Unlocking the Power of OPNET Modeler

ZHENG LU
HONGJI YANG

CAMBRIDGE UNIVERSITY PRESS
Contents

Preface xi
List of abbreviations xiii

Part I Preparation for OPNET Modeling

1

Introduction

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
3  OPNET Modeler user interface  
   3.1 Project management  
   3.2 Modeler preferences  
   3.3 OPNET editors  
      3.3.1 Project Editor  
      3.3.2 Node Editor  
      3.3.3 Process Editor  
      3.3.4 Link Editor  
      3.3.5 Packet Format Editor  
      3.3.6 ICI Editor  
      3.3.7 PDF Editor  
      3.3.8 Probe Editor  
   3.4 Simulation Results Browser  
   3.5 Animation Viewer  
   3.6 Using OPNET documentation  

Part II  Modeling Custom Networks and Protocols  

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

5  Creating and simulating custom models using OPNET APIs  
   5.1 General procedure for creating and simulating custom models  
   5.2 Custom models  
      5.2.1 Case 1  
      5.2.2 Case 2  
      5.2.3 Case 3
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
10 Modifying standard models 151
10.1 Introduction 151
10.2 Case study 151

Part IV OPNET Modeling Facilities 165

11 Debugging simulation 167
11.1 Debugging facilities in OPNET Modeler 167
11.1.1 Prerequisites for debugging 168
11.1.2 Preparing simulation scenario 168
11.1.3 Debugging with ODB 169
11.1.4 Debugging with CDB/GDB 175
11.1.5 Debugging with Microsoft Visual C++ Debugger 177
11.1.6 Debugging with animation 179

12 OPNET programming in C++ 182
12.1 Proto-C, C, and C++: language and library differences 182
12.2 Memory management differences between Proto-C APIs and C/C++ standard library functions 182
12.3 Proto-C data structures and algorithms packages, C++ standard template libraries (STL) and Boost C++ libraries 184
12.4 Environment configurations for C++ programming in OPNET 185
12.5 Case study on programming OPNET models in C++ 187

13 Traffic in OPNET simulation 194
13.1 Introduction 194
13.2 Explicit traffic 194
13.2.1 Explicit traffic based on application 195
13.2.2 Explicit traffic based on traffic generation parameters 196
13.2.3 Explicit self-similar traffic based on raw packet generator (RPG) model 197
13.3 Background traffic and hybrid simulation 200
13.3.1 Background traffic based on baseline load 201
13.3.2 Background traffic based on traffic flow 202

14 External model access (EMA) 207
14.1 What EMA is and reasons to use it 207
14.2 EMA case study 208

15 OPNET co-simulation with third-party programs 215
15.1 Co-simulation with external programs 215
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  

## 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