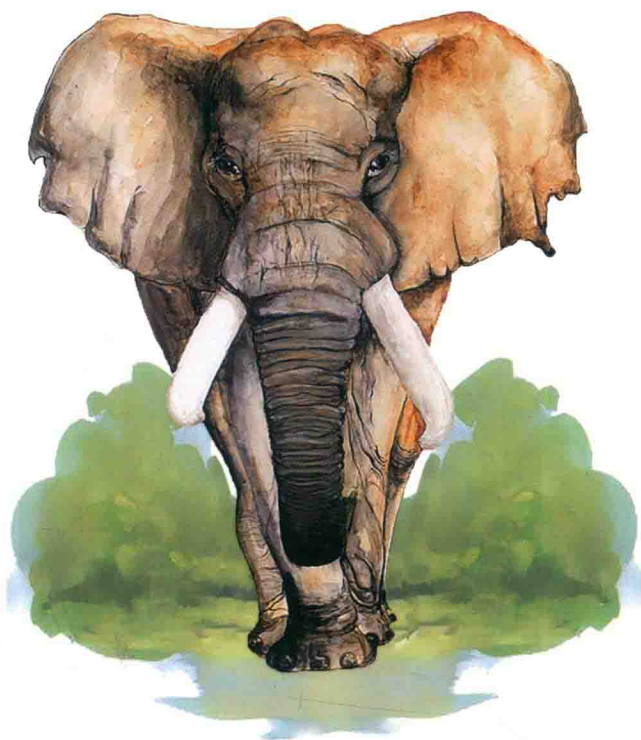


ns-3项目创始人、华盛顿大学Tom Henderson教授为本书作序

伴随着互联网上新型协议的普及和5G网络的问世，ns-3开源网络模拟器正在逐渐展现它强大的生命力。本书从使用和架构角度出发，首先全面讲解了ns-3的开源生态圈、使用方法和设计模式，在此基础上自顶向下地系统阐述了从应用层到物理层的各种ns-3关键模型。



开源网络模拟器ns-3 架构与实践

周迪之 / 编著



机械工业出版社
CHINA MACHINE PRESS

开源网络模拟器 ns-3 ——架构与实践

周迪之 编著

机械工业出版社

ns-3 是国际上一款主流的开源网络模拟器。本书深入浅出地讲解了 ns-3 的软件架构和使用方法, 希望能够降低 ns-3 的学习门槛, 让读者更加流畅地阅读并理解源代码。本书的目标读者为计算机网络专业的本科生、硕士和博士研究生, 以及工作中使用 ns-3 进行模拟实验的工程师们。

图书在版编目 (CIP) 数据

开源网络模拟器 ns-3: 架构与实践 / 周迪之编著. —北京: 机械工业出版社, 2018.11

ISBN 978-7-111-61340-4

I. ①开… II. ①周… III. ①计算机网络—计算机模拟
IV. ①TP393.01

中国版本图书馆 CIP 数据核字 (2018) 第 259880 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 张淑谦 责任编辑: 张淑谦

责任校对: 张艳霞 责任印制: 张 博

三河市宏达印刷有限公司印刷

2019 年 1 月 · 第 1 版第 1 次印刷

184mm×260mm · 12.25 印张 · 295 千字

0001—3000 册

标准书号: ISBN 978-7-111-61340-4

定价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
电话服务

服务咨询热线: (010) 88361066

读者购书热线: (010) 68322694

(010) 88379203

封面防伪标均为盗版

网络服务

机工官网: www.cmpbook.com

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

金书网: www.golden-book.com



献给我的故乡——白鹿原

Preface

As I write this preface in June 2018, the ns-3 discrete event network simulator is marking the tenth anniversary of its first public release (ns-3.1). Since that time, over two hundred people have joined together to collaborate on what has become a vital resource for the computer network researcher. Despite this past work, there remains a lot to work on, to keep ns-3 relevant to current telecommunications standards and protocols, and to keep up with external developments in computer languages, large-scale research testbeds, and Linux containerization technologies.

