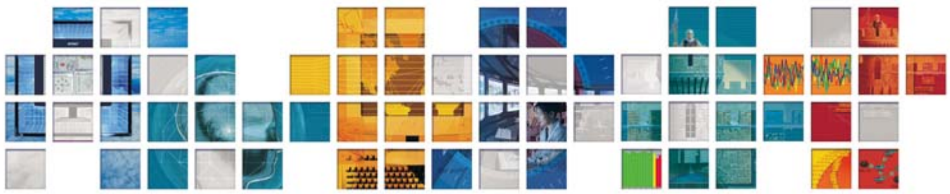
OPNET Technologies, Inc.

Making Networks and Applications Perform™

Introduction to OPNET Modeler



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

What is Modeler?

- An environment for building protocols and device models.
- An environment to plan changes by illustrating how the networked environment will perform.
- An environment that includes hundreds of pre-built models, used to study performance changes of your network: organizational scaling, technology changes, and application deployment.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Questions and Goals

- Questions to be answered in the OPNET Modeler Class
  - –What does Modeler do?
  - –How can it help me?
  - –How easy is it to use?
  - –How can I get started?
- Goals
  - –Learn what problems can be solved with Modeler
  - –Learn how to use the Modeler software in a hands-on environment
  - –Use Modeler to solve real problems

OPNET Training

Expectations

- Prerequisites:
  - –Ability to program in C or C++ or at least be comfortable in reading/understanding C or C++ code
  - –Basic understanding of networks
    - For basic information, see
      - –http://compnetworking.about.com/library/glossary/blglossary.htm
      - –http://www.whatis.com
- This class will move quickly
- Ask questions. It will enhance your learning experience, as well as the other students'.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Course Content

DAY 1:

- Introduction
- OPNET Technologies
  - History of OPNET
  - OPNET Services and Support
  - Practical applications of Modeler
- Features of the Project Editor {LAB}
- Building a First Network {LAB}
- Modeling Framework
- Other Editors {Mini-labs}
- General Simulation Methodology

DAY 2:

- Events and Event List Concepts
- Node Modeling
- Node Modeling {LAB}
- Process Modeling
- Process Modeling {LAB}
- Collecting Scalar Statistics
- Scalar Statistics {LAB}

DAY 3:

- Modeling Large Networks
- Importing Topology and Traffic
- Wireless Modeling
- Wireless Modeling {LAB}
- OPNET Debugger {LABs}
- Publishing & Animation {Mini-labs}
- Overview of Modules {Demo}
- Summary & Questions

Format

- Purpose of the class: To give students a hands-on learning experience and introduce many of the features of Modeler.
- About half lecture, half lab
- More lecture at the beginning – this will lay a strong foundation
- Lecture is interactive – do not hesitate to ask questions
- Labs not completed in class may be completed on your own

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

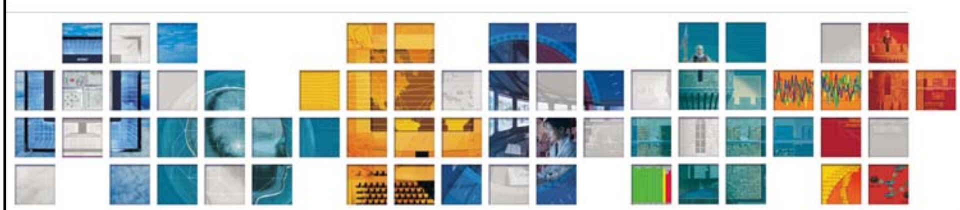
© Copyright 2007 OPNET Technologies, Inc.

Format

- During labs, raise your hand and an assistant will help you.
- Concepts will be taught during lecture, and you will apply them to labs. We encourage you to ask questions.
- Fill out the "sign-in" sheet and the "course evaluation" before you leave. Please provide constructive feedback for improvements to the OPNET software, user interface, documentation, and training session.

OPNET Training

OPNET Technologies



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET® Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

About OPNET Technologies, Inc.®

Corporate Overview

- Founded in 1986
- Publicly traded (NASDAQ: OPNT)
- HQ in Bethesda, MD
- 552 employees as of December 31, 2007
- Worldwide presence through direct offices and channel partners
- Global distribution agreement with CA

Best-in-class Software and Services

- Application Performance Management
- Network Operations
- Capacity Management

Strong Financial Track Record

- Long history of profitability
- Trailing 12-month revenue of $98.7M
- Re-invest heavily in R&D

Broad Customer Base

- Corporate Enterprises
- Government Agencies/DoD
- Service Providers

Network R&D Clients (partial list)

Aeronautic Computing Technique Research
Alcatel
Alion Science and Technology
Aperto Networks
AutoNetworks Technologies, Ltd.
Avaya
BAE Systems
Ball Solutions Group
Bechtel Telecommunications
Booz Allen & Hamilton
Broadcom
CACI Federal Systems
CAS Inc.
CISCO Systems
Concurrent Technologies Corp.
DISA
Defence Communication Services Agency UK
Defence Science & Technology Org. Australia
Department of National Defence - Canada
Dynamics Technology
EADS
Eagan, McAllister Associates
Eagle Alliance
Elbit Israel
Ericsson
Fujitsu
General Dynamics
Generic Systems Sweden AB
Georgia Tech Research Institute
Halliburton
Harris Communications
Hewlett Packard
Highland Systems
Hitachi Ltd.
Honeywell
Hughes Network Systems
IBM Global Services
Innovative Wireless Technologies
Intel Corporation
Internap
Japan Defense Agency
JHU APL
L-3 Communications
Lucent Technologies
MIT Lincoln Labs
MITRE
Mitretek
NASA
National Communications System
National Geospatial Intelligence Agency
National Institute of Standards and Tech.
NEC
Northrop Grumman
ntl
Oak Ridge National Lab
Philips
Raytheon
Research in Motion Ltd.
RF Monolithics
Ricoh Company Ltd.
Rockwell Collins
SAIC
Sandia National Labs
SELEX Communications Spa
Sonus Networks
Space and Naval Warfare Systems Command
Sparta
Stellaris Networks
Syracuse Research Corporation
Systems Planning & Analysis, Inc.
Tekelec
Telenor
Tenacity Solutions Inc
Textron
U.S. Air Force
U.S. Army
U.S. Army Research Lab
U.S. Central Command
U.S. Coast Guard
U.S. Department of Homeland Security
U.S. Marine Corps
U.S. Military Academy
U.S. Naval Research Lab (NRL)
U.S. Navy

