的网络, 它融合了 SDN 转发与控制分离、控制逻辑集中及网络能力开放的思想, 将网络底层资源抽象为虚拟的、逻辑独立的网络容器, 并向控制平面提供开放的可编程接口, 形成一个与底层物理网络无关的软件可控的系统。借助知识表示、知识工程与多智能体建模等方法, 人工网络可以构建大量可编程、可重构、可计算的软件定义的对象、软件定义的流程、软件定义的场景, 进而将它们共同组合成复杂网络的“计算实验室”^[46]。人工网络由真实网络的小数据驱动, 通过对真实网络的参数学习产生人工网络的初始参数, 并结合一定的规则学习方法生成智能体之间的交互规则、行为规则, 产生大量不同的人工网络子系统。人工网络中智能体的产生、交互和演化是完备的^[47], 能够产生大量且完备的人工网络大数据,

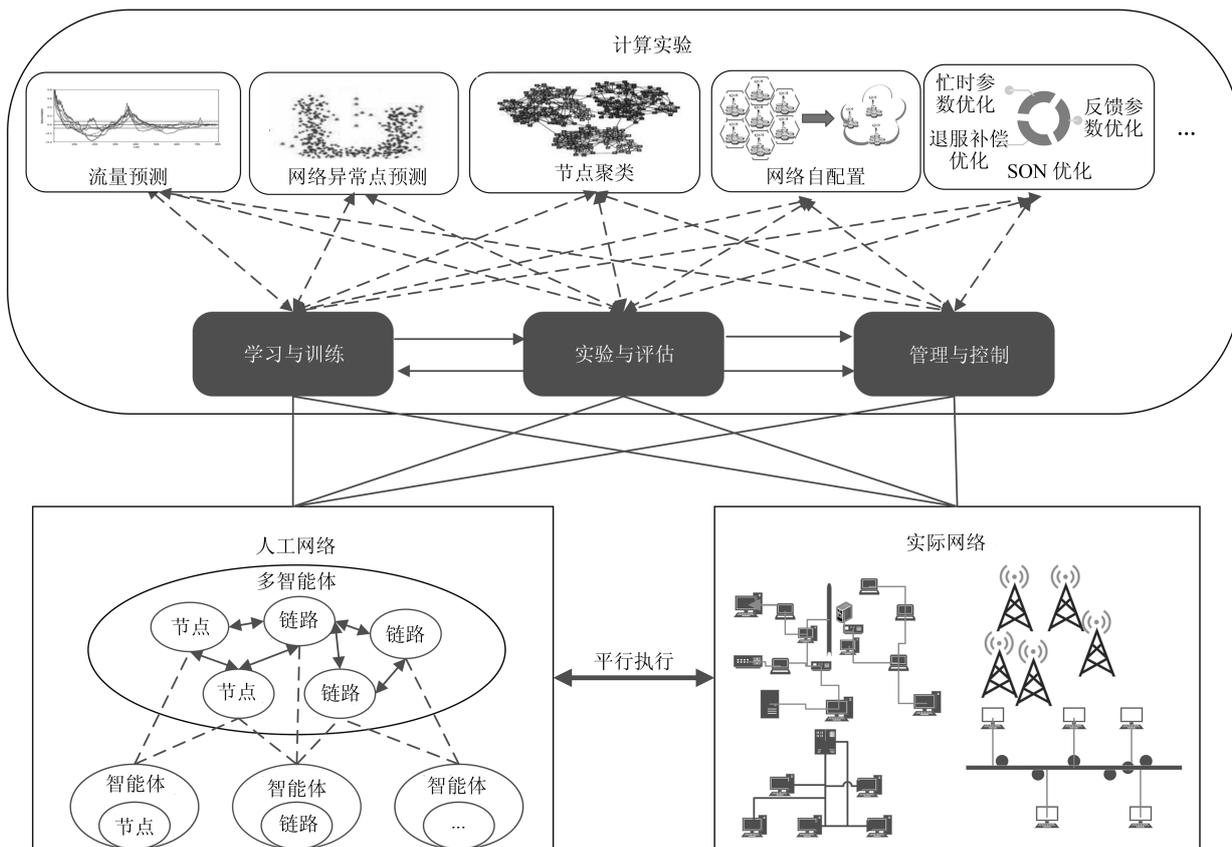


图5 平行网络及其计算解析

Fig. 5 Parallel network and its computational analysis

能够模拟所有可能的网络场景, 因而能够模拟整个网络系统.

2) 计算实验可以基于人工网络这一“计算实验室”设计各类网络智能体的组合及交互规则, 生成各种网络场景, 并且利用各种学习和训练手段求得适用于各种网络场景下的不同策略, 并基于一定的策略评估不同的方案, 得到针对不同网络场景的最佳方案^[48]. 因此, 计算实验主要包含两种模式, 即“学习与训练”和“实验与评估”^[49]. 在学习与训练模式下, 它能够自动学习和生成人工网络数据的标注信息, 从而可以基于数据学习到大量不同场景的知识, 进行许多无法在真实网络中开展的实验^[50]. 同时, 也可以设定具体的目标, 面向特定的网络场景和应用进行训练. 例如, 计算实验可以基于一定的规则生成许多危险网络场景如网络遭受攻击, 并基于此类场景的数据学习应对此类场景的策略. 在实验与评估模式下, 它能够模拟所有方案在人工网络中的效果^[51], 进而评估其在对应场景中的表现.

3) 计算实验能够基于人工网络“大数据”和实际网络“小数据”组成的“海量数据”进行大量的学习^[52], 训练并评估特定场景下最佳的网络运行方案,

此时, 人工网络的结果不再是实际网络的仿真, 而是系统运行的一种可能结果. 平行网络的目标是基于在人工网络上的计算实验训练及优化特定场景的人工网络模型参数^[53], 并通过平行执行引导实际网络朝着更优的人工网络优化运行. 一方面, 计算实验得到的优化参数和策略可以应用到实际网络上, 提升实际网络的性能; 另一方面, 人工网络接收到实际网络的反馈, 进行实时、在线的借鉴、预估和引导, 使两者平行调谐, 协同优化^[54].

4) 在数据层面, 平行网络基于实际网络数据, 借助生成式对抗网络等计算实验手段生成相对完备的“海量网络数据”, 通过对海量网络数据的学习、训练求解网络优化解并进行优化解评估, 使之具有更广泛的适用性; 在控制层面, 平行网络利用人工网络与实际网络的虚实交互、双向验证实现两者的协同进化以及对整个网络的多目标优化管理与控制. 它不仅优化整个网络, 还可以主动学习并评估网络管控效果并自适应调整未来策略, 因此更智能.