To date, there have been very few books written about ns-3, but I believe that this text and others like it will help to fill the gap that remains between the existing online tutorial and manual and the needs of the ns-3 beginner. ns-3 is a tool with a steep learning curve, because it is written directly in the C++ programming language and not with a more specialized language tailored to network simulation, and without an external integrated development or visualization environment. Users already fluent in other similar tools (like ns-2) or in the Linux kernel programming may have an easier time than most to learn ns-3, but for others, a text such as this will help to provide the background necessary to more quickly master ns-3.

Following a quick start overview of writing and running ns-3 simulations, this book covers ns-3 from the ground up, describing the enhancements to and design patterns of the C++ language used by ns-3, including memory management, management of default configuration values in the simulator, and a dynamic type system. Upon this basis, the models for the various protocol layers in ns-3, starting from the application layer and ending at the wireless layers, are outlined. In the remaining chapters, the reader is introduced to a framework for data collection and statistics, extensions to allow ns-3 simulations to interact with the outside world, and a sampling of related projects.

The author's journey with ns-3 began with his selection, as a student, for the 2012 Google Summer of Code, where he developed a set of schedulers for the ns-3 LTE models. Following this, he was selected for the 2013 European Space Agency's Summer of Code in Space, where he implemented a model for the Bundle Protocol used in the Delay-Tolerant Networking architecture for deep space communications. His expertise has grown since then to the point that he is now a mentor to other ns-3 students, supervising this summer a project aimed at enhancing the usability of ns-3 by developing a simulation execution manager for ns-3. I am hopeful that some readers of this text will follow in his footsteps of contributing code back to the open source project, and then taking the next step of supervising the development work of

others.

The computer networking landscape continues to be fast-moving, with new protocols taking hold in the Internet and with 5G wireless on the horizon, and ns-3 developers are working hard to keep pace. I envision that in the next decade, industrial and educational use of ns-3 will continue to grow the user base. One technical challenge that must be met is to improve the scalability of ns-3 wireless models to larger topologies. Some combination of approaches such as new parallel computing techniques, and more opportunities to abstract away details that are not relevant to the simulation, is likely needed to support the envisioned large wireless simulations of the future. Simulation is highly likely to keep a prominent role in research because of its accessibility, reproducibility, and scalability properties.

Like many open source projects, ns-3 relies on a committed community of volunteers to keep the software moving forward, and I hope that we are about to usher in another decade of ns-3 development with new ideas and new contributors.

Tom Henderson
Affiliate Professor
University of Washington
Executive Director
ns-3 Consortium
June 2018

序

当我在 2018 年 6 月为本书撰写序时，恰逢 ns-3 离散事件网络模拟器首版本（ns-3.1）发布 10 周年。在过去的 10 年中，先后有两百多名开发者联合起来，通力合作，使 ns-3 发展成为计算机网络学者的一项重要研究工具。尽管如此，我们依然有很多工作要做。ns-3 需要与现有的通信标准与协议保持同步，需要跟上编程语言、大规模实验平台和 Linux 容器化技术发展的步伐。

到目前为止，关于 ns-3 的书籍非常稀少。我相信这本著作和其他类似的作品将有助于弥合现有在线文档和 ns-3 初学者需求间的差距。ns-3 是一个具有陡峭学习曲线的工具，其源代码全部使用 C++ 语言编写（一个并非为了网络模拟而定制的编程语言）且没有一个外部集成开发或可视化环境。对于那些已经熟悉类似软件（如 ns-2）或 Linux 内核编程的用户，学习过程可能会相对轻松一些。但对于其他人来说，像本书这样的读物可以提供一个必要的知识基础，让人们更快地掌握 ns-3。

本书首先综述了如何编写和运行 ns-3 模拟脚本，然后从底层讲起，阐述了 ns-3 中 C++ 语言的扩展和设计模式，如内存管理、默认属性和动态类型系统。在此基础上，作者概述了从应用层到无线网络层的各种 ns-3 协议层模型。最后的几个章节分别介绍了数据收集与统计框架、与物理世界交互扩展以及一些相关项目等内容。