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Network R&D Solutions

- OPNET Modeler is the de-facto standard for:
  - –Network R&D
  - –Modeling and simulation
  - –Defense organizations
  - –Network equipment manufacturers

- Accelerate network R&D and planning:
  - –Design wired and wireless protocols and technologies
  - –Test and demonstrate designs in realistic scenarios before production
  - –Plan mobile network deployments that accurately incorporate terrain effects
  - –Design wireless network protocols to optimally support the warfighter
  - –Assess battlefield plans in light of communications effects

Solution Portfolio

OPNET Modeler
Accelerating Network R&D

OPNET Modeler Wireless Suite
Wireless Network Modeling and Simulation

OPNET Modeler Wireless Suite for Defense
Modeling and Simulation for Defense Communications

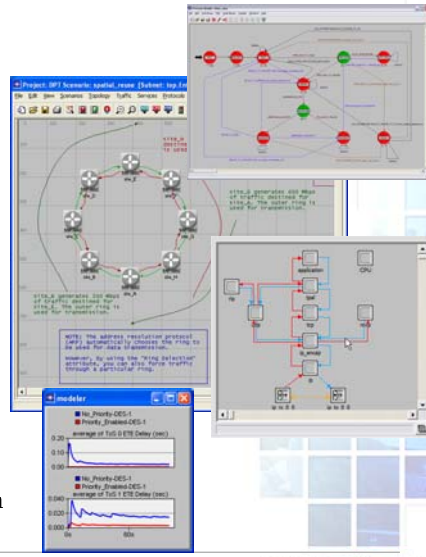
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Accelerating Network R&D

Modeler
Accelerating Network R&D

- Design network protocols and technologies
- Demonstrate technology designs in realistic scenarios
- OPNET Modeler offers
  - –Scalable simulation engine
  - –Hundreds of protocol and vendor device models with source code
- Test prototype network devices with System-in-the-Loop*
  - –Interface simulations with real hardware/software
  - –Combine the real world and "virtual" world
  - –Test scenarios that cannot be achieved in a lab environment

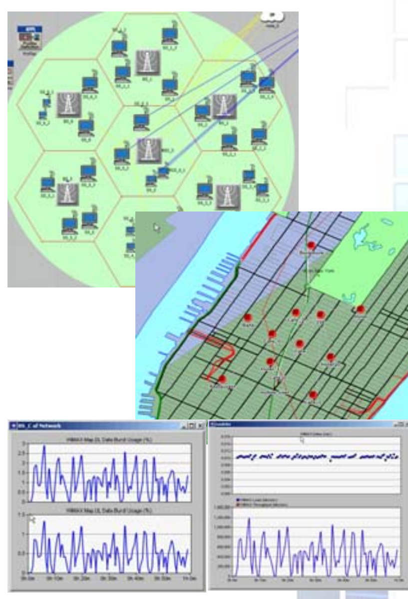


*Optional module

Wireless Network Modeling and Simulation

Modeler Wireless Suite
Wireless Network Modeling and Simulation

- Design wireless protocols and technologies
- Optimize protocol and application performance
- Supports numerous wireless systems (cellular, satellite, MANET, WiFi, WiMAX, etc.)
- Incorporate terrain effects into mobile network deployment planning
- OPNET Modeler offers:
  - –Scalable simulation engine
  - –Hundreds of protocol and vendor device models with source code
- Test prototype network devices with System-in-the-Loop*
  - –Interface simulations with real hardware/software
  - –Combine the real world and "virtual" world
  - –Test scenarios that cannot be achieved in a lab environment



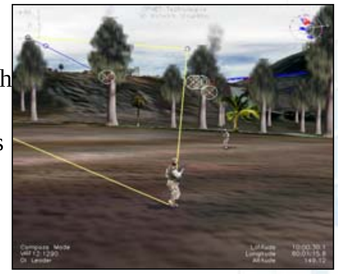
*Optional module

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

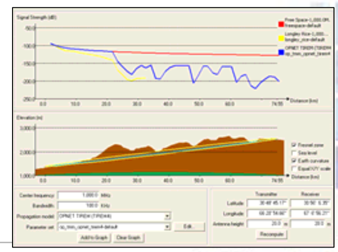
© Copyright 2007 OPNET Technologies, Inc.

Modeling and Simulation for Defense Communications

Modeler Wireless Suite for Defense
Modeling and Simulation for Defense Communications



- Adds to capabilities of Modeler Wireless Suite
- Design protocols and infrastructures to support th warfighter
- Demonstrate technology designs in 3D scenarios
- Conduct realistic wargaming exercises by incorporating communications effects
- OPNET Modeler offers:
  - –Scalable simulation engine
  - –Hundreds of protocol and vendor device models
- Test prototype network devices with System-in-the-Loop*
  - –Interface simulations with real hardware/software
  - –Combine the real world and "virtual" world
  - –Test scenarios that cannot be achieved in a lab environment



*Optional module

OPNET Products and Services

- Our web site - http://www.opnet.com
  - –Announcements
  - –Up-to-date logged problems
  - –User Forum Mailing Lists
  - –Download this course material
- Technical Support Center
  - –Available via telephone (240-497-3000), fax (240-497-3001), and e-mail (support@opnet.com)
  - –6 A.M. to 10 P.M. (EST), Monday through Friday (excluding holidays)
  - –Staffed by OPNET Engineers

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Products and Services

- Training Services – Regular and customized classes for all products
- C4 – Client Consultation and Computing Center
  - State-of-the-art computer resources
  - Application Engineers available
  - No charge to maintained commercial customers
  - Technical Support "in person"

OPNET Training

OPNET Consulting Services

- **IT Services for Commercial and Government Enterprises**
  - "Jump Start" Services
  - Application Performance Optimization
  - Network Performance Optimization
    - Capacity planning
    - Rules-based configuration analysis
    - Network security policy analysis
  - Server Capacity Planning and Consolidation
  - Disaster Recovery and Resiliency Planning
  - Enterprise Architecture Assessment
  - Network and Application Data Collection and Fusion

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Consulting Services

- Custom software development
  - Specialized planning solutions
  - Integration with 3rd party simulation and 3D visualization tools
  - C4ISR, optical, and wireless ad hoc network solutions
  - Network common operating picture (NETCOP) solutions
- Staff augmentation at customer sites
  - Fully cleared staff up to TS/SCI
  - SCIF at OPNET HQ
- Proposal support
- Custom model development, validation, performance optimization, etc.
- Modeling & simulation for NETWARS, transformational communications, battlefield networks and other scenarios

OPNET Training

August 27 - 31, Washington D.C.