5.2 面向平行网络的计算实验平台

传统的网络仿真技术通过数学建模和数值方法模拟网络运行过程, 其主要目的是逼近真实网络的

状态^[55-56]. 而计算实验方法通过在人工网络上构建大量的网络场景生成完备的人工网络数据, 基于对人工网络大数据和实际网络数据的训练和评估生成各种网络场景下的最优方案, 以优化的人工网络引导实际网络^[57], 从而提升网络的性能.

计算实验的主要流程是: 基于机器学习、统计分析等技术, 从真实网络的数据中提取网络的特征和规律^[58]. 随后, 基于提取到的特征构建人工网络智能体的行为和交互规则并生成网络场景, 实现对真实网络的模拟, 在此基础上, 可以针对特定的目标和应用设计特定的实验方案, 进而基于完备的人工网络数据训练和评估相应场景下的控制策略^[59], 并将最符合实验目标的控制策略作为网络的优化方案.

基于以上思想, 计算实验平台的系统流程图如图 6 所示, 它包含实验设计、动态仿真、学习优化、分析评估与数据中心五个主要模块^[60], 通过模块之间的交互和协作, 实现对复杂网络系统的解析、预测和引导^[61].

实验设计模块的主要任务是协助用户制定相应的实验方案, 并生成相应的人工网络规则和场景. 它的主要功能是通过物理信号传感器和社会信号传感器获取实际网络信号和社会舆情信号并借助机器学习、数据挖掘等技术对其进行学习, 从而获取实际网络的参数. 通过自然语言处理、平行学习等知识自动化技术, 它可以解析用户对网络的需求, 从而理解网络优化任务, 并通过智能规划、专家系统等方法生成相应的实验方案. 基于解析的实验目标和方案, 它可以自动生成人工网络智能体的行为和交互规则, 进而生成相应的网络场景.

动态仿真模块基于传统的网络仿真平台, 能够执行基于智能体的网络实验, 演化出完备的人工网络数据. 它的主要功能是基于实验设计模块产生的网络场景和规则生成一定数量的智能体并模拟它们的行为及交互, 并采集实验的过程数据, 同时根据指定的目标对数据进行标注或处理.

学习优化模块的主要功能是对动态仿真产生的人工网络大数据结合实际网络数据进行分析和学习, 从而获取网络的具体知识, 产生网络的优化调参方案. 学习优化模块包含模糊控制、动态规划、人工神经网络、遗传算法、蚁群算法等控制算法, 也包含深度学习、强化学习等人工智能算法, 能够基于大量的计算实验优化网络参数. 同时, 学习优化模块还具有一定的自我学习和调整能力, 可以灵活地对计算实验用到的算法进行选择和优化.

分析评估模块主要负责对计算实验的过程和结果数据评价和分析, 从而判断最符合实验目标的网络运行方案, 并对实验过程进行检查, 查看实验过程是否产生错误并对错误进行分析和纠错.

基于大量的场景实验和训练评估, 计算实验平台能够生成大量针对不同场景的优化方案和控制策略. 为了便于在面临新的网络状况时快速给出相应的解决方案, 数据中心模块将存储各种网络场景下的最优方案, 并可基于实际网络的参数快速查询相应场景的解决方案^[53].

由此可见, 仿真器是网络智能体运行和交互的场所, 是计算实验平台的核心, 基础仿真平台的选择将直接影响到计算实验平台的性能. 平行网络计算实验平台仿真器的选择将主要基于以下几点:

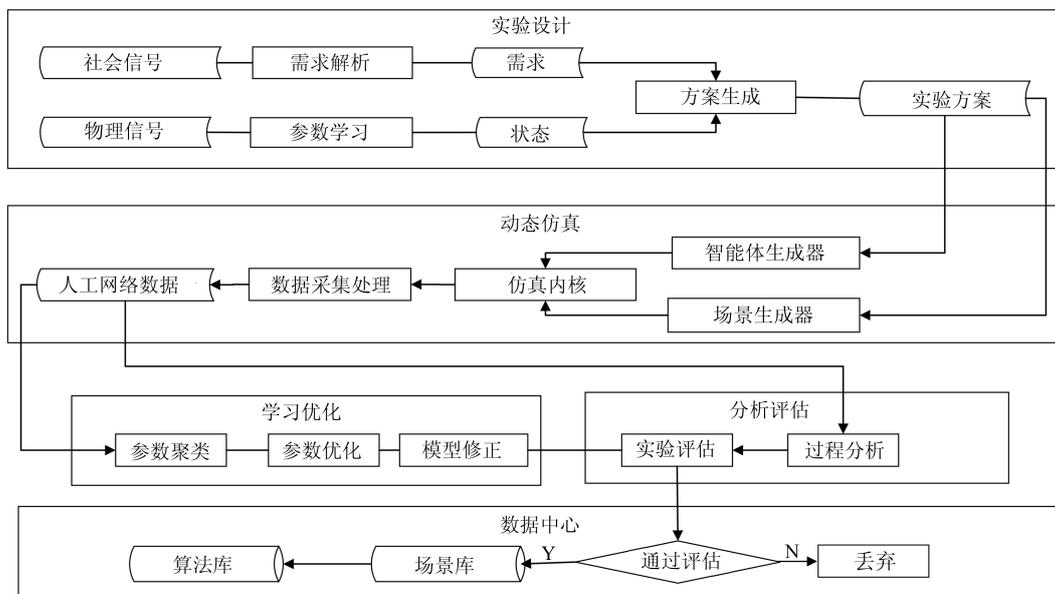


图 6 计算实验平台系统流程图

Fig. 6 System flowchart of computational experiment platform

是否拥有标准的可编程接口、是否开源、是否具有较高的灵活性和可扩展性、是否具有全面的说明文档等^[55]。基于本文对通用网络仿真平台的介绍和分析, 满足以上需求的仿真平台有很多, 例如 NS-3 等都可以作为候选的仿真器嵌入计算实验平台。

但是, 计算实验方法与仿真方法有着本质的不同, 计算实验是一种主动控制技术^[57], 它不仅能够模拟和预测真实网络的状态, 而且能够基于对完备的人工网络数据的学习找到各种网络场景下和优化参数和最佳控制方案, 进而引导实际网络系统优化运行^[60]。