本书作者的 ns-3 之旅是从他作为一名学生入选 2012 年谷歌编程之夏（Google Summer of Code）项目开始的。在那里，他为 ns-3 的 LTE 模块开发了一系列的分组分发器。之后作者又入选了 2013 年欧洲航天局的太空编程之夏项目，负责开发深空通信中延迟容忍网络架构的 Bundle 协议模块。随着技术水平的不断提高，作者已成长为 ns-3 组织在谷歌编程之夏项目中的一名导师，指导 2018 年夏天的一项意在扩展 ns-3 可用性的模拟运行管理项目。我希望本书的读者中能够有人追随作者的脚步，向开源项目回馈代码，进而指导其他人的开发工作。

伴随着互联网上新型协议的普及和 5G 网络的问世，ns-3 开发者们正在努力工作，保持与日新月异的计算机网络技术同步发展。展望未来 10 年，我相信 ns-3 在工业和教育中的应用会继续促进用户基数的增长。在这个过程中，如何提高各种大型拓扑环境中 ns-3 无线结点的可扩展性是我们未来必须面对的一项技术挑战。为此，我们可能需要综合使用一些如新型并行计算和抽象非模拟细节等方法。考虑到可行性、可重复性和可扩展性，网络模拟极有可能在研究活动中继续发挥其突出作用。与众多开源项目一样，一个由众多尽责志愿者组成的社区是 ns-3 继续前行的基石。希望我们以新的想法和新的朋友一起迎来 ns-3 发展的下一个 10 年。

Tom Henderson

委任教授

华盛顿大学

执行董事

ns-3 联盟

2018 年 6 月

前 言

这是一本关于开源网络模拟器 ns-3 的书。

网络模拟器可以让人们在一台计算机中搭建出不同规模和类型的网络，从而以一种经济的方式模拟物理网络环境，为人们研究和学习网络协议提供一种方法与平台。编者希望通过本书，将 ns-3 的设计理念和用法系统地介绍给读者。使读者能够有效地利用 ns-3 这一功能强大的网络模拟器，更好地为自己的学习和研究服务。

全书分 3 篇，共 14 章。

第 1 篇 全局之观，共 2 章。这一部分意在让读者建立对 ns-3 的宏观概念，了解 ns-3 的背景、基本原理和用法。

第 2 篇 设计之道，共 5 章。这一部分意在让读者了解 ns-3 的核心设计思想和整体软件架构。熟悉这些设计特性对理解 ns-3 源代码大有裨益。

第 3 篇 模拟之路，共 7 章。前 4 章按照自顶向下的顺序分别讲解了不同网络层协议的基本架构和模拟方法。后 3 章分别介绍了数据收集与统计、与物理网络交互和网络分组 3 个常用功能。

附录部分介绍了 ns-3 网络地址的用法和几个比较成熟的 ns-3 第三方开源项目。

本书的大部分内容均来自 ns-3 用户文档、邮件列表和项目主页等公开网络资源。编者结合多年的 ns-3 开发经验，对这些零散的材料进行了重新整理、总结和再加工，对其已经过时的内容进行了更正。但需要强调的是，本书不是 ns-3 用户文档的替代。对于一些操作细节和非核心功能，读者还需要查阅相关文档资料。

对于 ns-3 初学者，编者建议首先阅读第 1 篇，以熟悉 ns-3 的基本概念和用法。第 2 篇和第 3 篇并无阅读优先级顺序。读者既可以按序阅读，先建立对源代码的理解能力，再深入了解如何模拟某个网络协议；也可以先阅读第 3 篇中自己感兴趣的网络协议章节。如果感到阅读源代码较为吃力，则可以查阅第 2 篇中的相关内容。

本书基于 ns-3.28 版本编写。大部分章节均为 ns-3 核心特性，未来版本产生重大改动的可能性不大。考虑到 ns-3 一年两至三次的版本发布频率，对于一些诸如参数设置、函数名称等细节内容，编者还是建议在阅读时以 ns-3 最新官方文档为准。