OPNETWORK is OPNET's annual technology conference, focusing on intelligent analysis of networks, applications, and systems. It is the largest event of its kind, attracting thought leaders from industry, government, and academic communities from all over the world.

RSVP at: www.opnet.com/opnetwork2007

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Features of the Project Editor

OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Overview

- Project Editor:
  - –Workflow
  - –More Details of the Project Workspace
    - Map Backgrounds
    - Zooming
    - Threshold Value
    - Annotation Palette
    - Project Workspace {Lab}
  - –Models
    - Deriving and Creating New Devices {Lab}
  - –Object Attributes
    - Client-Server Configuration Example
  - –Statistics
  - –Product Documentation

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

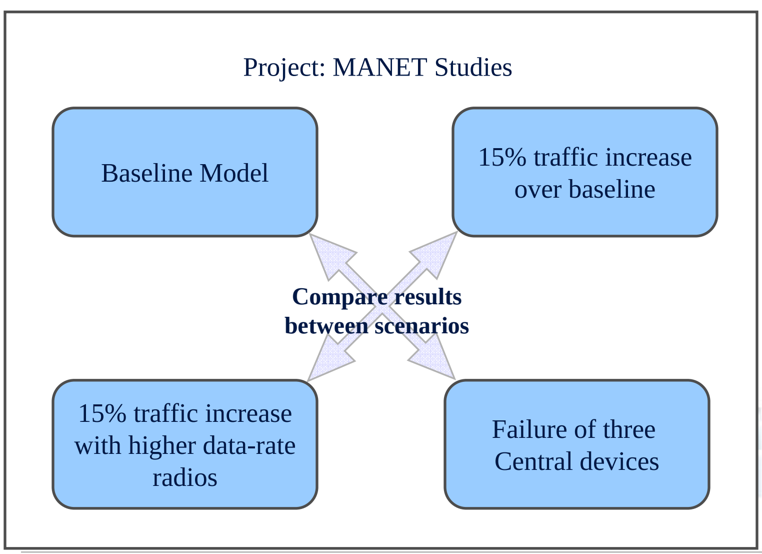
© Copyright 2007 OPNET Technologies, Inc.

Projects and Scenarios

- Modeler uses a Project-and-Scenario approach to modeling networks.
- A *Project* is a collection of related network scenarios in which each explores a different aspect of network design. All projects contain at least one scenario.
- A *Scenario* is a single instance of a network. Typically, a scenario presents a unique configuration for the network, where configuration can refer to aspects such as topology, protocols, applications, baseline traffic, and simulation settings.

OPNET Training

Projects and Scenarios



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

The Project/Scenario Workflow

- Create project
- Create baseline scenario
  - –Import or create topology
  - –Import or create traffic
  - –Choose results and reports to be collected
  - –Run simulation
  - –View results
- Duplicate scenario
  - –Make changes
  - –Re-run simulation
  - –Compare results

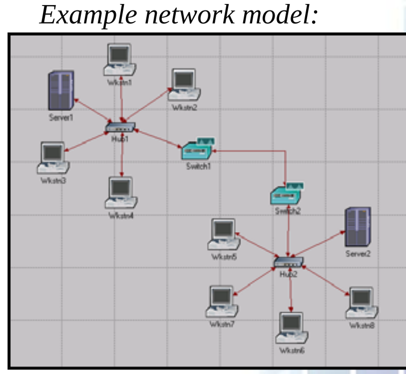
Iterate

Project Editor

- Several tools are included in Modeler to intuitively map from network specifications to a modeled network.

- –Use the **Startup Wizard** to specify the initial environment of a scenario.
- –Select objects from an **Object Palette.**
- –Use **Node and Link** objects to represent topology.
- –Use **Rapid Configuration** to quickly deploy common network topologies.
- –Edit the **Attributes** of nodes and links to customize their behavior.

Example network model:

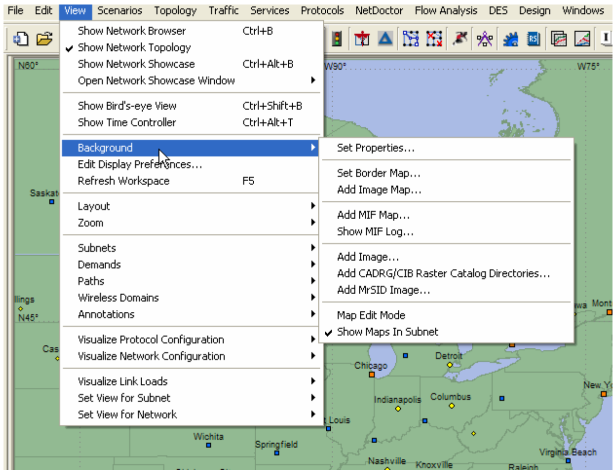


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Setting Background Maps

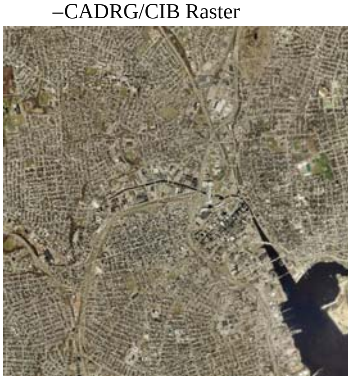
- View/hide maps by choosing **View / Background / Set Properties**
- **Geotiff, MapInfo, MrSID, and CADRG** images automatically appear at the correct latitude and longitude position.