基于其更全面的控制策略和全局网络的自适应优化能力, 计算实验将在下一代网络中具有广泛且重要的应用, 为网络升级和服务能力提升提供强有力的助力。例如, 当前, 中国的 4G 网络基站数已经达到了 100 万以上, 长期演进 (Long term evolution, LTE) 网络有大量的参数可供优化, 存在很大的服务质量提升空间。然而, 上万级的规模为网络参数的最优配置带来了困难, 既很难在海量的参数中选择有价值的参数进行优化, 又难以在实际基站小区中进行实验, 同时面对复杂多变的网络场景也不能自适应地进行调整。为此, 可以构造与真实小区相对应的软件定义的人工网络, 在其上进行大量的计算实验, 综合运用专家系统推荐算法、机器学习算法、聚类算法等^[78] 提取网络的个性化特征, 自动推荐个性化的网络参数, 对重要的参数计算分析其优化值。由此, 便可自动完成海量参数的选择、优化, 并可通过对大量场景的学习提升网络的自适应调节优化能力。

5.3 计算实验设计与验证

计算实验方法能够有效利用社会信号^[79-81] 和物理网络信号解析网络, 针对具体场景提供最佳的网络优化指导。为了验证计算实验方法的有效性, 本节设计了基于该方法的网络优化实验。本文以微信网络的数据传输为例, 基于多种计算实验方法分析其朋友圈链接内容被浏览并转发的可能性, 并进一步将相关数据缓存至转发者附近的服务器, 从而降低转发者的跟随好友的访问时延。

微信是国内使用最广泛的即时通信软件, 微信朋友圈之间存在一种双向链接关系^[82], 即通信双方彼此为好友。微信朋友圈只能被自己的好友看到, 但同时好友具有一定的转发权限, 进而影响好友的好友, 使信息进一步扩散出去。通常, 微信好友多为与用户拥有更多相似性的本地好友, 数据多存放在更多被访问的发布者本地服务器, 因此, 转发者及其跟随者访问数据也需要访问较远距离的本地服务器, 增加了跟随者的访问时延和本地服务器的负载。

为了从整体上提高微信用户的访问效率, Zhang

等^[83] 提出了一种对服务器重新分配位置的反向贪婪算法, 通过计算所有可能的组合形式, 并计算每种可能组合的整体通信负载选择具有最小负载的方案, 进而提升网络的服务质量。然而, 该算法需要遍历所有可能的组合, 计算复杂度高, 且该方法对网络的优化需要将服务器的位置重新安置, 成本需求较高。在通信网络中, 缓存机制被认为是一种有效的降低网络负载并提高用户体验的技术并得到了广泛的研究。通常, 缓存是将一片区域内最流行的内容存储到该区域内的服务器^[84], 当用户提出请求浏览该内容时, 可以就近从本地服务器上获取, 进而降低传输时延, 并且避免大量请求造成的通信过载。

为了更有效地优化微信数据传输网络, 我们首先基于对微信转发行为的特征提取分析微信用户的行为模式, 进而利用多种机器学习方法预测用户的转发行为, 分析用户对网络的需求。接着, 我们通过缓存机制优化网络性能, 并基于仿真平台预测和评估网络的性能, 进而验证该方案的效果。

我们基于文献 [85] 公布的微信朋友圈链接浏览记录数据集分析用户的行为特征, 该数据集由大量的五元组构成, 包含链接 ID、发布者 ID、浏览者 ID、浏览者地址、浏览时间等内容。浏览者在浏览发布者发布的链接内容时, 可能对感兴趣的链接进行转发, 而后转发者的好友仍可能浏览或转发该链接。为了分析微信朋友圈的转发行为, 我们首先对数据集进行标注, 将由转发产生的数据标注为 1, 其他标记为 0, 从而将问题转化为一个二分类问题^[86-87]。接着, 我们对能够有效预测用户行为的特征进行提取, 数据中的特征包括 (浏览日期、持续浏览时间、浏览页数、总页数、二者之比) 等。同时, 用户的个性是对其转发与否的重要影响因素, 个别用户比较倾向于转发好友的链接。为此, 我们基于数据集计算生成了一个新的特征, 即用户的历史转发频度。在此, 我们选取 3 个主观上最有价值的特征, 即持续浏览时间、浏览页数/总页数、用户的历史转发频度。

分类算法的性能一般用准确率来表示, 其定义如式 (1) 所示, 其中, TP、TN、FP、FN 分别表示转发样本被正确预测数、未转发样本被正确预测数、未转发样本被错误预测数及转发样本被正确预测数。

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

通常, 转发数据相比一般的浏览数据较为稀疏, 在本数据集中只有 2.5% 左右。而为了较高精度的做到提前缓存, 我们更关注转发数据被正确预测的比例, 即分类模型的召回率:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

随机选取一个链接, 提取其数据记录并将其分为训练集和测试集, 基于不同的特征组合和分类算法, 我们得到转发预测的召回率如表 5 所示. 表中未列出的特征和分类算法组合不能正确分类转发与未转发数据, 由预测结果可知, 基于浏览时间和历史转发频度, 以支持向量机 (Support vector machine, SVM) 作为分类器可以达到最高的分类性能, 其预测召回率高达 92.07%. 因此, 这两类简单的社交网络特征可以帮助我们精准预测用户的转发行为, 进而为网络优化提供参考.

表 5 转发预测召回率

Table 5 Recall rate of retweet prediction

| | 线性分类器 | 决策树 | SVM | 随机森林 |
|--------------------------|-------|-------|-------|-------|
| 历史转发频度 | 5.6 | 71.09 | 89.9 | 71.09 |
| 浏览时间 + 历史转发频度 | 5.6 | 71.23 | 92.07 | 0 |
| 浏览页数/总页数 + 历史转发频度 | 5.7 | 71.09 | 3.03 | 0 |
| 浏览时间 + 浏览页数/总页数 + 历史转发频度 | 5.91 | 71.09 | 2.38 | 57.1 |

