网络模拟软件 NS2 与 OPNET 的剖析比较①

王　波　周志伟 (重庆大学 计算机学院建筑智能化与城市数字化研究室 重庆 400044)

摘　要：NS2 和 OPNET 是目前主流的网络模拟软件，为深入地剖析比较它们的特点及差别以供学习和选用时参考，从软件技术、使用方法和引擎性能三个方面进行了详细对比分析。在软件技术方面分别从体系结构、支持的网络技术、网络设备模型库和软件可扩展性上进行分析比较；在模拟器引擎性能方面，通过在相同模拟条件下，选取内存消耗和 CPU 运算时间两个指标，针对不同规模的抽象网络，分别对 OPNET 和 NS2 的引擎性能进行了对比实验。结果表明，OPNET 具有商业软件的优势，在用户友好性、模拟速度、内存消耗等方面优于 NS2，而 NS2 则在开源和易扩展性方面优于 OPNET，适合于模拟中小型网络。

关键词：网络模拟；网络模拟测试；NS2；OPNET

Comparative Analysis on Network Simulation Software NS2 and OPNET

WANG Bo, ZHOU Zhi-Wei (Building Intelligentization and City Digitalization Lab, College of Computer Science of Chongqing University, Chongqing 400044, China)

Abstract: This paper presents the features and differences of NS2 and OPNET by comparing them in the aspects of software technique, operation method and engineer performance. As to the software technique, it includes four points for comparison: system architecture, supported network technique, network equipment model bank and software scalability. To compare the performance of the simulator engine, an abstract network which is not real was established and simulated by these two simulators separately under identical conditions. Meanwhile, two performance indices (memory consumption and computation time) were measured, recorded and analyzed. The comparison result indicates that OPENT as a commercial software is better than NS2 in user-friendliness, simulation speed and memory consumption while NS2 is superior to OPNET in the aspect of open-source and scalability and well suited for small scale network simulation.

Keywords: network simulation; network simulation test; NS2; OPNET

1　引言

在计算机网络技术日新月异、高速发展的今天，一方面各种网络应用种类不断增加，新的协议、算法层出不穷，验证、测试和分析这些新应用、新协议和新算法的课题亟需解决，另一方面规划、设计新网络或者分析、测试和更新现有网络需要更加便捷、快速、准确的方法，各式各样的网络模拟工具应运而生，目前常见的主要有：NS(Network Simulator)与 OPNET (Optimized Network Engineering Tool) 以及 OMNeT++和 SSFNet 等，其中在教育、科研、商业等领域使用最广泛的是 NS2 与 OPNET。

NS 起源于 1986 年的 Real Network Simulator 项目，1995 年在施乐公司(Xerox)的支持下加入 VINT 项目[1]。在不断吸收世界各地研究者的成果过程中，

① 基金项目:国家水体污染控制与治理科技重大专项(2008ZX07315-001)
收稿时间:2009-09-24;收到修改稿时间:2009-11-08

NS从原始版本进化到较成熟版本NS2，最新版本NS3也已经面世。虽然NS3版本较NS2新，但它是一个全新的模拟器，不向下兼容前一版的NS2，不支持NS2的API[2]，在诸多方面与NS2相差较大。由于目前NS3在无线网络模拟等方面功能欠缺，且用户数与NS2相差悬殊，因此本文选择NS2作为研究对象。

OPNET是与NS2类似的网络模拟软件，1986年由麻省理工大学的两个博士创建。随后他们从网络模拟中发现了蕴藏的巨大的商业价值，于1987年建立了商业化的OPNET。目前用户范围广泛，包括企业、网络运营商、仪器配备厂商，以及军事、教育、银行、保险等领域。至今OPNET已经发行10.0以上版本，其产品线主要有Modeler、IT Guru、SP Guru和WDM Guru等。每种产品所针对的用户群不同，功能也不尽相同。其中OPNET Modeler几乎包含其他产品的功能，其主旨在于为网络技术人员提供一个网络技术和产品的模拟平台，以帮助其设计和分析网络拓扑和通信协议[3]，在功能和用途上与NS2相似，因此本文选择OPNET Modeler作为与NS2进行比较的对象(下文简称OPNET)。