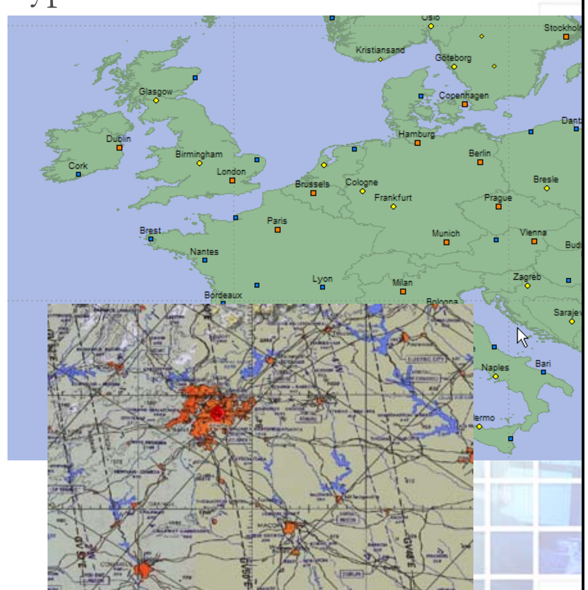


OPNET Training

Map Backgrounds Types

- MIF maps (default)
- TIFF and GeoTIFF
- Geospatial maps not provided, but can be displayed
  - –MrSID
  - –GeoTIFF
  - –CADRG/CIB Raster





OPNET Training

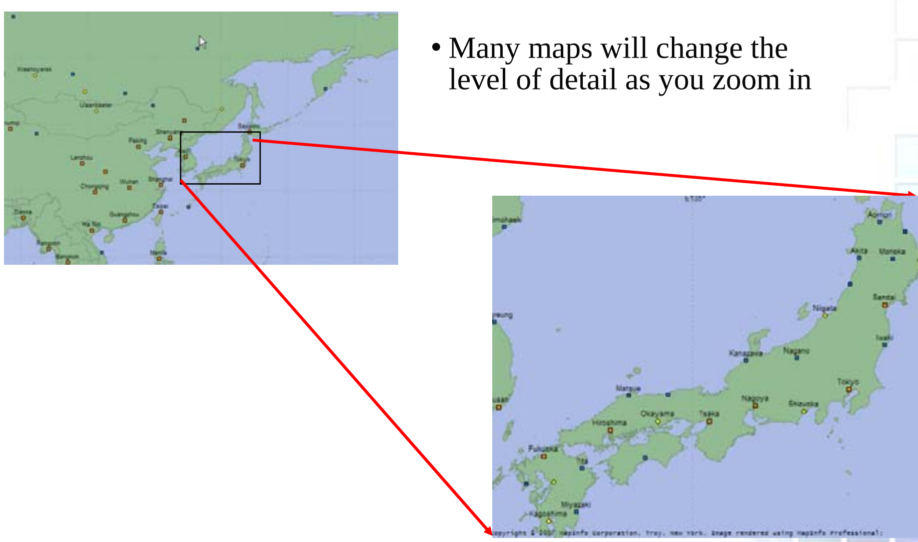
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Zooming and Panning

- Zooming can be performed with the scroll wheel on your mouse:
  - –To zoom in, turn the scroll wheel forward or use the Page Down button.
  - –To zoom out, push the scroll wheel backward or use the Page Up button.
- The “Zoom to Window” feature allows you to return to the original magnification level
- You can right-click in the workspace and choose
  - –“Zoom to Selection” to zoom in on a selected object
  - –“Zoom to All” to zoom to all objects in the subnet
  - –“Zoom In”
  - –“Zoom Out”
  - –“Zoom to Window” to zoom to the full subnet view
- To pan, hold down the scroll wheel and drag the mouse pointer. Optionally, hold down the right and left mouse buttons and drag the mouse pointer.

Zooming and Panning



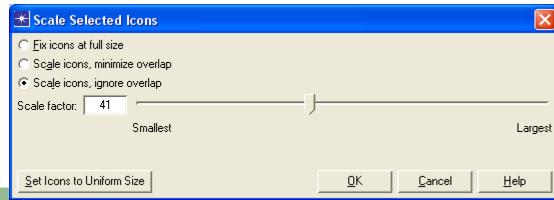
- Many maps will change the level of detail as you zoom in

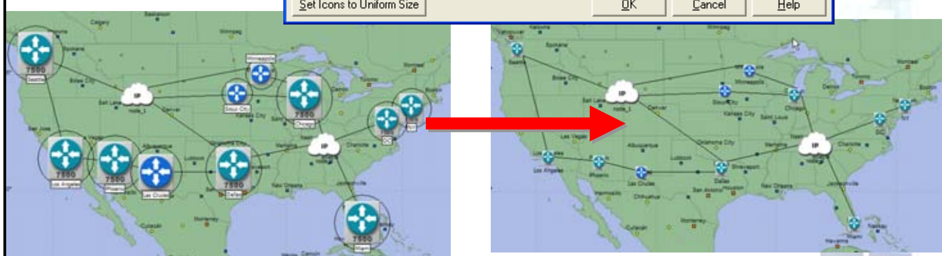
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Changing Icon Size

- You can scale the size of an icon on the screen to improve the appearance of the topology
- **View / Layout / Scale Node Icons Interactively…**

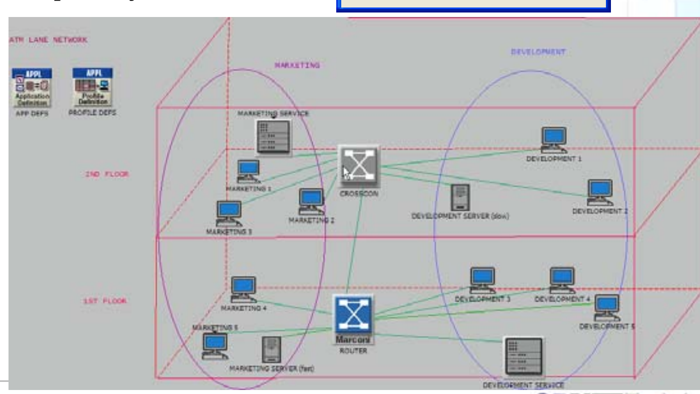




OPNET Training

Annotation Palette

- Open by selecting **Topology / Open Annotation Palette**
- Add rectangles, circles, lines and text to models to enhance their appearance
- This method is an effective way to graphically illustrate and document the changes made to a model
- Does not affect analysis
- Annotations can be temporarily hidden



OPNET Training

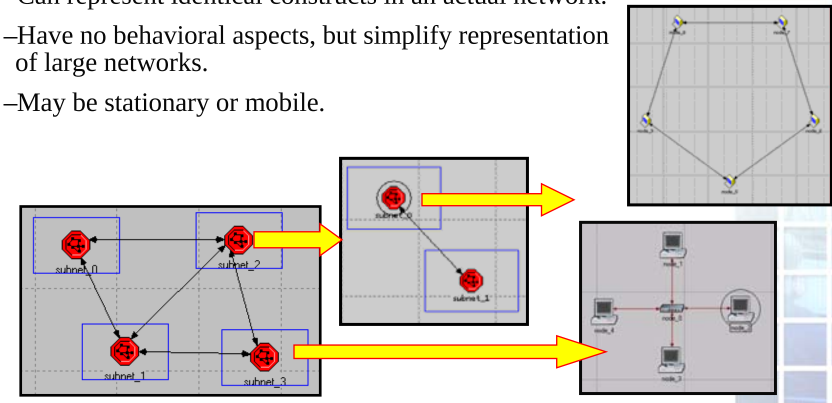
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Subnets

- A subnet object abstracts network components specified within it into a single container
- Subnet objects:
  - –Are simply containers used to organize your nodes
  - –Can represent identical constructs in an actual network.
  - –Have no behavioral aspects, but simplify representation of large networks.
  - –May be stationary or mobile.