本书假设读者具有计算机网络的基础知识并了解 C++ 语言的基本特性。

感谢 ns-3 项目联合创始人、执行董事、华盛顿大学 Tom Henderson 教授为本书作序。感谢褚兴军、申砾和王思轩三位专家为本书撰写推荐语。感谢 ns-3 社区的 Nicola Baldo、Marco Miozzo 和 Tommaso Pecorella 对编者在学习和使用 ns-3 过程中的耐心指

导。感谢 Google 公司和欧洲航天局对 ns-3 社区的资金支持。感谢我的博士生导师，加拿大 UNB 大学宋薇教授在我攻读博士期间对我的包容与帮助。特别感谢我的家人，她们对我的付出与理解，是我坚持写作的基石。

由于编者水平有限，书中难免有错谬之处，还望广大读者批评指正，将不胜感激。
编者的联系邮箱：dizhizhou@hotmail.com。

周迪之
2018 年 5 月

目 录

Preface

序

前言

第1篇 全局之观

第1章 概述.....2	2.4.2 Wi-Fi 无线网络: third 脚本.....22
1.1 什么是 ns-3.....2	2.4.3 Python 脚本.....29
1.2 Hello, world.....3	2.5 再探脚本: 常用技术解析.....31
1.3 ns-3 社区.....4	2.5.1 脚本助理: 助手类.....31
1.4 ns-3 简史.....6	2.5.2 参数输入: 属性变量.....32
1.5 写作习惯.....7	2.5.3 数据输出: trace 变量.....37
第2章 通览.....9	2.5.4 行为控制: 命令行.....43
2.1 下载、编译与运行.....9	2.5.5 计划事件: Schedule 函数.....44
2.2 源代码的目录结构.....12	2.5.6 回调函数: Callback 类.....45
2.3 再谈编译: “./waf configure” 命令.....13	2.5.7 辅助信息: Log 系统.....47
2.4 初识脚本.....15	2.6 网络是如何模拟的.....50
2.4.1 点对点有线网络: first 脚本.....15	2.7 信息入口: 结点类.....53
	2.8 ns-3 与编程之夏.....56

第2篇 设计之道

第3章 对象模型架构.....62	6.1 对象聚合.....76
第4章 Ptr 智能指针.....65	6.2 Object 的创建与获取.....80
4.1 设计原理.....65	第7章 ObjectBase 类.....83
4.2 使用实例.....67	7.1 属性配置.....83
4.3 适用范围.....70	7.1.1 创建属性.....83
第5章 对象模型的基石: 元信息.....71	7.1.2 属性类型.....85
5.1 什么是元信息.....71	7.1.3 ConfigStore.....87
5.2 元信息存储: IidManager 类.....71	7.1.4 全局属性.....88
5.3 元信息管理接口: TypeId 类.....72	7.2 trace 变量.....89
5.4 TypeId 的使用.....73	7.2.1 创建 trace.....89
第6章 Object 类.....76	7.2.2 trace 类型.....90