面对NS2与OPNET这两种优秀的网络模拟软件，初次接触的网络研究人员和网络规划设计工程师容易面临选择困惑，且目前尚未检索到针对这两种软件的对比分析文章。本文结合在教改项目“计算机网络教学实验模拟平台[4]”中研究、使用NS2和OPNET的认识与经验，选取软件技术(包括体系结构、支持的网络技术、网络设备模型库和软件可扩展性)、实际的应用步骤和核心模拟引擎性能等方面，详细、深入地对NS2与OPNET进行剖析与对比。

2 NS2与OPNET的技术对比分析

2.1 体系结构

NS2是一种可扩展、可重用、基于离散事件驱动、面向对象、采用分裂对象模型的网络仿真器。它使用两种编程语言，一种是由麻省理工学院开发的面向对象工具命令语言OTcl，另一种是C++。采用这两种语言是因为C++是高效的编译执行语言，能够高效、快速地完成具体协议模拟和实现报头(Packet Header)、数据等的处理工作，同样也能够应用各种算法在大量数据集合上进行操作。但是用C++运行模拟环境、寻找和修复Bug，以及重新编译和执行的时间都较长。然而OTcl却能有效地弥补这一缺陷，在不重新编译的情况下用OTcl编写的网络模拟脚本能够快速地设置和改变网络组件和环境的具体参数、模拟网络场景(scenario)，并且发现、修改和修复程序中的Bug也较为便捷[1]。

NS2是通过TclCL将C++和OTcl两种语言的对象和变量关联起来的，C++的类和对象为编译类和编译对象，而OTcl的类和对象为解释类和解释对象。TclCL是在OTcl基础上的封装。NS2的整体架构如图1所示。

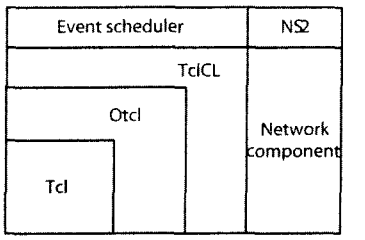

图1 NS2分裂对象模型架构图

与NS2相似，OPNET也是基于离散事件驱动模拟机理，采用面向对象技术的网络模拟工具，但与NS2的分裂对象模型架构不同，OPNET采用的是一种层次架构。OPNET模型层次结构如图2所示，可以分为三层，从低到高依次为：进程(process)模型层，节点链路(node and link)模型层和网络(network)模型层。采用层次的模型结构为不同层次间模型的重用提供了便利。

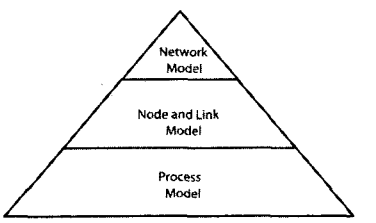

图2 OPNET层次模型架构图

三层模型中最底层的进程模型主要由强迫或非强迫状态和有条件或无条件转移线组成，用来刻画节点模型里的处理机以及队列模型的行为，并且以由有限状态机(Finite State Machine)的状态转移图(STD)、标准的C/C++语言以及OPNET提供的核心函数(Kernel Procedure)集合而成的Proto-C语言作为建模语言。

节点模型基于进程模型，主要由各种模块即由处

理器、队列、各种类型的接收器、发送器，以及数据包流线、统计数值线和逻辑关联线组成。这些模块大致可分为两种，一是具有内建参数的模块，例如各种类型的接受器、发送器，二是高度可编程模块，例如处理器和队列模块，这两种模块均可使用用户自定义的进程模型来设定模块行为。

网络模型则是用来建立一个具体的物理网络拓扑模型，利用模型库中已有的各种通信实体模型可以方便地构建网络拓扑，并可对设备的属性进行设置。同时，网络模型中也可以根据用户自身需要将节点建模中建立起来的网络设备互联成网络。

2.2 支持的网络技术

OPNET 和 NS2 均支持许多无线和有线网络中使用的协议、路由算法，以及各种应用技术。而与 NS2 不同，OPNET 除了支持这些协议、算法、应用外还对众多网络设备供应商的各种型号的产品建有具体的产品模型库。

