Unlocking the Power of OPNET Modeler

For fast, easy modeling, this practical guide provides all the essential information you need to know. A wide range of topics is covered, including custom protocols, programming in C++, External Model Access (EMA) modeling, and co-simulation with external systems, giving you the guidance not provided in the OPNET documentation. A set of high-level wrapper APIs is also included to simplify programming custom OPNET models, whether you are a newcomer to OPNET or an experienced user needing to model efficiently. From the basic to the advanced, you’ll find topics are easy to follow with theory kept to a minimum, many practical tips and answers to frequently asked questions spread throughout the book, and numerous step-by-step case studies and real-world network scenarios included.

Zheng Lu received his Ph.D. from the University of Essex, after which he stayed on to research optical networks and wireless sensor networks. He is experienced in modeling network protocols and has many years of experience using OPNET Modeler in his research and laboratory demonstrations.

Hongji Yang is currently a Professor at the Software Technology Research Laboratory, De Montfort University. He received his Ph.D. from Durham University in 1994 and was a main contributor to the Distributed Computer Networks project sponsored by the Chinese Ministry of Education, 1982–1986.
Unlocking the Power
of OPNET Modeler

ZHENG LU
HONGJI YANG
# Contents

*Preface*  xi

*List of abbreviations*  xiii

**Part I Preparation for OPNET Modeling**  1

1 **Introduction**  3
   1.1 Network modeling and simulation  3
   1.2 Introduction to OPNET  4
   1.3 OPNET Modeler  5
   1.4 Summary  6
   1.5 Theoretical background  6
      1.5.1 Simulation and principles of simulator  6
      1.5.2 Hybrid simulation  9

2 **Installation of OPNET Modeler and setting up environments**  11
   2.1 System requirements for using OPNET Modeler  11
   2.2 Installation on Windows  11
      2.2.1 Installation of OPNET Modeler  12
      2.2.2 Installation and configuration of Microsoft Visual C++  15
      2.2.3 OPNET Modeler preferences for C/C++ compiler  17
      2.2.4 Licensing  19
   2.3 Installation on Linux  20
      2.3.1 Installation of OPNET Modeler  20
      2.3.2 Installation and configuration of GCC compiler  21
      2.3.3 OPNET Modeler preferences for GCC compiler  21
      2.3.4 Licensing  22
   2.4 Theoretical background  23
      2.4.1 Compilation and linking options  23
      2.4.2 Simulation models compilation and linking  23
### Contents

#### Part I OPNET Modeler user interface

3 OPNET Modeler user interface 24

3.1 Project management 24
3.2 Modeler preferences 26
3.3 OPNET editors 29
  3.3.1 Project Editor 29
  3.3.2 Node Editor 31
  3.3.3 Process Editor 31
  3.3.4 Link Editor 32
  3.3.5 Packet Format Editor 32
  3.3.6 ICI Editor 34
  3.3.7 PDF Editor 35
  3.3.8 Probe Editor 35
3.4 Simulation Results Browser 37
3.5 Animation Viewer 37
3.6 Using OPNET documentation 39

#### Part II Modeling Custom Networks and Protocols

4 OPNET programming interfaces 43

4.1 Introduction to OPNET programming 43
4.2 OPNET API categorization 44
4.3 Kernel APIs/Kernel Procedures (KPs) 45
  4.3.1 Distribution Package 46
  4.3.2 Packet Package 49
  4.3.3 Queue Package and Subqueue Package 51
  4.3.4 Statistic Package 51
  4.3.5 Segmentation and reassembly package 52
  4.3.6 Topology package 52
  4.3.7 Programming Support APIs 54
4.4 Theoretical background 54
  4.4.1 Proto-C specifications 54
  4.4.2 Process model and external model access (EMA) program 56
  4.4.3 OPNET Modeler model programming external interfaces: co-simulation, external tool support (ETS) and OPNET Development Kit (ODK) 56

5 Creating and simulating custom models using OPNET APIs 58

5.1 General procedure for creating and simulating custom models 58
5.2 Custom models 59
  5.2.1 Case 1 59
  5.2.2 Case 2 68
  5.2.3 Case 3 70
### Contents