第3篇 模拟之路

第8章 应用层.....94	8.1 架构与脚本.....94
----------------	------------------

8.2 内置应用: Application	96	10.6.2 路由协议	145
8.2.1 分组产生器	100	第 11 章 链路层与物理层	148
8.2.2 Internet 应用	105	11.1 有线网络: PPP 与 CSMA	148
8.2.3 应用层 trace	106	11.1.1 架构与脚本	148
8.3 自定义应用层协议	106	11.1.2 属性与 trace	149
8.3.1 Socket 原理与使用	107	11.1.3 队列与错误模型	150
8.3.2 Application 类	113	11.2 无线局域网: Wi-Fi	151
第 9 章 传输层	115	11.2.1 架构与脚本	151
9.1 架构与脚本	115	11.2.2 属性与 trace	154
9.2 TCP	117	11.3 蜂窝网: LTE	156
9.2.1 属性与 trace	118	11.3.1 网络架构	156
9.2.2 TCP 分组头结构	119	11.3.2 示例脚本 1: 无线接入网	157
9.2.3 TCP 连接建立与关闭	120	11.3.3 示例脚本 2: 核心网	159
9.2.4 窗口管理	121	11.3.4 属性与 trace	161
9.2.5 拥塞控制	122	11.4 其他无线网络模块简介	162
9.3 UDP	125	第 12 章 数据收集与统计	164
9.3.1 属性与 trace	125	12.1 一个例子	164
9.3.2 UDP 分组头结构	125	12.2 数据收集	165
9.3.3 广播与多播	125	12.3 数据统计	169
9.4 传输层协议开发	126	12.4 局限	171
9.5 示例脚本	128	第 13 章 与物理网络交互	172
9.6 非发行版的传输层协议	128	13.1 TapBridge: 模拟网络+物理 结点	172
第 10 章 网络层	130	13.2 FdNetDevice: 模拟结点+物理 网络	174
10.1 架构与脚本	130	第 14 章 网络分组 Packet 类	177
10.2 脚本配置与 trace	132	14.1 分组结构	177
10.3 IP 分组头结构	134	14.2 分组创建	177
10.4 IP 地址分配	135	14.3 分组头与分组尾	178
10.4.1 手动分配	135	14.4 分段与重组	179
10.4.2 自动分配	137	14.5 标签: Tag 类	180
10.5 路由协议	139	附录	183
10.5.1 列表路由	139	附录 A 网络地址	183
10.5.2 打印路由表	140	附录 B 第三方 ns-3 项目	184
10.5.3 静态路由	141	参考文献	185
10.5.4 全局路由	142		
10.6 网络层协议开发	142		
10.6.1 隧道协议	142		

第1篇 全局之观

题西林壁

苏轼（宋）

横看成岭侧成峰，远近高低各不同。
不识庐山真面目，只缘身在此山中。

第 1 章 概 述

1.1 什么是 ns-3

ns-3 (network simulator) 是一款由 C++ 语言编写的开源项目, 主要运行平台是 GUN/Linux (如 CentOS、Ubuntu、Fedora 等)。虽然 Windows 用户也可使用 Cygwin 或 Visual Studio 运行 ns-3, 但不能使用某些功能 (如与物理网络交互)。ns-3 同时也是一款自由软件。任何组织和个人均可免费下载、使用和修改 ns-3 源代码。

ns-3 主要用于模拟计算机网络。它可以在一台计算机上模拟物理世界中各种类型和规模的网络结构。ns-3 并没有一个图形用户界面。要使用其进行网络模拟, 用户需要经过下载源代码、编译源代码、编写模拟脚本和运行模拟脚本 4 个基本步骤 (在 ns-3 中, 把用于构建虚拟网络的程序叫作模拟脚本)。从这点上讲, ns-3 其实更像是一个程序库。它提供各种用于网络模拟的应用程序接口 (Application Programming Interface, API)。用户在模拟脚本中调用这些 API 来构建自己的虚拟网络结构。目前 ns-3 的模拟脚本可以支持 C++ 和 Python 两种程序语言。

那么 ns-3 是如何模拟出一个物理网络的呢?

这就要从网络的组成说起。大体来讲, 一个计算机网络可以分为以下两个部分。

- 由若干结点和连接这些结点的信道所组成的网络拓扑。
- 运行在结点和信道中的网络协议。

首先, 在 ns-3 模拟的虚拟网络中, 网络拓扑中的结点和信道被抽象成各种 C++ 类。结点和信道的连接操作被抽象成不同 C++ 对象之间的关联。通过这种抽象, 可以在 ns-3 程序中模拟出各种类型的网络拓扑, 如有线网络中的点对点协议 (Point to Point Protocol, PPP) 与总线网络, 无线网络中的无线局域网 (基于 IEEE 802.11 系列标准) 与长期演进技术 (Long-Term Evolution, LTE) 等。这些网络拓扑可以包含几十、几百甚至上千个网络结点。其次, 对于网络协议的模拟, ns-3 使用了一种名叫离散事件 (Discrete Event) 的模拟技术。简单地说, 这种技术就是把物理世界中一个连续的过程抽象成了虚拟世界中的一系列离散的事件。这种技术使得 ns-3 可以非常逼真地模拟物理世界中的各种网络协议, 如应用层的各种分组产生器、传输层的 TCP 和 UDP、网络层的 IPv4 和 IPv6 协议、链路层和物理层的 PPP、IEEE 802.11a/b/g/n 和 LTE 协议等 (见图 1-1)。