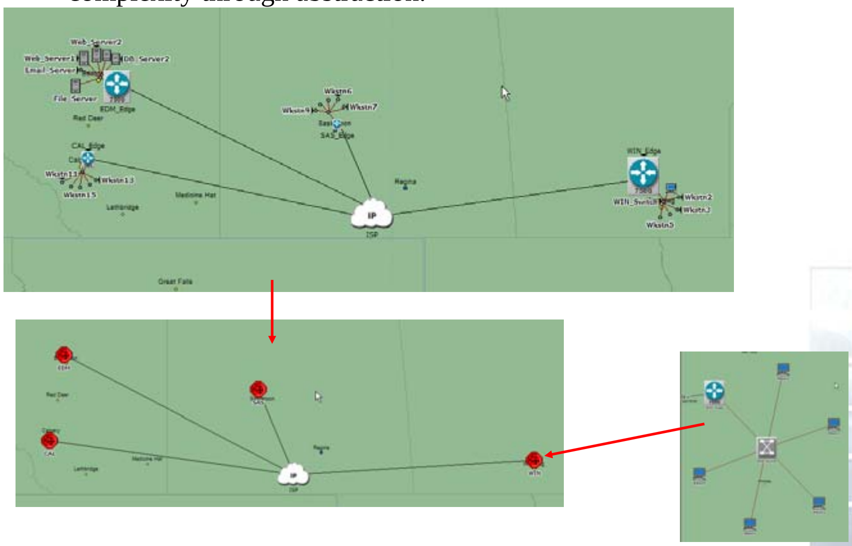
4 Subnet Types



OPNET Training

Why Use Subnets?

- Subnets are a powerful mechanism for breaking down a system's complexity through abstraction.



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Object Hierarchy

- Subnets represent various network components in a single object. This object could contain various LAN, node, and link models.
- LANs represent a local area network abstracted into a single node.
- Nodes represent servers, workstations, switches, gateways or any other physical devices associated with communication networks.
- Links represent the physical links between nodes.

Locating the Right Models and Components

- Selecting the right models and components is essential to accurate modeling.
- Modeler provides a suite of standard and vendor models.
- You can also create new models or derive models from an existing database.
- Modeler’s naming conventions, custom palettes, icons, and derived models all facilitate your selection of the correct components for your network.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Model Library

- The "Model Library" is an extensive library of standards-based and vendor models.
- Our Modeling Dept. is responsible for design, development, and on-going evolution of the "model library".
- Models are created using published protocol standards and other widely used vendor implementations.
  - IEEE Specifications (e.g., 802.1q, 802.3, 802.11, etc.)
  - ANSI Standards (e.g., X3.139, T1.513, etc.)
  - ATM Forum Specifications (e.g, UNI 3.1, TM 4.0, etc.)
  - RFC Documents (e.g., RFC-793, RFC-1058, RFC-1771, etc.)
  - Vendor Technologies (e.g., VLAN tagging, EIGRP, etc.)

Model Library Components

- The "Model Library" contains a variety of objects used for creating networks
  - Traffic generators (workstations, servers, stations, etc.)
  - Network devices (hubs, bridges, switches, routers, etc.)
  - Links (SONET, PPP, FDDI, 10BaseT, ISDN, xDSL, etc.)
  - Vendor device models (Cisco Systems, 3Com, Nortel, Lucent, HP, etc.)
- These models are ready-to-use to create networks and predict performance behavior.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Model Libraries

- Standard
  - –150+ built-in protocol and vendor device models including:
    - Transport: TCP (ECN, New Reno, Reno, SACK, Tahoe), UDP, …
    - Network & Routing: BGP, EIGRP, IGRP, IP, IPv6, OSPF, OSPFv3, RIP, RIPng, …
    - MAC: 802.1p, ATM, Ethernet (802.3,a,i,j,u,y,z,ab,ac,ad,ae), Spanning Tree, VLAN, …
    - Wireless: MANET (AODV, OLSR, DSR, TORA, OSPFv3), Mobile IPv4, Mobile IPv6, WLAN (802.11a,b,e,g), …
    - Vendor device: Cisco, Extreme, Foundry, HP, IBM, Intel, Juniper, Lucent, Motorola, NEC, Nortel, Sun, …
- Specialized
  - –Selected protocol models available separately
  - –IPv6, MPLS, DOCSIS, UMTS, and PNNI)

Model Libraries

- NETWARS (for DoD and approved contractors)
  - –100+ military protocol and device models available through the NETWARS program
  - –Compatible with all Modeler solutions
  - –Satellites and earth terminals, encryption devices, multiplexers, tactical radios, etc.
- Contributed
  - –Models contributed by OPNET users
  - –1,000+ universities using OPNET software and 25,000+ academic users
- Model Development Consortium
  - –Consortium of prominent organizations that jointly guide model development (WiMAX (802.16e))

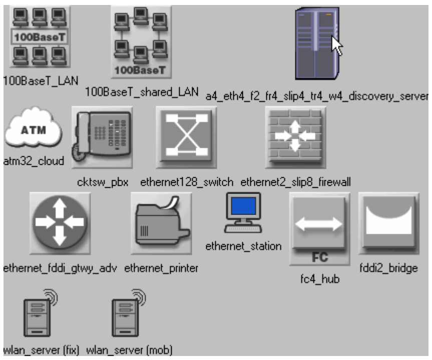
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

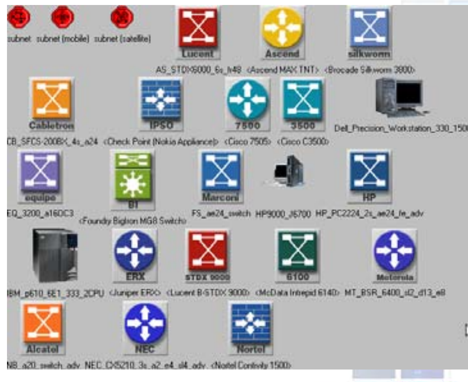
Network Objects

- Network models consist of *nodes, links and subnets*.
- Nodes represent network devices -- servers, workstations, routers, etc.
- Links represent point-to-point and bus links.