5.2.4 Case 4 74
5.2.5 Case 5 79
5.2.6 Case 6 83
5.2.7 Case 7 95
5.3 Model optimization and validation 96

6 High-level wrapper APIs 100

6.1 Why and how to use wrapper APIs 100
6.2 Wrapper APIs provided with the book 101
   6.2.1 Geo_Topo wrapper APIs 102
   6.2.2 Routing wrapper APIs 104
   6.2.3 Flow wrapper APIs 106
6.3 How to write your own wrapper API 107

7 Modeling with high-level wrapper APIs 110

7.1 Revisit of previous case 110
7.2 Creating connection-oriented communications 112
   7.2.1 Single flow 114
   7.2.2 Trunk of flows 119

Part III Modeling and Modifying Standard Networks and Protocols 123

8 Modeling wired networks with standard models 125

8.1 Client/server structure 125
   8.1.1 Creating a network model 125
   8.1.2 Task, application, and profile configurations 127
   8.1.3 Choosing and viewing statistic results 131
8.2 Local area network 132
8.3 Wide area IP network 132
8.4 Automatic network deployment 134
8.5 Summary 135

9 Modeling wireless networks with standard models 137

9.1 Basics of wireless modeling 137
9.2 Wireless local area networks (WLANs) 138
   9.2.1 Communication within WLANs 138
9.3 Communication between WLANs 140
9.4 Wireless mobile networks 143
   9.4.1 Movement via trajectories 143
   9.4.2 Facilities for random mobility 146
   9.4.3 Movement via programming interfaces 148
9.5 Automatic network deployment 148
<table>
<thead>
<tr>
<th>Part</th>
<th>Title</th>
<th>Sections</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Modifying standard models</td>
<td>10.1 Introduction, 10.2 Case study</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td><strong>Part IV OPNET Modeling Facilities</strong></td>
<td></td>
<td>165</td>
</tr>
<tr>
<td>11</td>
<td>Debugging simulation</td>
<td>11.1 Debugging facilities in OPNET Modeler, 11.1.1 Prerequisites for debugging, 11.1.2 Preparing simulation scenario, 11.1.3 Debugging with ODB, 11.1.4 Debugging with CDB/GDB, 11.1.5 Debugging with Microsoft Visual C++ Debugger, 11.1.6 Debugging with animation</td>
<td>167</td>
</tr>
<tr>
<td>12</td>
<td>OPNET programming in C++</td>
<td>12.1 Proto-C, C, and C++: language and library differences, 12.2 Memory management differences between Proto-C APIs and C/C++ standard library functions, 12.3 Proto-C data structures and algorithms packages, C++ standard template libraries (STL) and Boost C++ libraries, 12.4 Environment configurations for C++ programming in OPNET, 12.5 Case study on programming OPNET models in C++</td>
<td>182</td>
</tr>
<tr>
<td>13</td>
<td>Traffic in OPNET simulation</td>
<td>13.1 Introduction, 13.2 Explicit traffic, 13.2.1 Explicit traffic based on application, 13.2.2 Explicit traffic based on traffic generation parameters, 13.2.3 Explicit self-similar traffic based on raw packet generator (RPG) model, 13.3 Background traffic and hybrid simulation, 13.3.1 Background traffic based on baseline load, 13.3.2 Background traffic based on traffic flow</td>
<td>194</td>
</tr>
<tr>
<td>14</td>
<td>External model access (EMA)</td>
<td>14.1 What EMA is and reasons to use it, 14.2 EMA case study</td>
<td>207</td>
</tr>
<tr>
<td>15</td>
<td>OPNET co-simulation with third-party programs</td>
<td>15.1 Co-simulation with external programs</td>
<td>215</td>
</tr>
</tbody>
</table>
15.1.1 Introduction 215
15.1.2 Co-simulation with an external C program 216
15.1.3 Creating simulation models 217
15.1.4 Creating an external C co-simulation controller program 221
15.1.5 Running co-simulation 224
15.1.6 Co-simulation with other systems 225

15.2 Co-simulation with MATLAB 225
15.2.1 Setup of environment variables 226
15.2.2 Modifying OPNET models and external code 226

16 Model authoring and security 232
16.1 Introduction 232
16.2 Protecting a model 232
16.3 Making a demo model 234
16.4 Licensing a model 234

References 236
Index 237
Preface

Network simulation is an important methodology in network research fields and OPNET Modeler is a very useful tool for network modeling and simulation. OPNET Modeler is generally used by researchers, protocol designers, university teachers and students in the fields of electronic engineering, computer science, management information systems, and related disciplines. The friendly design of its graphical user interface (GUI) makes it nice and easy to start with. However, the complexity of OPNET Modeler and lack of useful support material make it difficult for many users to fully make use of its benefits. OPNET Modeler has its documentation covering many aspects on using the modeler. However, it covers too many aspects in parallel form rather than a step-forward form, making users unable to decide where to start and causing them to lose focus.