NS2 广泛支持各种网络协议和算法，本文所使用的 NS 2.28 版支持的核心网络技术[5,6]见表 1，OPNET8.1 支持的网络技术[5]见表 2。

表 1　NS2 支持的网络协议

Application level Protocols	HTTP, Telnet, FTP, CBR
Transport level Protocols	TCP, UDP, RTP, SRM
MAC Protocols	802.3, 802.11, TDMA
Scheduling disciplines	Drop Tail, FIFO, RED, CBQ, WFQ, SFQ, DRR FQ,
Traffic Characterization	Poisson, Exponential, Pareto, CBR
Wireless Networking	Distance Vector and Link State, Ad-Hoc Routing protocols, Multi-hop routing protocols(DSR, AODV, TORA, DSDV)

2.3 网络设备模型库

NS2 并未提供路由器、交换机等具体网络设备模型库，建模用的网络设备模型均需用户手动编写 OTcl 脚本建立且参数配置复杂。此外，用户对任何原有参数的改变都要在多个源代码文件中修改且需重新编译生成 NS2，可谓牵一发而动全身。完成此工作需要用户有一定的编程基础且对 NS2 本身构架有一定了解。

表 2　OPNET 支持的网络协议与技术

Application level Protocols	HTTP, Telnet,DNS,POP3,SMTP,FTP
Transport level Protocols	TCP, UDP, RTP, SRM, NCP
Data Link Technology	ATM, FDDI, FR, LANE, LAPB, STB, SRP, SNA, TR, VLAN, LAPB, STB, SRP, SNA, TR, X.25
Scheduling disciplines	Drop Tail, FIFO, RED, CBQ, WFQ, SFQ, DRR,FQ, RIP, OSPF, BGP, IGRP, EIGRP, IS-IS
Traffic Characterization	Poisson, Exponential , Pareto, CBR , binomial, weibull, triangular, Rayleigh, pareto, normal
Wireless Networking	Distance Vector and Link State, Ad-Hoc Routing protocols, Multi-hop routing protocols(DSR, AODV, TORA, DSDV)
Equipment donors	3Com, Avici Systems, Cabletron, Cisco, Extreme Networks, Foundry, Networks, Hewlett-Packard, Juniper, Lucent &Ascend, Marconi Systems (Fore Sys2tems), Newbridge , Nortel &Bay Networks

OPNET 由于是一个商业软件，为了满足不同层次人群的需要，它为用户提供了一个详尽的模型库。该模型库几乎囊括了大部分网络设备供应商生产的路由器、交换机、服务器、客户机等设备模型，使用户能根据现实中准备采用或使用中的网络设备来选择生产厂商、型号相同的模型，从而保证了模拟结果的准确性，同时还为用户选择具体的设备提供参考。除使用已有模型之外，用户还可以在原有模型的基础上做出小的修改实现快速新建模型，以节省建模时间。

2.4 可扩展性

NS2 允许用户对其功能进行扩充以研究新的网络协议和算法。NS2 原有协议、算法的源代码均以层次化的方式组织在 NS2 的安装目录下，同时该目录下 makefile 文件包含了每次编译、生成新的 NS.exe 文

件时必须的编译文件信息。对于扩充的文件，可以通过继承等方法部分利用 NS2 原有的一些类，因此需要用户仔细研究原有代码。扩充新协议的具体步骤如下[7]：

(1) 定义或继承 C++协议类

(2) 编写该类成员函数和协议算法

(3) 定义 TCL 相关的类和变量

(4) 把 C++代码绑定到 TCL

(5) 修改 makefile 文件，重新编译生成 NS。

OPNET 也允许扩展其协议库，用户添加新的协议需要分别在各个模型层做出相应修改。

3 NS2与OPNET的应用对比分析

虽然 NS2 和 OPNET 的在实施模拟的细节上存在差异，但完成实际模拟任务的总体步骤是相同的。大体上说，完成一次模拟的生命周期可分为三个部分：建立模拟场景，运行模拟程序，对模拟结果数据进行分析。下面详细比较在实际应用方面的不同之处。