应用层	分组产生器、应用层协议（ping 等）
传输层	TCP、UDP
网络层	IPv4、IPv6、ICMP、ICMPv6、路由协议
链路层	无线：LTE、Wi Fi、车载网、个域网、水下通信网
物理层	有线：点对点网络，总线 CSMA 网络 其他：ARP

图 1-1 ns-3 支持的网络协议

只能模拟物理网络还远远不够。为了帮助用户更加便捷地进行网络模拟，ns-3 还提供了一系列的辅助功能。例如，trace 生成功能使用户可以直接通过第三方软件（wireshark、tcpdump）对 ns-3 产生的数据进行分析。移动模块可以为结点自动分配起始位置和移动轨迹。此外，ns-3 所构建的虚拟网络还可以与物理网络环境高度融合。一方面，ns-3 中的虚拟结点可以利用物理网络收发数据；另一方面，物理结点也可利用 ns-3 构建的虚拟信道收发数据包。这样，用户就可以直接使用 ns-3 模拟大规模网络，进而对真实的网络协议代码进行测试，达到节省开发成本的目的。

至此可以看出，ns-3 主要用于模拟网络拓扑和运行其中的网络协议。其关注的是协议行为。其他诸如结点内部的硬件延迟、CPU 使用率等指标是不属于 ns-3 能力范畴的。

自 2008 年 7 月首个版本（ns-3.1）发布以来，ns-3 一直保持着每年 2~3 个版本的发布速度[⊖]。截至本书完稿时，ns-3 已发布至第 29 版本（ns-3.29）。目前，ns-3 已被广泛应用于第五代移动通信（5G）、物联网、软件定义网络、数据中心等计算机网络的前沿研究领域。

1.2 Hello, world

在屏幕上输出“Hello, world”一直是各种编程语言的标准示例程序。ns-3 也不例外。它自带了一个名为“hello-simulator”的 C++模拟脚本，用以向屏幕上输出“Hello Simulator”字样。

先来看一下这个脚本的实际运行效果。下面的第一行代码是在 Linux 命令行中使用 waf 命令执行 hello-simulator 脚本（waf 是 ns-3 项目中最常用的命令。它负责 C++源代码编译和脚本运行）。第二行代码是执行结果，即在屏幕中显示“Hello Simulator”。

```
./waf --run hello-simulator
Hello Simulator
```

[⊖] 2017 年除外。

hello-simulator 脚本的 C++源代码如下。

```
#include "ns3/core-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("HelloSimulator");

int
main (int argc, char *argv[])
{
    NS_LOG_UNCOND ("Hello Simulator");
}
```

可以看出，ns-3 脚本和其他的 C++程序一样，都有一个作为起始入口 main 函数。这里的 NS_LOG_UNCOND 是一个 ns-3 中的宏定义。它的作用就是打印括号中的字符串。此外，整个 ns-3 项目的源代码都受 ns3 名字空间保护。这样可以将 ns-3 项目与非 ns-3 项目隔离，便于与其他项目整合。

这个例子中的 main 函数只有一行代码。在真正的模拟脚本中，构建网络拓扑、配置参数、数据生成等操作均需要在 main 函数中完成。

1.3 ns-3 社区

在自由软件领域，社区（community）的重要性不言而喻。一个成熟的社区环境是自由软件健康发展的必要保障。同时，社区也是开发者、用户工作、学习和交流的平台。ns-3 组织并未对其社区做出任何官方定义。为了便于读者理解，编者结合 ns-3 文档和多年的开发经验，把 ns-3 社区划分为核心社区和延展社区两部分（见图 1-2）。