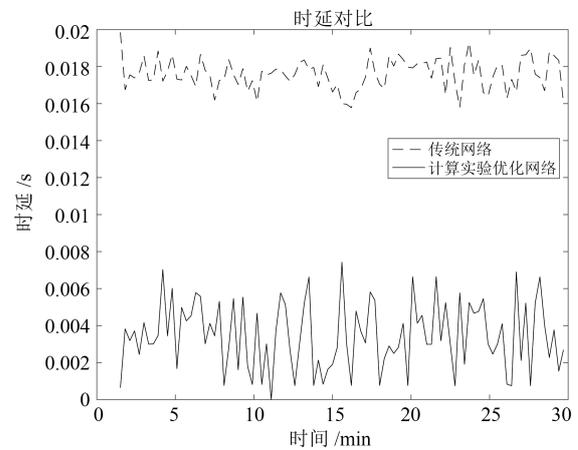
基于对用户行为和网络需求的精准分析, 可以通过将数据预先缓存至转发者附近的服务器, 降低后续浏览者的访问时延, 优化网络的服务. 为了验证该方案的效果, 我们选取了一段时间内某链接的浏览记录进行了简化处理. 本实验通过 OPNET 仿真软件进行了对比仿真, 对相应的时延和本地服务器吞吐量进行了分析. 在该实验中, 共有 50 个不同省市的节点向位于北京的发布者本地服务器请求数据, 其中, 有 16 个节点为位于广东的转发者的本地跟随者, 根据以上设计, 本实验设定了两种实验场景, 一种为所有节点向北京本地服务器发送请求, 另一种为基于转发预测结果将数据缓存至广州本地服务器, 跟随者节点向转发者本地服务器发送请求. 网络为客户机/服务器模式, 网络拓扑结构为星型, 服务器节点模型选择 Sm_Int_server, 客户机节点模型为 Sm_Int_wkstn, 网络业务为 Light database 业务, 链路带宽为 10 Mbps, 仿真时间设置为 30 分钟.

两种方式的整体时延和北京本地服务器的负载分别如图 7(a) 和 7(b) 所示. 可见, 基于转发预测及缓存方法, 可以有效降低用户的平均访问时延, 同时, 服务器的负载得以均衡化, 有利于整体提升网络的利用率和服务质量. 因此, 基于计算实验方法, 能够基于数据有效解析用户的网络需求, 并评估网络优化方案的效果, 以此指导实际网络的优化.

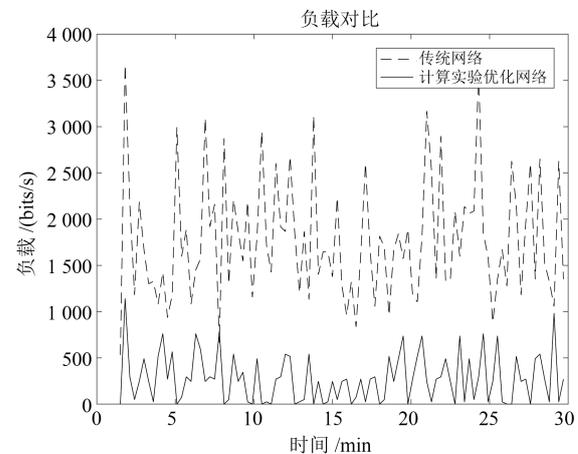
5.4 小结与讨论

本节介绍了平行网络架构及其计算实验方法, 该方法基于实际网络数据建模人工网络, 并进一步

生成完备的人工网络大数据, 同时, 它可以基于社会信号解析用户的网络需求, 基于特定的目标从人工网络大数据和实际网络数据中学习和训练针对具体场景的网络优化方案. 基于微信朋友圈转发预测和内容缓存的网络优化实验证明计算实验方法可以有效提升网络的服务能力, 更好地满足用户的需要.



(a) Comparison of network transmission delay



(b) Comparison of network load

图 7 传统网络与计算实验优化性能对比

Fig. 7 Performance comparison of traditional method and parallel network computational experiment

6 总结与展望

网络仿真技术为网络设计、开发和诊断提供了强有力的工具. 本文全面调研了主流的通用型网络仿真平台, 对各种软件的主要特点和技术基础进行了介绍, 并对比分析了不同平台的优缺点. 同时, 本文还对网络仿真技术的主要问题和发展趋势进行了分析, 在此基础上介绍了基于 ACP 方法的平行网络架构, 并重点介绍了其计算实验平台.

平行网络有效提升了网络的灵活性及自适应优化能力, 其计算实验方法综合仿真、数据挖掘、机器

学习及知识自动化等技术, 以可计算、可比较、可实现的方式解析复杂的网络系统, 为未来网络的智能管理和控制提供了新思路及方法。

此外, 网络系统是一个涉及到多方个体的复杂系统, 与用户的主观和客观因素有很大关系。ACP方法中基于多智能体的建模方式能够有效模拟各种个体的行为及其相互影响, 为车联网等多种网络系统的规划、管理和资源分配等提供适应的解决方案, 使网络从“人适应系统”向“系统适应人”转变。平行网络系统作为一个集成验证平台, 必将成为未来网络系统发展的一个重要方向, 为不断发展的网络业务需求, 特别是物联网^[88-89]的需求提供有力支持。

