Introduction to OPNET Modeler

Steven Gordon

Sirindhorn International Institute of Technology
Thammasat University

June 2010
Contents

OPNET and OPNET Modeler

Simulation Analysis Methodology with OPNET
OPNET and OPNET Modeler

Who is OPNET? www.opnet.com

- US company creating/selling network design and management software
- Founded by MIT graduate that developed a network simulator OPNET Modeler

What is OPNET Modeler?

- Discrete event network simulation software
- Used for network/protocol design and analysis

Other OPNET Products

- ACE Analyst, IT Guru Systems Planner, . . .
- IT Sentinel, nCompass, IT Guru Network Planner, . . .
Why OPNET Modeler for Research?

Alternatives: ns2/ns3, Qualnet/Glomosim, OMNeT++, ...

Advantages

▶ Extensive library of detailed models
▶ Long development history, experience
▶ Detailed documentation and technical support
▶ Built-in statistics collection and results presentation
▶ Logical, hierarchical structure of models

Disadvantages

▶ Expensive!
▶ Closed-source simulator
▶ Slower to include models of new technologies
Why OPNET Modeler for Business?

Design New Networks

- Design WiMax, IPv6, 3G/LTE networks
- Compare alternative technologies: WiMax vs 3G

Plan Network Upgrades

- What-if scenarios: “What if replace DSL with WiMax?”, “What if video calls are supported?”
- Capacity planning: “How many base stations are needed to support $X$ users?”

Evaluate Custom Algorithms and Applications

- Optimise scheduling algorithms, evaluate QoS policies
- Impact of your applications on network performance
Why OPNET Modeler for Business?

Customers

Aperto Networks Uses OPNET Modeler to Reduce Time-to-Market for New WiMAX Solutions
Booz Allen Hamilton Chooses OPNET for SAFECOM Public Safety Program
BT Selects OPNET for Researching End-to-End QoS
Clean Earth Technologies Selects OPNET Modeler to Design Communication Systems for Airborne Networking
Datang Mobile Uses OPNET Modeler to Validate 3G Network Device Designs Before Production
Ericsson Leverages OPNET Modeler to Develop MPLS/QoS Mechanisms for 3G Wireless Access Networks
Motorola Leverages OPNET to Design the neuRFon Wireless Information Network
NEC Selects OPNET for WiMAX Base Station Design
Philips Designs WiMedia Standard using OPNET Modeler
Workshop Outcomes

Apply OPNET Modeler to network research

1. Understand how OPNET Modeler is structured
2. Navigate the menus and operations of OPNET Modeler
3. Create models of networks using existing models
4. Configure, run and analyse results from simulations
5. Knowledge of existing device models and capabilities
6. Understand the implementation of Process Models
7. Create your own model of a protocol
Assumed Knowledge

- Computer networking, e.g. layering, protocol mechanisms, performance metrics
- C programming
- Ability to explore menus, manuals and files on your own
Workshop Format

Topics

1. Introduction to OPNET
2. Analysing networks
3. Editors and features
4. Process Models

Hands On

1. OPNET examples
2. Ethernet/WLAN
3. OPNET examples
4. Sensor Network
Contents

OPNET and OPNET Modeler

Simulation Analysis Methodology with OPNET
Performance of Real Networks

A real network has:

- Topology: arrangement of devices and links
- Configuration: protocols, parameter values and options selected
- Users: generating and receiving traffic via applications
- Mobility: users/devices moving

Understanding performance of the network

Specify a set of statistics to measure during operation

- Application response time, link utilization, data throughput, error rates, ...
A Methodology for Simulation Performance Analysis

1. Create a network topology
Select the area of the network; position hosts, switches, routers, antennas, servers in the area; connect devices via links
A Methodology for Simulation Performance Analysis

2. Configure devices and protocols

Examples: set link data rate to 1Mb/s; router forwarding to 500,000pps; mobile host to use IEEE 802.11g
A Methodology for Simulation Performance Analysis

3. Specify traffic from users

Select the types of applications users are using, e.g. web browsing, email, voice call; specify the characteristics of the applications, e.g. voice source generates 50pps at 128Bytes.
A Methodology for Simulation Performance Analysis

4. Specify the mobility of users/devices

Specify a trajectory of devices throughout network area; user-defined or modelled (e.g. random waypoint, grid-based)
5. Select statistics to measure
Per node or entire network; throughput, bytes sent/received, delay, jitter, server load, link utilisation, ...
A Methodology for Simulation Performance Analysis

6. Setup and run simulations
Duration of simulation; number of runs and random seeds; simulation parameters
A Methodology for Simulation Performance Analysis

7. Analyse the results
Raw data, plots and reports
OPNET Modelling Hierarchy

**Network Model**
Create network topology using existing node models

**Node Model**
Models of devices (e.g. switches, PCs, routers, links)
Created using existing process models

**Process Model**
Models of applications and protocols (e.g. HTTP, TCP, IEEE 802.11)
Created using state diagrams and C source code
Model Library

Protocols
ATM, BGP, DHCP, DOCSIS, EIGRP, EthCoax, Ethernet, FibreChannel, FDDI, FrameRelay, H323, IGRP, IKE, IP, ISIS, L2TP, LANE, LAPB, OSPF, RIP, RSVP, RTP, SIP, TCP, TDMA, X25, xDSL

Applications
HTTP, FTP, Email, Database, Voice, Video, Print, Rlogin, Generic request/response

Wireless
IEEE 802.11, MANET (AODV, DSR, OLSR, …), ZigBee, WiMax, UMTS

Others
MPLS, PNNI, PSTN, Servers, community contributed models
Internal Structure of OPNET

- A network consists of multiple nodes; each node consists of multiple processes
- For a simulation, OPNET compiles source code (C/C++) of each Process model used
- Also included is Simulation Kernel:
  - Controls the execution of processes
  - Manages list of events to occur
- Resulting executable is then run, saving output in results files
States and Events

Process models are state-based
In state $x$ if event $e$ occurs then action $a$ is taken and enter state $y$
Implemented as graphical state-machines and C/C++ code

Events are called interrupts

► A process schedules interrupts, e.g. timeouts
► Kernel issues interrupts, e.g. packet arrives, channel status changes

Kernel manages the Event List
0.004503 Timer $t_1$ expires at process $X$
0.004515 Packet $p_1$ arrives at process $Y$
0.004602 Packet $p_2$ arrives at process $X$
0.004603 Timer $t_2$ expires at process $Z$
...