Generic Devices



Vendor Devices



OPNET Training

Model Naming Convention

- OPNET's standard model suite uses a specific naming convention for its node models:

  <*protocol*n>…<*protocol*n >_<*function*>_<*modifier*>

- <*protocol*n> is an abbreviation that specifies a particular protocol that is supported within the model. The variable n represents the number of protocol interfaces within that model.
- <*function*> is an abbreviation that indicates the general function of the node, e.g. gateway, workstation, server.
- <*modifier*> indicates the level of derivation of the model.

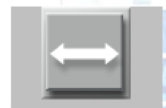
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Model Naming Convention (cont.)

- Levels of derivation (<*modifier*> value)
  - –*adv*: advanced model. All node attributes are available and attribute values are set to their defaults.
  - –*int*: intermediate model. Attributes that are unlikely to be changed are hidden and typical values are applied to the visible attributes.
  - –Models with no value for modifier are derived from intermediate models. Additional attributes are hidden and only attributes needed for parametric studies are visible.



- Example: ethernet128_hub_adv

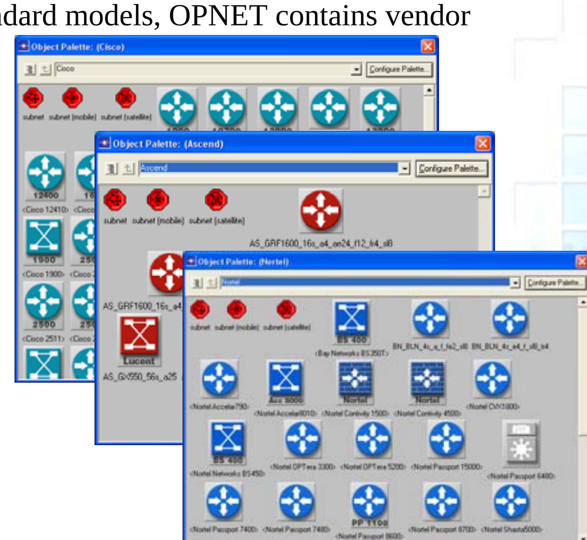
  This model represents a hub with 128 ethernet interfaces. Since it is an advanced model, all the model's attributes are visible and editable.

OPNET Training

Vendor Models

- In addition to our standard models, OPNET contains vendor specific models.

3Com	Fore Systems
Ascend	Foundry
Avici	Hewlett Packard
Bay Networks	Juniper Networks
Brocade	Lucent
Cabletron	McData
Cisco Systems	NEC
Equipe	Newbridge
eXtreme	Nortel



OPNET Training

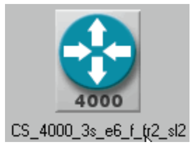
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Vendor Models (cont.)

- Naming convention for vendor models

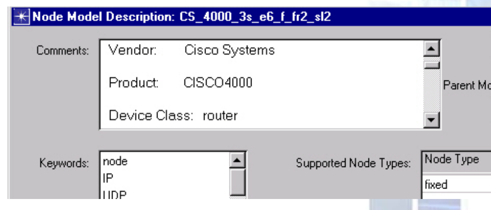
  <Vendor_Name>_<Device_Name>_<Number_of_Slots>_<Configuration>



Example:
CS_4000_3s_e6_f_fr2_sl2

This model represents a Cisco Systems 4000 Router with 3 slots, 6 ethernet ports, 1 FDDI port, 2 frame relay ports, and 2 SL-IP ports.

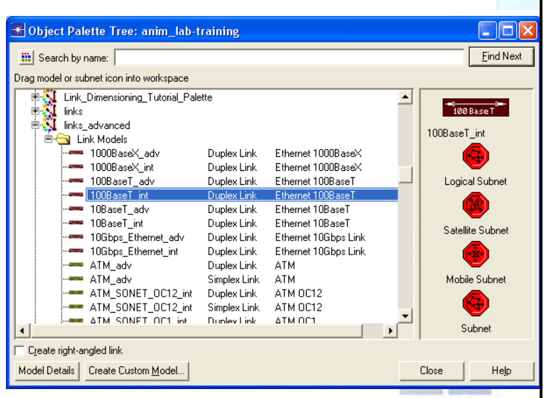
- View description of the node by:
  - –Right-clicking on the node in the object palette.
  - –Right-clicking on the node in the workspace and selecting "View Node Description".



OPNET Training

Choosing Link Models

- An appropriate link must be chosen to match interfaces on connected devices.
  - –Data rate, protocol, and number of available interfaces are the requirements that must be met.
- Link model naming convention:
  - –<protocol>_<modifier>
- Example:
  - –100BaseT_int



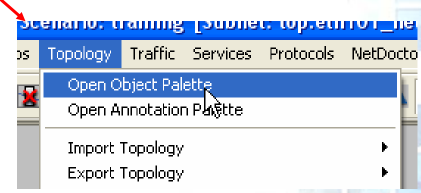
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Organizing Models - Object Palettes

- An object palette is a dialog box that displays a group of node and link models
- Open the object palettes by
  - –Clicking on the “Open Object Palette” button on the toolbar
  - –Choose “Open Object Palette” under the menu “Topology”
- The object palettes have two viewing modes: Tree View and Icon View