References

- Wang Fei-Yue, Yang Liu-Qing, Hu Xiao-Ya, Cheng Xiang, Han Shuang-Shuang, Yang Jian. Parallel networks and network softwarization: a novel network architecture. *Scientia Sinica Informationis*, 2017, **47**(7): 811-831
(王飞跃, 杨柳青, 胡晓娅, 程翔, 韩双双, 杨坚. 平行网络与网络软件化: 一种新颖的网络架构. 中国科学: 信息科学, 2017, **47**(7): 811-831)
- Yu Chen, Wang Yan-Nian. The status and prospect of computer network application. *China Computer & Communication*, 2018, (9): 168-169
(俞晨, 王延年. 计算机网络应用的现状与展望. 信息与电脑(理论版), 2018, (9): 168-169)
- Yang Ren-Yu, Han Yi-Gang, Zhang Fan, Feng Fei. Survey of content centric network based on SDN. *Computer Science*, 2019, **46**(1): 13-20
(杨仁宇, 韩毅刚, 章帆, 冯飞. 基于 SDN 的内容中心网络研究综述. 计算机科学, 2019, **46**(1): 13-20)
- Breslau L, Estrin D, Fall K, Floyd S, Heidemann J, Helmy A, et al. Advances in network simulation. *Computer*, 2000, **33**(5): 59-67
- 方维维, 王锐. 网络模拟和仿真技术. 计算机教育, 2005, (12): 51-54
- Hou Zong-Hao, Wang Bing-Kang, Huang Yong-Xiang. Research of the network simulation. *Computer Simulation*, 2003, **20**(10): 89-91, 136
(侯宗浩, 王秉康, 黄泳翔. 网络仿真的研究. 计算机仿真, 2003, **20**(10): 89-91, 136)
- 吕方瑶. 当前主要网络仿真工具研究和实践. 信息通信, 2011, (4): 3-5
- Heidemann J, Mills K, Kumar S. Expanding confidence in network simulations. *IEEE Network*, 2001, **15**(5): 58-63
- Mehta S, Sulatan N, Kwak K S. Network and system simulation tools for next generation networks: a case study. *Modelling, Simulation and Identification*. InTech, 2010.
- Yuan Xiao, Cai Zhi-Ping, Liu Shu-Hao, Yu Ying. Large-scale network emulation software and its key technologies. *Computer Technology and Development*, 2014, (7): 9-12
(袁晓, 蔡志平, 刘书昊, 喻颖. 大规模网络仿真软件及其仿真技术. 计算机技术与发展, 2014, (7): 9-12)
- Zhang Xian-Jun, Li He. Computer network simulation technology and simulation tools. *Ship Electronic Engineering*, 2013, **33**(1): 25-27
(张贤军, 李鹤. 计算机网络仿真技术及仿真工具. 舰船电子工程, 2013, **33**(1): 25-27)
- Siraj S, Gupta A K, Badgujar R. Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 2012, **1**(4): 201-210
- Issariyakul T, Hossain E. Introduction to network simulator 2 (NS2). *Introduction to Network Simulator NS2*. Boston, MA: Springer, 2012. 21-40
- Ru Xin-Yu, Liu Yuan. Analysis and research on network simulator 3. *Computer Technology and Development*, 2018, **28**(3): 72-77
(茹新宇, 刘渊. 网络仿真器 NS3 的剖析与探究. 计算机技术与发展, 2018, **28**(3): 72-77)
- Bagrodia R, Meyer R, Takai M, Chen Y A, Zeng X, Martin J, et al. Parsec: a parallel simulation environment for complex systems. *Computer*, 1998, **31**(10): 77-85
- Zeng X, Bagrodia R, Gerla M. GloMoSim: a library for parallel simulation of large-scale wireless networks. *ACM SIGSIM Simulation Digest*, 1998, **28**(1): 154-161
- Farooq J, Rauf B. Implementation and Evaluation of IEEE 802.11e Wireless LAN in GloMoSim. Department of Computing Science Umea University Sweden, 2006.
- Yang Lu-Ming, Li Xie-Hua. Comparison & analysis of network simulators and discussion on process of network simulation. *Computing Technology and Automation*, 2002, **21**(4): 65-70
(杨路明, 李谢华. 网络仿真软件比较分析及仿真过程探讨. 计算技术与自动化, 2002, **21**(4): 65-70)
- Spaho E, Barolli L, Mino G, Xhafa F, Kolici F. VANET simulators: a survey on mobility and routing protocols. In: Proceedings of the 2011 International Conference on Broadband and Wireless Computing, Communication and Applications. Barcelona, Spain: IEEE, 2011. 1-10
- Shi Huai-Wei, Wang Hua, Zhang Nian-Jun, Zhao Lin-Sheng, Si Tong-Guang. Analysis of OPNET simulation technology and its application. *Computer Engineering and Design*, 2006, **27**(17): 3309-3311
(石怀伟, 王华, 张念军, 赵林胜, 同光. OPNET 网络仿真技术及其应用分析. 计算机工程与设计, 2006, **27**(17): 3309-3311)
- Kumar A, Kaushik S K, Sharma R, Raj P. Simulators for wireless networks: a comparative study. In: Proceedings of the 2012 IEEE International Conference on Computing Sciences. Phagwara, India: IEEE, 2012. 338-342
- Chen Min. *OPNET IOT Simulation*. Wuhan: Huazhong University of Science and Technology Press, 2015. 55-75
(陈敏. OPNET 物联网仿真. 武汉: 华中科技大学出版社, 2015. 55-75)
- 陈敏. OPNET 网络仿真. 北京: 清华大学出版社, 2004. 48-53
- Zhou Hui. The Study of OPNET Simulation and its Application [Master thesis], Wuhan University of Science and Technology, China, 2009. 15-18
(周慧. OPNET 网络仿真及其应用研究 [硕士学位论文], 武汉科技大学, 中国, 2009. 15-18)
- Miller J A, Nair R S, Zhang Z, Zhao H. JSIM: a JAVA-based simulation and animation environment. In: Proceedings of IEEE 30th Annual Proceedings of Simulation Symposium. Atlanta, Georgia, USA: IEEE, 1997. 31-42
- Cao N, Liu P Z, Li G F, Zhang C, Cao S H, Cao G S, et al. Evaluation models for the nearest closer routing protocol in wireless sensor networks. *IEEE Access*, 2018, **6**: 77043-77054
- Kargl F, Schoch E. Simulation of MANETs: a qualitative comparison between JiST/SWANS and NS-2. In: Proceedings of the 1st International Workshop on System Evaluation for Mobile Platforms. San Juan, Puerto Rico: IEEE, 2007. 41-46