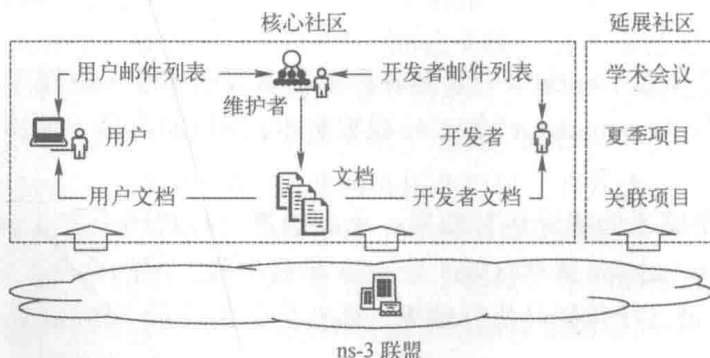


图 1-2 ns-3 社区组成

核心社区包括围绕 ns-3 项目本身的一系列参与方，如 ns-3 用户、开发者和维护者等。

延展社区是 ns-3 项目的外延, 包括与 ns-3 紧密相关的学术会议、面向学生群体的实习项目和其他一些 ns-3 的关联项目。延展社区对 ns-3 的普及和可持续发展至关重要。ns-3 联盟 (ns-3 Consortium) 是由华盛顿大学和法国国家信息与自动化研究所 (INRIA) 于 2012 年 10 月创立的一个组织机构。它负责社区的统一规划、组织协调、资金筹集等一系列长期性战略问题, 为 ns-3 社区的发展提供基础支持。

ns-3 是一个由不同模块 (model) 所组成的项目。每一个模块负责实现一个重要网络功能。例如, LTE 模块负责实现 LTE 网络协议栈。Internet 模块负责实现 TCP/IP 协议栈。Wi-Fi 模块负责实现 IEEE 802.11 系列协议。在核心社区中, 每一个模块都有至少一名维护者。该模块的未来扩展、补丁发布、文档更新及邮件列表答疑等事项均由该维护者负责。这些维护者大多是该模块的主要开发人员。

文档系统和邮件列表系统则是用户、开发者和维护者进行问题讨论、技术交流的主要平台。

首先来看一看 ns-3 文档系统。它主要包含以下几部分内容^[1]。

- ns-3 tutorial: ns-3 的基本概念和使用方法。适合地毯式阅读以了解基本概念。
- ns-3 manual: 深入讲解 ns-3 的架构、设计原理和 ns-3 核心模块中的主要技术细节。适合想进一步了解 ns-3 原理的读者。
- Model library: 由各个模块的开发者编写而成的文档, 分别讲解每个模块的背景知识、设计架构以及使用方法。适合在编写某一模块的脚本或二次开发时使用。
- Doxygen: 从源代码中生成的在线文档。用户可以在 Doxygen 网页中浏览源代码、查询 ns-3 中 C++ 类、函数和变量的定义, 了解类与类之间的继承关系等程序细节信息。Doxygen 是一个很好的工具网站, 在编写模拟脚本和进行扩展开发的过程中非常有用。
- ns-3 wiki 主页: 安装指南、代码提交流程、版本发布等辅助性信息^[2]。ns-3 wiki 主页同时也维护着一些没有整合入主程序的模块与补丁。读者可以从中查找自己所需的相关代码, 避免重复工作。

如果在这些文档中无法找到解决方案, 则读者可以尝试在邮件列表发问。

ns-3 社区成员遍布世界各地。人们主要通过邮件列表沟通交流。这也是很多自由软件项目所常用的方式。ns-3 社区中使用频率最高的邮件列表有两个: Google ns-3 用户组和 ns-3 开发者邮件列表^[3]。前者是用户就 ns-3 安装、脚本编写、模块原理等使用问题进行咨询、讨论的平台。用户组里经常有各个模块的维护者对相关问题进行答疑。后者则是开发者交流意见的主战场。开发者们在此会探讨一些新功能的实现、新版本发布、bug 修改等技术问题。这里需要特别提醒读者的是, 在邮件列表发送邮件之前一定要先做好自己的功课, 确保查阅文档、wiki 主页和搜索邮件列表历史都没有得到满意的答案之后再发问。这也是每一个自由软件人的基本素质体现。