This book is an effort to partially fill this gap and should be useful for courses on network simulation and OPNET modeling for university students, as well as for the researchers on this topic. The book covers a wide range of knowledge from basic topics to advanced topics. All case studies in the book are step-by-step and progressive. Relevant files and sources can be downloaded from the publisher’s website. A set of high-level wrapper APIs are provided to help even new users to write complex models, and experienced users to write large, complex models efficiently. Question-and-answer pairs are spread over the chapters to answer the most common questions users may experience in practice.

The book is composed of four parts. Part I: Preparation for OPNET Modeling introduces OPNET and OPNET Modeler. It leads the reader through the required basics on using OPNET Modeler and provides familiarization with OPNET Modeler user interfaces. Part II: Modeling Custom Networks and Protocols first teaches the reader how to create custom models by directly using OPNET API packages. It then introduces a high-level wrapper API package and demonstrates how to model systems easily using these high-level wrapper API packages instead. Part III: Modeling and Modifying Standard Networks and Protocols teaches the reader how to model networks and protocols based on existing standard OPNET modules and how to modify existing standard models in order to extend standard protocols by adding custom features. Part IV: OPNET Modeling Facilities covers content that is used to facilitate OPNET modeling, including debugging, hybrid simulation, External Model Access (EMA), co-simulation, programming OPNET models in C++, etc.
We thank deeply the various people who, during the months over which this endeavor lasted, provided us with useful and helpful assistance. Without their care and consideration, this book would likely not have matured.

First, we thank Dr. David K. Hunter and Dr. Yixuan Qin, who gave us useful suggestions and comments before and during the writing of the book.

Second, we thank the publisher and people who demonstrated interest in publishing this book. The production team at Cambridge University Press has been great. Many thanks go to people who helped us with the book development, including Mrs. Sarah Marsh and Dr. Julie Lancashire.

Dr. Zheng Lu would like to thank his wife Dr. Gui Gui; without her support, he could not have got through that difficult time and thrown himself into finishing the book.

Professor Hongji Yang would like to thank his wife, Xiaodong Zhang, for her full support in finishing the writing of this book.

Trademark acknowledgments: OPNET is a trademark of OPNET Technologies, Inc. All other product names mentioned herein are the trademarks of their respective owners. The relevant screenshots in this book are used with authorization by OPNET Technologies, Inc.

Zheng Lu
Hongji Yang
Abbreviations

API application programming interface
BSS basic service set
CDB Microsoft Console Debugger
CMO Categorized Memory
DB diagnostic block
DES discrete event simulation
EMA External Model Access
ESA External Simulation Access
ESD External System Definition
Esys External System
ETS external tool support
FB function block
FPP Fractal Point Process
GDB GNU Project Debugger
GUI graphic user interface
HB header block
ICI Interface Control Information
IDE Integrated Development Environment
KP Kernel Procedure
LAN local area network
MSVC Microsoft Visual C++ Debugger
ODB OPNET Simulation Debugger
ODK OPNET Development Kit
PDF probability density function
PMO Pooled Memory
PPP Point to Point Protocol
QoS quality of service
RPG raw packet generator
SDK software development kit
STD state transition diagram
STL Standard Template Library
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV</td>
<td>state variable</td>
</tr>
<tr>
<td>TB</td>
<td>termination block</td>
</tr>
<tr>
<td>TV</td>
<td>temporary variable</td>
</tr>
<tr>
<td>UI</td>
<td>user interface</td>
</tr>
<tr>
<td>WAN</td>
<td>wide area network</td>
</tr>
<tr>
<td>WLAN</td>
<td>wireless local area network</td>
</tr>
</tbody>
</table>