- 28 Barr R, Haas Z J, van Renesse R. JiST: an efficient approach to simulation using virtual machines. *Software: Practice and Experience*, 2005, **35**(6): 539–576
- 29 Riley G F. The Georgia tech network simulator. In: Proceedings of the 2003 ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research. Karlsruhe, Germany: ACM, 2003. 5–12
- 30 Liu Tao, Nie Xiao-Feng, Jing Ji-Wu, Wang Yue-Wu. Memory management in worm simulation based on small object memory allocation technique on the GTNetS. *Journal of Graduate University of Chinese Academy of Sciences*, 2012, **29**(1): 131–135
(刘涛, 聂晓峰, 荆继武, 王跃武. 基于小型对象分配技术的 GTNetS 蠕虫仿真内存管理. 中国科学院大学学报, 2012, **29**(1): 131–135)
- 31 Cowie J, Ogielski A, Nicol D. The SSFNet network simulator [Online], <http://www.ssfnet.org/homePage.html>, 2002
- 32 Yoon S, Kim Y B. A design of network simulation environment using SSFNet. In: Proceedings of the 1st International Conference on Advances in System Simulation. Porto, Portugal: IEEE, 2009. 73–78
- 33 Yang Jun-Dong, Yin Qiang, Zhang Shuo. Simulation and performance analysis of SDN based on Mininet. *Information & Communications*, 2017, (3): 189–191
(杨俊东, 尹强, 张硕. 基于 Mininet 的 SDN 仿真与性能分析. 信息通信, 2017, (3): 189–191)
- 34 Wang S Y. Comparison of SDN OpenFlow network simulator and emulators: EstiNet vs. Mininet. In: Proceedings of the 2014 IEEE Symposium on Computers and Communication. Funchal, Portugal: IEEE, 2014. 1–6
- 35 Wang S Y, Chou C L, Yang C M. EstiNet openflow network simulator and emulator. *IEEE Communications Magazine*, 2013, **51**(9): 110–117
- 36 Imran M, Said A M, Hasbullah H. A survey of simulators, emulators and testbeds for wireless sensor networks. In: Proceedings of the 2010 International Symposium on Information Technology. Kuala Lumpur, Malaysia: IEEE, 2010. 897–902
- 37 Zhou S J, Zhou Y H, Luo J Q. Research on simulators for peer-to-peer systems. In: Proceedings of the 2013 IEEE 3rd International Conference on Computer Science and Network Technology. Dalian, China: IEEE, 2013. 726–731
- 38 Owczarek P, Zwierzykowski P. Review of simulators for wireless mesh networks. *Journal of Telecommunications Information Technology*, 2014, **14**(3): 82–89
- 39 Wang Run-Min, Deng Xiao-Feng, Xu Zhi-Gang, Zhao Xiang-Mo. Survey on simulation testing and evaluation of internet of vehicles. *Application Research of Computers*, 2019, (7): 1–9
(王润民, 邓晓峰, 徐志刚, 赵祥模. 车联网仿真测试评价技术研究综述. 计算机应用研究, 2019, (7): 1–9)
- 40 McKeown N. Software-defined networking. *INFOCOM Keynote Talk*, 2009, **17**(2): 30–32
- 41 Wang Fei-Yue. Parallel system methods for management and control of complex systems. *Control and Decision*, 2004, **19**(5): 485–489
(王飞跃. 平行系统方法与复杂系统的管理和控制. 控制与决策, 2004, **19**(5): 485–489)
- 42 Shen D Y, Wang X, Wang J, Guan X Y, Yang P H, Xu L. Parallel intermodal road-rail transportation system based on ACP approach. In: Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics. Banff, Canada: IEEE, 2017. 444–449
- 43 Wang Fei-Yue. Artificial societies, computational experiments, and parallel systems: a discussion on computational theory of complex social-economic systems. *Complex Systems and Complexity Science*, 2004, **1**(4): 25–35
(王飞跃. 人工社会、计算实验、平行系统 — 关于复杂社会经济系统计算研究的讨论. 复杂系统与复杂性科学, 2004, **1**(4): 25–35)
- 44 Wang Fei-Yue, Yang Jian, Han Shuang-Shuang, Yang Liu-Qing, Cheng Xiang. The framework of parallel network based on the parallel system theory. *Journal of Command & Control*, 2016, **2**(1): 71–77
(王飞跃, 杨坚, 韩双双, 杨柳青, 程翔. 基于平行系统理论的平行网络架构. 指挥与控制学报, 2016, **2**(1): 71–77)
- 45 Wang F Y, Yang L Q, Cheng X, Han S S, Yang J. Network softwareization and parallel networks: beyond software-defined networks. *IEEE Network*, 2016, **30**(4): 60–65
- 46 Wang Fei-Yue, Wang Xiao, Yuan Yong, Wang Tao, Lin Yi-Lun. Social computing and computational societies: the foundation and consequence of smart societies. *Chinese Science Bulletin*, 2015, **60**(5–6): 460–469
(王飞跃, 王晓, 袁勇, 王涛, 林懿伦. 社会计算与计算社会: 智慧社会的基础与必然. 科学通报, 2015, **60**(5–6): 460–469)
- 47 Liu Shuo, Wang Shuai, Meng Qing-Zhen, Ye Pei-Jun, Wang Tao, Huang Wen-Lin, et al. Parallel modeling of criminal subjects behavior based on ACP behavioral dynamics. *Acta Automatica Sinica*, 2018, **44**(2): 251–261
(刘烁, 王帅, 孟庆振, 叶佩军, 王涛, 黄文林, 等. 基于 ACP 行为动力学的犯罪主体行为平行建模分析. 自动化学报, 2018, **44**(2): 251–261)
- 48 Zhang W W, Wang K F, Qu H, Zhao J H, Wang F Y. Scene-specific pedestrian detection based on parallel vision. arXiv preprint arXiv: 1712.08745, 2017.
- 49 Yuan Yong, Wang Fei-Yue. Parallel Blockchain: concept, methods and issues. *Acta Automatica Sinica*, 2017, **43**(10): 1703–1712
(袁勇, 王飞跃. 平行区块链: 概念、方法与内涵解析. 自动化学报, 2017, **43**(10): 1703–1712)
- 50 Liu Xin, Wang Xiao, Zhang Wei-Shan, Wang Jian-Ji, Wang Fei-Yue. Parallel data: from big data to data intelligence. *Pattern Recognition & Artificial Intelligence*, 2017, **30**(8): 673–681
(刘昕, 王晓, 张卫山, 汪建基, 王飞跃. 平行数据: 从大数据到数据智能. 模式识别与人工智能, 2017, **30**(8): 673–681)
- 51 Li Li, Lin Yi-Lun, Cao Dong-Pu, Zheng Nan-Ning, Wang Fei-Yue. Parallel learning: a new framework for machine learning. *Acta Automatica Sinica*, 2017, **43**(1): 1–8
(李力, 林懿伦, 曹东璞, 郑南宁, 王飞跃. 平行学习 机器学习的一个新型理论框架. 自动化学报, 2017, **43**(1): 1–8)
- 52 Wang Fei-Yue, Zhang Mei, Meng Xiang-Bing, Wang Rong, Wang Xiao, Zhang Zhi-Cheng, et al. Parallel surgery: an ACP-based approach for intelligent operations. *Pattern Recognition & Artificial Intelligence*, 2017, **30**(11): 961–970
(王飞跃, 张梅, 孟祥冰, 王蓉, 王晓, 张志成, 等. 平行手术: 基于 ACP 的智能手术计算方法. 模式识别与人工智能, 2017, **30**(11): 961–970)
- 53 Wang Fei-Yue, Gao Yan-Chen, Shang Xiu-Qin, Zhang Jun. Parallel manufacturing and industries 5.0: from virtual manufacturing to intelligent manufacturing. *Science & Technology Review*, 2018, **36**(21): 10–22
(王飞跃, 高彦臣, 商秀芹, 张俊. 平行制造与工业 5.0: 从虚拟制造到智能制造. 科技导报, 2018, **36**(21): 10–22)
- 54 Meng Xiang-Bing, Wang Rong, Zhang Mei, Wang Fei-Yue. Parallel perception: an ACP-based approach to visual SLAM. *Journal of Command and Control*, 2017, **3**(4): 350–358
(孟祥冰, 王蓉, 张梅, 王飞跃. 平行感知: ACP 理论在视觉 SLAM 技术中的应用. 指挥与控制学报, 2017, **3**(4): 350–358)