再看看 ns-3 延展社区。

首先是学术会议。与 ns-3 直接相关的学术会议主要有两个: ns-3 研讨会 (ns-3 workshop) 和 EAI Simutools。这两个会议中的论文往往涉及一些 ns-3 的新功能。此外,

ns-3 研讨会每年还会提供培训课程^[4]。这些课程涉及范围很广，从基本概念到新扩展，且往往配有视频和幻灯片，与 ns-3 文档形成很好的互补。

其次是夏季项目。ns-3 的夏季项目主要面向擅长程序设计的学生群体，意在培养未来潜在的开发者。ns-3 夏季项目主要有以下 3 个。

- Google 编程之夏 (Google Summer of Code): Google 公司主办的年度带薪程序设计项目^[5]。
- 欧洲航天局太空编程之夏 (Summer of Code in Space): 欧洲航天局举办的面向航天航空领域的年度带薪程序设计项目^[6]。
- 夏季指导项目 (Mentored Summer Projects): ns-3 组织自己推出的不带薪指导项目。主要针对没有入选前两个编程之夏项目，但仍有意愿为 ns-3 社区贡献代码的学生^[7]。

参与这 3 个项目的学生会在 ns-3 资深开发人员的指导下完成指定课题。今天很多 ns-3 模块最初就是从这些夏季编程项目衍生而来的。编程之夏对 ns-3 的后续发展帮助很大。对此感兴趣的朋友可以跟踪编者维护的“编程之夏”微信公众号 (微信号: codesummer, 二维码见本书勒口)。编者在 2.9 节还会对这一项目做更为详细的介绍。

最后是关联项目。ns-3 关联项目分为两类: 由 ns-3 组织维护和由外部机构维护。由 ns-3 组织维护的关联项目有以下几个。

- Direct Code Execution: 一个能够让 Linux 协议直接运行在 ns-3 中的软件框架。
- Netanim: 一款能够以动画显示模拟过程的软件。
- Bake: 一个方便用户一次性安装多个 ns-3 关联项目的工具程序。

此外, 也有很多 ns-3 项目由外部机构开发^[8]。比较著名的有西班牙 CTTC 研究所开发的 LTE 模拟器^[9]和芬兰 Magister Solution 公司开发的卫星通信模拟器 SNS3^[10]等。由于 ns-3 主程序库有严格的测试程序, 因此这些项目目前尚未或仅部分并入 ns-3 发行版。但其均与 ns-3 社区保持有密切联系。附录 B 列举了几个开发状态比较活跃、且符合目前计算机网络研究热点的第三方 ns-3 项目。

1.4 ns-3 简史

ns 系列共有 3 款模拟器: ns-1、ns-2 和 ns-3。ns-2 向前兼容 ns-1。而 ns-3 则是一款全新的模拟器, 不向前兼容。尽管如此, ns-3 和其前身依然有着深厚的历史渊源。

ns-3 的历史最早可追溯至 1988 年发布的 REAL 模拟器^[11]。该模拟器由当时还在加州大学伯克利分校攻读博士的 Srinivasan Keshav 编写。20 世纪 90 年代中期, REAL 被拓展成为 ns-1。1996 年, ns-1 的 tcl 代码被 OTcl 语言代替, 进而演化成 ns-2 模拟器^[12]。此后的 10 年间, 计算机网络技术突飞猛进, ns-2 也迎来了黄金的发展期。先后有美国国防高级研究计划局 (DARPA)、南加州大学信息科学研究所 (USC/ISI) 等多所大型研究机构参与到 ns-2 项目的开发中来。

但是, 随着时间的推移, ns-2 软件架构中的一些问题也逐渐暴露出来, 如其缺乏对真