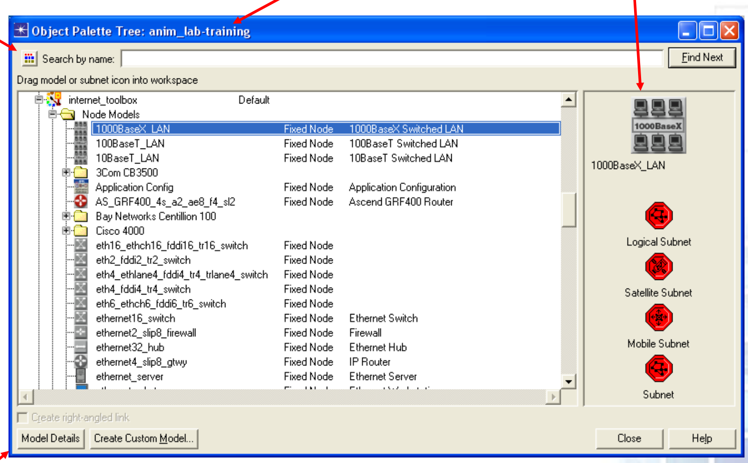


Object Palettes – Tree View

Name of current Project and Scenario

Selected model icon

Switch to icon view



Display details of currently selected model

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

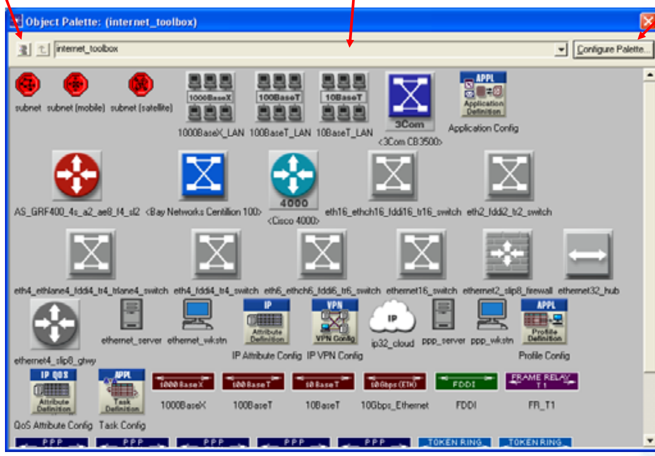
© Copyright 2007 OPNET Technologies, Inc.

Object Palettes – Icon View

Switch to tree view

Model list pull-down menu

Modify this palette or create new palette



Right-click an icon to display details of a model

OPNET Training

Customizing Object Palettes

- The palette always contains subnet objects
- Object palettes can be customized to contain only the nodes and links that you require
- There are several different methods for customizing palettes

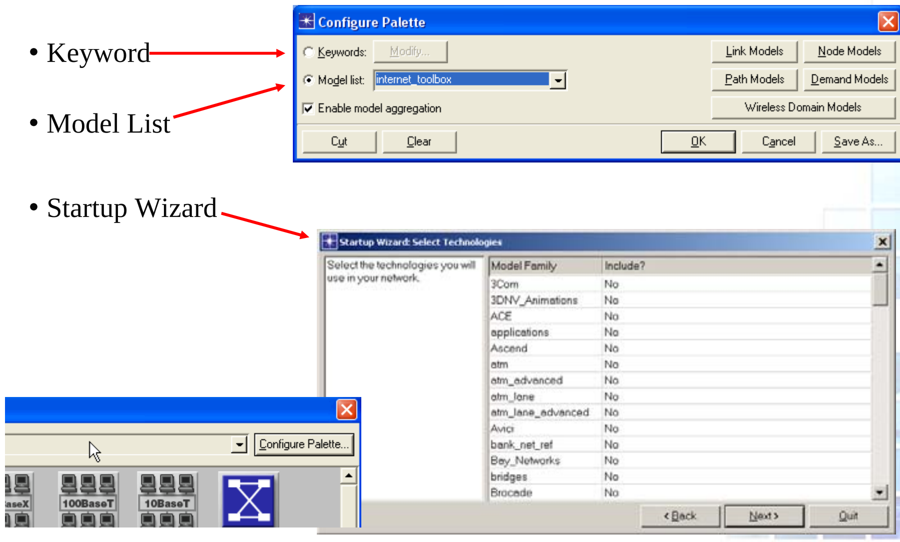
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

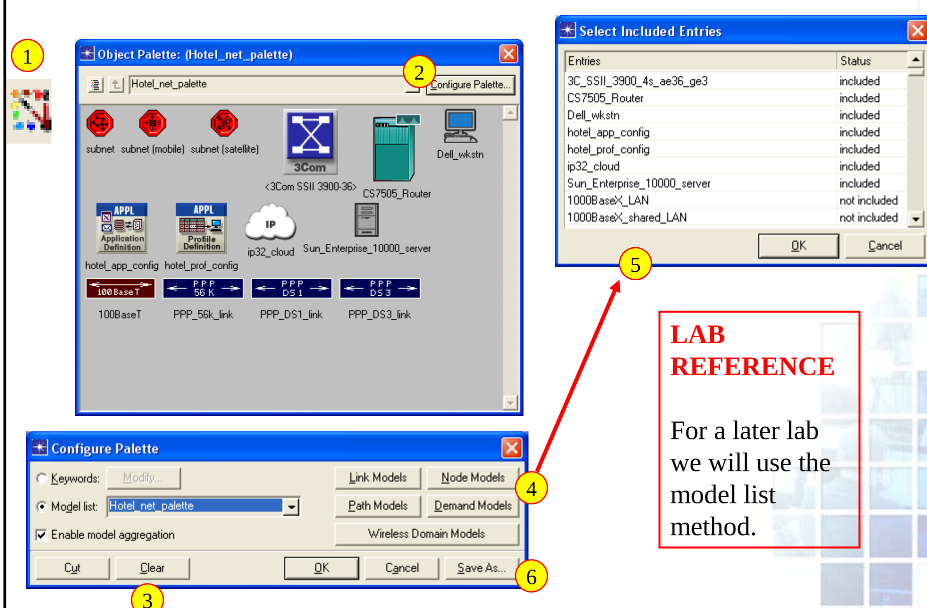
Customizing Palettes

- Keyword
- Model List
- Startup Wizard



OPNET Training

Creating a New Palette - Model List Method



LAB REFERENCE

For a later lab we will use the model list method.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

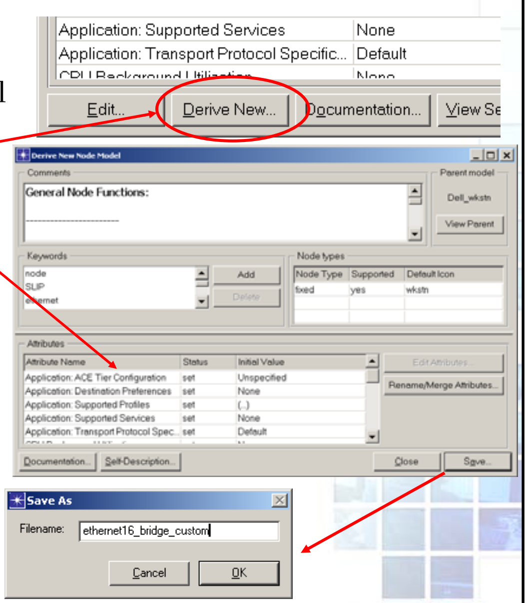
© Copyright 2007 OPNET Technologies, Inc.