在只使用NS2内建的模型或协议库而不添加任何模型或协议的前提下，用 NS2 进行模拟的主要过程如图 3 所示。首先用 OTcl 语言编写模拟脚本，构建网络拓扑结构，配置各项网络参数，例如链路延迟、带宽以及节点协议类型等，设置整个模拟进程运行时间和产生结果文件存放路径等模拟器参数。然后对编写好的脚本用NS2模拟器解释执行，结果得到两类文件，一类是 nam 后缀文件，用于 NAM 对网络模拟过程动态再现演示，另一类 tr 后缀文件，用 NS2 规范格式详细记录了整个模拟过程中每一个发生事件的资料，包括事件的性质、源节点、目的节点、数据包类型和大小、源节点和目的节点的地址，以及流标号等信息，利用这个文件可以编写处理代码计算出各项网络性能指标。由于使用 C 或 C++处理这些数据，效率不高且所写的程序也会很复杂，所以 NS2 使用能便利快速处理这些数据的 AWK 语言进行分析处理，对 trace 文件用 AWK 程序处理之后得到的数据，可利用 GNUplot，XGraph 等绘图工具以图形化的形式显示。

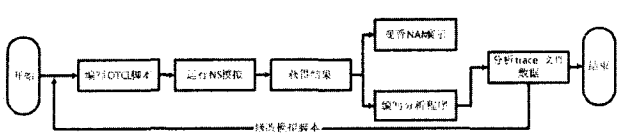

图 3　NS2 模拟流程

在 OPNET 的环境下建立该网络的模型必须按照图 2 所示的层次模型对每层依次建模，整个流程如图 4 所示。首先是最重要的进程层，它决定了数据包在节点中处理的程序。在进程模型定义完毕以后，便可以使用这些进程模型建立节点模型。使用节点模型可以建立网络拓扑，然后在运行模拟场景之前需要选定将要收集的统计量(statistic)，最后，与 NS2 有所不同，OPNET 在模拟程序运行完成之时便已经将统计量的值计算完毕，并以向量的形式保存起来，且能直接用 OPNET 自带的绘图工具显示出来，并不需要用户编程处理跟踪数据，这一点上 OPNET 要比 NS2 方便得多。还有值得一提的是，OPENT 提供了顺序仿真(simulation sequence)的功能，对于同一个场景(scenario)的同一个参数可用不同的值顺序运行模拟程序，方便比较某一参数对同一网络场景的影响程度。而相较之下，完成此功能 NS2 却要复杂的多，每改一次参数必须返回将整个模拟流程重复一遍，花费大量时间和精力在重复劳动上。

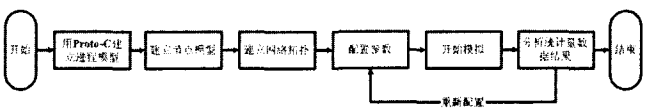

图 4　OPNET 模拟流程

4 NS2与OPNET引擎性能比较

NS2 与 OPNET 在具体网络协议的模拟性能上已有相关研究，并得出一些结论[8,9]，但这些结论只是在某种限定条件下成立，不能够客观反映模拟器性能本质，具有一定的局限性。为了突破这一局限，这里选取模拟器的核心引擎作为研究对象，屏蔽差异，比较相同条件下完成模拟任务时的内存和 CPU 资源消耗情况，其数据能够较客观地反映出两种模拟器的性能差异。

4.1 建立模拟场景

为了模拟器引擎性能对比结果的准确性和客观性，设计了一种抽象网络作为模拟对象。

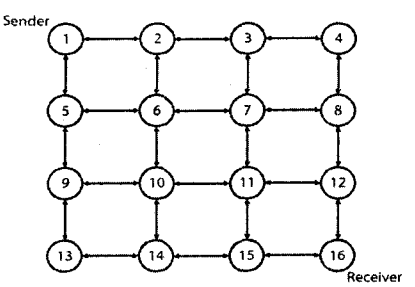

图 5　4×4 规模的抽象网络

如图 5 所示，该网络由 $n \times n(n \in N+, n \in [2, 32])$ 个节点互联成方阵拓扑，其中 1 号节点作为发送端产生数据包并发送给其相邻节点，然后由各个节点转送给各自的相邻节点，最后到达与发送端同在一条对角线上的接收端。为了使网络尽量简单和在两个模拟器中使用的模拟建模脚本尽可能保持一致，该网络不使用任何模拟器提供的网络协议和技术。数据包从发送端经过众多中间节点传输到接收端，除了人为在链路上设置数据传送中的丢包率外，碰撞和拥塞也都不予考虑，这样设计的网络与现实网络有很大差别，这也是其被称为抽象网络的原因。