- 55 Yang Liu-Qing, Wang Fei-Yue, Zhang Yan-Li, Han Shuang-Shuang, Yang Jian, Zhao Kai, et al. The urban parallel parking system based on ACP approach. *Journal of Command and Control*, 2015, **1**(4): 384–390
(杨柳青, 王飞跃, 张艳丽, 韩双双, 杨坚, 赵恺, 等. 基于 ACP 方法的城市平行停车系统. 指挥与控制学报, 2015, **1**(4): 384–390)
- 56 Wang Xiao, Yao Ting-Ting, Han Shuang-Shuang, Cao Dong-Pu, Wang Fei-Yue. Parallel internet of vehicles: the ACP-based networked management and control for intelligent vehicles. *Acta Automatica Sinica*, 2018, **44**(8): 1391–1404
(王晓, 姚婷婷, 韩双双, 曹东璞, 王飞跃. 平行车联网: 基于 ACP 的智能车辆网联管理与控制. 自动化学报, 2018, **44**(8): 1391–1404)
- 57 Wang Fei-Yue, Sun Qi, Jiang Guo-Jin, Tan Ke, Zhang Jun, Hou Jia-Chen, et al. Nuclear Energy 5.0: new formation and system architecture of nuclear power industry in the new IT era. *Acta Automatica Sinica*, 2018, **44**(5): 922–934
(王飞跃, 孙奇, 江国进, 谭珂, 张俊, 侯家琛, 等. 核能 5.0: 智能时代的核电工业新形态与体系架构. 自动化学报, 2018, **44**(5): 922–934)
- 58 Bai Tian-Xiang, Wang Shuai, Shen Zhen, Cao Dong-Pu, Zheng Nan-Ning, Wang Fei-Yue. Parallel robotics and parallel unmanned systems: framework, structure, process, platform and applications. *Acta Automatica Sinica*, 2017, **43**(2): 161–175
(白天翔, 王帅, 沈震, 曹东璞, 郑南宁, 王飞跃. 平行机器人与平行无人系统: 框架、结构、过程、平台及其应用. 自动化学报, 2017, **43**(2): 161–175)
- 59 Wang Fei-Yue, Li Chang-Gui, Guo Yuan-Yuan, Wang Jing, Wang Xiao, Qiu Tian-Yu, et al. Parallel gout: an ACP-based system framework for gout diagnosis and treatment. *Pattern Recognition and Artificial Intelligence*, 2017, **30**(12): 1057–1068
(王飞跃, 李长贵, 国元元, 王静, 王晓, 邱天雨, 等. 平行高特: 基于 ACP 的平行痛风诊疗系统框架. 模式识别与人工智能, 2017, **30**(12): 1057–1068)
- 60 Liu Zhi-Jie, Ouyang Yun-Cheng, Song Yu-Hua, He Wei, Wang Fei-Yue. Parallel control of distributed parameter systems: from model based control to data driven intelligent control. *Journal of Command & Control*, 2017, **3**(3): 177–185
(刘志杰, 欧阳云呈, 宋宇骅, 贺威, 王飞跃. 分布参数系统的平行控制: 从基于模型的控制到数据驱动的智能控制. 指挥与控制学报, 2017, **3**(3): 177–185)
- 61 Yin Pei-Li, Wang Jian-Hua, Chen Yang-Quan, Wang Fei-Yue. Parallel measurements: a new theory and framework for complex measurement system and a case study. *Acta Automatica Sinica*, 2018, **44**(3): 425–433
(尹培丽, 王建华, 陈阳泉, 王飞跃. 平行测量: 复杂测量系统的一个新型理论框架及案例研究. 自动化学报, 2018, **44**(3): 425–433)
- 62 Floyd S, Paxson V. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, 2001, **9**(4): 392–403
- 63 Gehrke J, Madden S. Query processing in sensor networks. *IEEE Pervasive Computing*, 2004, **3**(1): 46–55
- 64 Lu S W, Schormans J A. Simulation acceleration techniques for mobile ad hoc networks. In: Proceedings of the 2003 London Communications Symposium. London, UK: IEEE, 2003. 9–14
- 65 Hogue L, Bouvry P, Guinand F. An verview of MANETs simulation. *Electronic Notes in Theoretical Computer Science*, 2006, **150**(1): 81–101
- 66 Lessmann J, Janacik P, Lachev L, Orfanus D. Comparative study of wireless network simulators. In: Proceedings of the 7th International Conference on Networking. Cancun, Mexico: IEEE, 2008. 517–523
- 67 Suliman I A, Abdalla G M T. Comparative study of wireless LAN using OPNET and NS-2. In: Proceedings of the 2016 IEEE Conference of Basic Sciences and Engineering Studies. Khartoum, Sudan: IEEE, 2016. 196–200
- 68 da Fontoura A C, de Oliveira Schmidt R, Trentin M A S. Simulation of Ad-hoc routing protocols in an interurban university mesh network. *IEEE Latin America Transactions*, 2009, **7**(1): 122–125
- 69 Wang Bo, Zhou Zhi-Wei. Comparative analysis on network simulation software NS2 and OPNET. *Computer Systems & Applications*, 2010, **19**(6): 90–95
(王波, 周志伟. 网络模拟软件 NS2 与 OPNET 的剖析比较. 计算机系统应用, 2010, **19**(6): 90–95)
- 70 Weingartner E, Vom Lehn H, Wehrle K. A performance comparison of recent network simulators. In: Proceedings of the 2009 IEEE International Conference on Communications. Dresden, Germany: IEEE, 2009. 1–5
- 71 Walsh K, Sirer E G. Staged simulation: a general technique for improving simulation scale and performance. *ACM Transactions on Modeling and Computer Simulation*, 2004, **14**(2): 170–195
- 72 Li Z, Guo G, Chen B, Qiu X G. Accelerating large scale artificial society simulation with CPU/GPU based heterogeneous parallel method. In: Proceedings of the 19th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications. Chengdu, China: IEEE, 2015. 155–162
- 73 Lin S M, Cheng X Q, Lv J M. A visualized parallel network simulator for modeling large-scale distributed applications. In: Proceedings of the 2007 Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies. Adelaide, Australia: IEEE, 2007. 339–346
- 74 Jin D, Zheng Y, Nicol D M. A parallel network simulation and virtual time-based network emulation testbed. *Journal of Simulation*, 2014, **8**(3): 206–214
- 75 White B, Lepreau J, Stoller L, Ricci R, Guruprasad S, Newbold M, et al. An integrated experimental environment for distributed systems and networks. *ACM SIGOPS Operating Systems Review*, 2002, **36**(SI): 255–270
- 76 Yang F, Wang Z K, Chen J Y, Liu Y J. Design of parallel packet switch simulation system based on NS2. In: Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing. Beijing, China: IEEE, 2009. 1–4
- 77 Gawlowicz P, Zubow A. ns3-gym: extending OpenAI gym for networking research. arXiv preprint arXiv: 1810.03943, 2018
- 78 Wang Lei. Intelligent LTE configuration optimization system based on machine learning technology. *Telecom Engineering Technics and Standardization*, 2018, (1): 39–42
(王磊. 基于机器学习技术的 LTE 网络智能优化系统设计. 电信工程技术与标准化, 2018, (1): 39–42)
- 79 Wang X, Zeng K, Zhao X L, Wang F Y. Using web data to enhance traffic situation awareness. In: Proceedings of the 17th International Conference on Intelligent Transportation Systems. Qingdao, China: IEEE, 2014. 195–199
- 80 Zeng K, Liu W L, Wang X, Chen S H. Traffic congestion and social media in China. *IEEE Intelligent Systems*, 2013, **28**(1): 72–77
- 81 Zheng X H, Chen W, Wang P, Shen D Y, Chen S H, Wang X, et al. Big data for social transportation. *IEEE Transactions on Intelligent Transportation Systems*, 2016, **17**(3): 620–630