将整个网络每条链路的丢包率固定设置为 0.1，整个模拟过程历时 100s，在准备时间 10s 后，发送端节点 1 开始发送数据 80s。参数设置完毕后将网络规模从 2×2 逐步扩大到 32×32，分别对每一种网络大小用 NS2 和 OPNET 进行模拟并记录内存消耗和 CPU 运算时间数据，最后将这些数据汇总并用图形化的方式显示出来，从而分析比较两个模拟软件的性能差别。

在两种模拟器中使用的模拟脚本除了做出必要的细节修改以外，在设计上大体相同，主要包括节点(Node)类、链路(Link)类、数据包(Packet)类。数据包类代表了网络中传送的数据包指明了唯一 ID 号；节点类负责产生、发送和接收数据包，并记录下每一个接收到的数据包的 ID。由于一个节点可能从不同的链接接收到同一个数据包，因此在每个节点中设计了一个链表用以记录该节点接收到的所有的数据包，如果发现接收到的数据包已经存在于列表中则立即丢弃；链路类存储了连接其上的两端的节点并能够发送数据包，每接收到一个数据包便按照设置的丢弃概率将其丢弃然后传送给下一个节点。

4.2 模拟性能对比

用于运行模拟的计算机配置为 Intel pentium4 3.00GHz CPU 和 704MB 内存，NS2 版本为 2.28，OPNET 版本为 OPNET Modeler 10.0。图 6 和图 7 分别是在丢包率为 0.1 情况下模拟不同规模网络时的内存消耗和 CPU 机时数据图。由图可知，虽然理论上 NS2 可以模拟 16,000 个节点的网络[10]，但是根据图中趋势当网络节点数目到达或者超过 1000 个节点时，对资源消耗过大，需要的时间太长，所以用其模拟大型网络效果不理想。OPNET 以其性能良好的模拟引擎和高效的资源利用效率，在模拟大规模网络时相对 NS2 较为优越，在内存消耗和 CPU 计算时间上均显著少于 NS2。

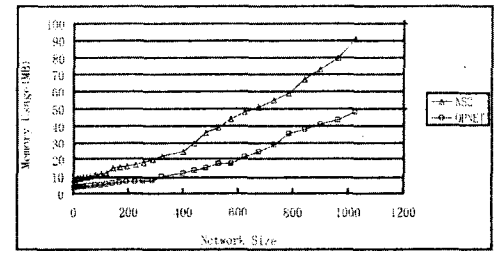

图 6　内存占用量(丢包率=0.1)

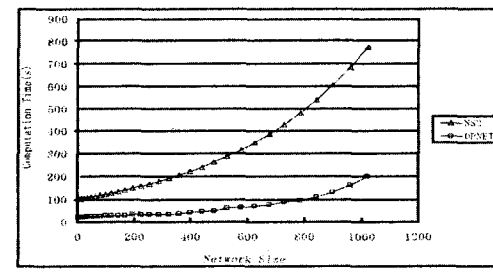

图 7　CPU 计算时间(丢包率=0.1)

4.3 对比结论

通过以上的对比，NS2 与 OPNET 在软件技术和应用方面的差别已很清楚，但还存在一些其他差别，表 3 是 NS2 与 OPNET 的一个全面对比。

表 3　NS2 与 OPNET 对比结论

比较点	NS2	OPNET
软件图形用户界面(GUI)	无，只有 NAM(Network Animator) 有用户界面	有，且图形工具丰富
获取方式	免费	需购买且价格昂贵
好学易用	无正式文档且无 GUI，学习使用上存在困难	文档详细且出色的 GUI，好学易用
面向对象设计	是	是
建模方式	分裂对象模型	层次模型
模拟器效率	不支持并行模拟，网络规模增大时模拟效率不高	支持并行模拟
扩展性	非层次结构，扩展方便	逐层扩展
动态演示功能	有，NAM	有
建模语言	OTcl	在进程模型中使用特有的 Proto-C
数据绘图工具	无内建工具，借助 Xgraph，GNUplot 等其他工具	有内建工具，且功能强大具有多种图形显示模式

5 结语

本文主要从软件的技术、应用和核心引擎性能三个方面，对网络模拟工具 NS2 与 OPNET 进行了详细的对比分析，有以下结论：

(1) 在技术方面：两款软件模拟机理相同，但是软件架构存在较大差异，NS2 采用分裂对象模型而 OPNET 则是层次模型架构。作为商业软件，OPNET 模型库较 NS2 要丰富许多，但是 NS2 开放的特质和强大的用户支持使其技术发展潜力胜于 OPNET。

(2) 在应用方面：虽然完成模拟工作的总体步骤相同，但是实施细节上繁简不一。NS2 缺乏用于网络建模的 GUI， NS2 中建立网络模型需使用脚本语言 OTcl，处理模拟结果也需编码实现，用 NS2 作为模拟工具代码量较大，用户的学习周期长且使用中代码工作量繁重。相较之下，OPNET 图形界面丰富，网络模型全面，API 函数库提了大部分常用函数，建模层次分明简便，较易上手，且建模所需手动编写的代码量少。

(3) 在模拟器引擎性能上：当网络规模较小时，NS2 与 OPNET 在同等情况下所需内存 CPU 资源相当，但随着网络中节点数逐渐增加，OPNET 引擎并行模拟的能力发挥效力，其模拟性能的优越性逐渐显现。结果数据显示出 NS2 在模拟大型网络方面的性能差强人意。

总体上说，虽然 NS2 存在用户友好性差，性能不够理想等方面的缺陷，但以其开源、易于扩展等特质在教学、学术领域有一定的声望。而 OPNET 作为商业软件，功能强大，性能卓越，虽价格昂贵但仍具有强大的吸引力。

参考文献

1 徐雷鸣,庞博,等. NS 与网络模拟,北京:人民邮电出版社, 2003.

2 NS-3 Network Simulator. [2009-9-21]http://www.nsnam.org/.

3 陈敏. OPNET 网络仿真.北京:清华大学出版社. 2004. 6－13.

4 王波,孙燚,周志伟.计算机网络实验综合模拟平台的研发.计算机教育, 2009,(3):85-88.

5 侯宗浩,王秉康,等.网络仿真的研究.计算机仿真, 2003,20(10):89－91.

6 The Network Simulatorns-2: ValidationTests. [2009-9-2].http:// www.isi.edu/nsnam/ns/ns-tests. html.

7 王晓曦,王秀丽. NS2 网络仿真器功能扩展方法及实现.小型微型计算机系统, 2004,25(6):1009－1014.

8 G. Flores-Lucio. Opnet-modeler and NS-2: Comparing the accuracy of network simulators for packet-level analysis using a network testbed. Proc. Int. Conf. Simul., Model., Optim. 2003. 700－707.

9 Garrido PP, Malumbres MP. NS-2 vs. OPNET: a comparative study of the IEEE 802.11e technology on MANET environments. Proc. of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops. 2008.

10 Kroller A. Shawn:A new approach to simulating wireless sensor networks. Proc. Design, Analysis, and Simulation of Distributed Systems. 2005. 117－124.

(上接第 20 页)

pages/ research/ p2p2004.php

7 Wei HY, Lin YD. Assessing and Improving TCP Rate Shaping over Edge Gateways. IEEE Communications Surveys and Tutorials, 2005.4:240－248.

8 Sun YS, Lee C, Berry R, Haddad AH. An Application of the Control Theoretic Modeling for a Scalable TCP ACK Pacer. Proc. of the 2004 American Control Conference, 2004,(5):40－48.

9 CaserriC, Meo M. A new approach to model the stationary behavior of TCP connection. IEEE INFOCOM, 2000,7(5):143－149.

10 Sivaraman V, Fabio M, Gerla CM. Traffic Shaping for End-to-End Delay Guarantees with EDF Scheduling. IEEE/ACM Transactions. 2000. 10－18.

11 Dimitrios Stiliadiss, Anujan Varma.Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms. IEEE/ACM Transactions on Networking, 1998,(8):432－438.

12 Wehrle L, Paehlke F. The Linux Network Architecture: Design and Implementation of Network Protocols in the Linux kernel. Pearson-Prentice Hall, 2005.