- 82 Onnela J P, Saramäki J, Hyvönen J, Szabó G, Lazer D, Kaski K, et al. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences of the United States of America*, 2007, **104**(18): 7332–7336
- 83 Zhang Y X, Li Z Q, Gao C L, Bian K G, Song L Y, Dong S L, et al. Mobile social big data: WeChat moments dataset, network applications, and opportunities. *IEEE Network*, 2018, **32**(3): 146–153
- 84 Elsayed S A, Abdelhamid S, Hassanein H S. Proactive caching at parked vehicles for social networking. In: *Proceedings of the 2018 IEEE International Conference on Communications*. Kansas City, MO, USA: IEEE, 2018. 1–6
- 85 Zhang Y X, Li Z Q, Gao C L, Bian K G, Song L Y, Dong S L, et al. WeChatNet, a mobile social big dataset of WeChat moments [Online], available: <https://github.com/pkumobile/WMdata>, April 4, 2010.
- 86 Wang Z T, Chen C Y, Li W J. Predictive network representation learning for link prediction. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Shinjuku, Tokyo, Japan: ACM, 2017. 969–972
- 87 Shan S Q, Liu M N, Xu X B. Analysis of the key influencing factors of haze information dissemination behavior and motivation in WeChat. *Information Discovery and Delivery*, 2017, **45**(1): 21–29
- 88 Wang Fei-Yue, Zhang Jun. Internet of minds: the concept, issues and platforms. *Acta Automatica Sinica*, 2017, **43**(12): 2061–2070
(王飞跃, 张俊. 物联网: 概念、问题和平台. *自动化学报*, 2017, **43**(12): 2061–2070)
- 89 Wang Fei-Yue, Zhang Jun, Zhang Jun, Wang Xiao. Industrial internet of minds: concept, technology and application. *Acta Automatica Sinica*, 2018, **44**(9): 1606–1617
(王飞跃, 张军, 张俊, 王晓. 工业物联网: 基本概念、关键技术与核心应用. *自动化学报*, 2018, **44**(9): 1606–1617)



杨林瑶 中国科学院自动化研究所复杂系统管理与控制国家重点实验室硕士研究生。2017 年获得山东大学信息科学与工程学院学士学位。主要研究方向为车联网大数据分析智能交通。

E-mail: yanglinyao2017@ia.ac.cn

(**YANG Lin-Yao** Master student at the the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. He received his bachelor degree in internet of things engineering from Shandong University in 2017. His research interest covers parallel internet of vehicles, big data analysis and intelligent transportation.)



韩双双 中国科学院自动化研究所复杂系统管理与控制国家重点实验室助理研究员。2013 年获得加拿大阿尔伯塔大学博士学位。主要研究方向为平行网络, 物联网, 智能交通, 无线通信关键技术。

E-mail: shuangshuang.han@ia.ac.cn

(**HAN Shuang-Shuang** Assistant professor at the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. She

received her Ph.D. degree from University of Alberta, Canada, in 2013. Her research interest covers parallel networks, internet of things, intelligent transportation system and wireless communications.)

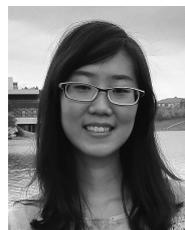


王晓 中国科学院自动化研究所复杂系统管理与控制国家重点实验室副研究员。2016 年获得中国科学院大学社会计算博士学位。主要研究方向为社会交通, 动态网群组织, 人工智能和社交网络分析。本文通信作者。

E-mail: x.wang@ia.ac.cn

(**WANG Xiao** Associate professor

at the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. She received her Ph.D. degree in social computing from University of Chinese Academy of Sciences in 2016. Her research interest covers social transportation, cyber movement organizations, artificial intelligence and social network analysis. Corresponding author of this paper.)



李玉珂 中国科学院自动化研究所复杂系统管理与控制国家重点实验室博士研究生。2014 年获得西安电子科技大学通信工程学士学位。主要研究方向为车联网, 物联网, 智能交通。

E-mail: liyuke2014@ia.ac.cn

(**LI Yu-Ke** Ph.D. candidate at the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. She received her bachelor degree in communication engineering from Xidian University, in 2014. Her research interest covers internet of vehicles, internet of things, intelligent transportation system.)



王飞跃 中国科学院自动化研究所复杂系统管理与控制国家重点实验室主任, 国防科技大学军事计算实验与平行系统技术研究中心主任, 中国科学院大学中国经济与社会安全研究中心主任, 青岛智能产业技术研究院院长。主要研究方向为平行系统的方法与应用, 社会计算, 平行智能以及知识自动化。

E-mail: feiyue.wang@ia.ac.cn

(**WANG Fei-Yue** State specially appointed expert and director of the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. Professor of the Research Center for Computational Experiments and Parallel Systems Technology, National University of Defense Technology. Director of China Economic and Social Security Research Center in University of Chinese Academy of Sciences. Dean of Qingdao Academy of Intelligent Industries. His research interest covers methods and applications for parallel systems, social computing, parallel intelligence, and knowledge automation.)