Creating New Devices

- Although OPNET Modeler includes hundreds of standard and vendor devices, you may need to customize models.
- There are several ways to create new devices or derive models from the existing database to custom specifications.
  - Two methods discussed here to change an existing model
    - Deriving a new model
    - Using Device Creator
  - Modeling new devices and protocols discussed later in course

Deriving New Models

- From the Model Description dialog box of an existing model
- Derive a new model based on the existing standard or vendor model
- Alter the attributes of the new model
- Save it as a new model with a new name



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

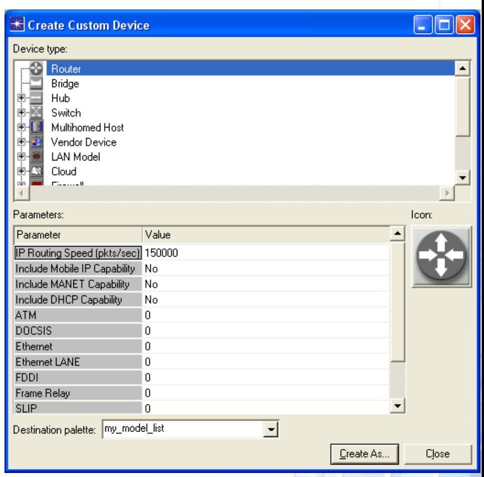
© Copyright 2007 OPNET Technologies, Inc.

Advantages of Deriving Models

- Makes it easy for you to configure and integrate models for specialized needs.
- Allows you to benefit from past work by basing new, specialized models on previously created ones.
- Gives you benefit of control over certain aspects of a new model while preserving its essential structure.
- Allows creation of a new device that has preset and possibly hidden attributes.

Device Creator

- Create Custom Device Model
  - –Provides the ability to automatically create a particular device with a specific configuration.
  - –Typical devices include routers, switches, hubs, bridges, vendor specific devices, and LAN models.
- Under "Topology" pull-down menu: "Create Custom Device Model…"
- Advantages
  - –Allows you to select any number of interfaces and protocols that a device will need to support.
  - –Provides you more control over memory efficiency, reducing simulation run-time.



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Derive New vs. Device Creator

- Both start with an existing model
- Using "Derive New" is for creating a different device with different attribute settings, like the configurations of protocols
- Using the Device Creator is for creating a different device by customizing the number of ports or interfaces it has

OPNET Training

Object Attributes

- All Modeler objects have attributes
- Attributes define the object and control its behavior
  - All devices have a "model" attribute – this is the filename of the object
  - Attribute values may vary between objects of the same model
    - Example: Two routers of the same model may have different routing parameters, two demands may represent different amounts of traffic
  - Attributes may have sub-attributes
- To view or change an object's attributes
  - Right-click on the object
  - Select "Edit Attributes" or "Edit Attributes (Advanced)"

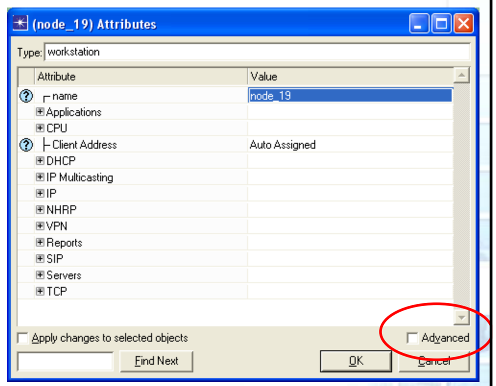
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Object Attributes (cont.)

- All objects have attributes that control aspects of their behavior.
- Attributes may vary from one model to the next.
- Attribute values may vary between objects of the same model type.
- Right click on an object and select "Edit Attributes" to view or change its attributes
- Click the "Advanced" checkbox to see built-in attributes
  - –Position on map
  - –Model type
  - –Etc.



OPNET Training

Selecting Objects

- Select objects to modify, cut, delete, move, or copy them
- Several methods to select multiple objects:
  - –Shift-click or Control-click
  - –Right-click and choose **Select Similar Nodes/Links/Demands**
    - Selects all objects with the same model name
  - **–Edit / Select Objects**
    - Advanced object selection used to select objects with certain attribute values
    - Excellent way to explore configuration of an existing network model
  - **–Edit / Find Node/Link**
    - Select objects based on object name
  - –Draw a selection box around a group of objects
- Click on empty workspace to deselect all objects
- You can save and recall a selection set from the "Edit" menu

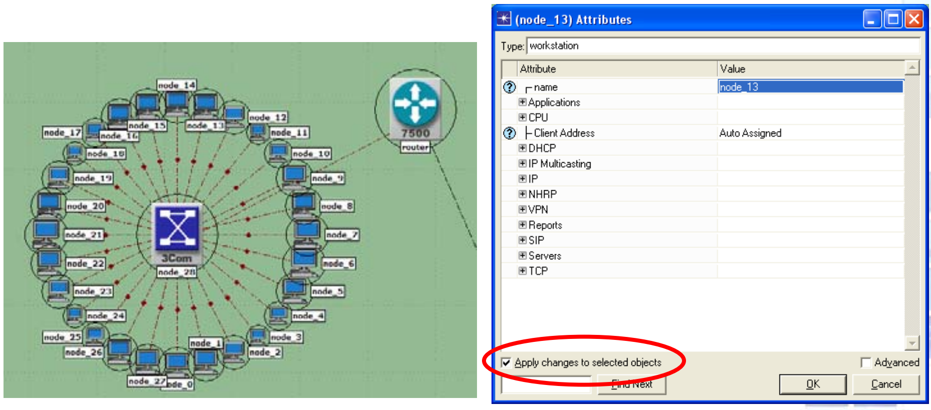
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Group Attribute Assignment

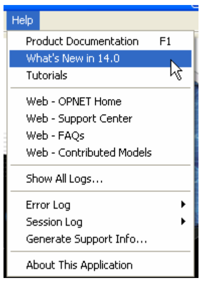
- A group of objects can have their attributes assigned simultaneously by using the “Apply Changes to Selected Objects” button.
- Note that selected objects that do not contain the attribute being changed will not be affected by the group assignment.



OPNET Training

Where to Get Help

- To view the manual, choose Product Documentation from the Help menu.
- Model help is accessed by right-clicking icons in the object palette or by right clicking objects in the Project workspace and selecting “View Node Description.”
- Click the question mark in attribute dialog boxes to view the description of that attribute.
- Tool Tips: place your cursor over any object and wait one second to get a brief description of that object.

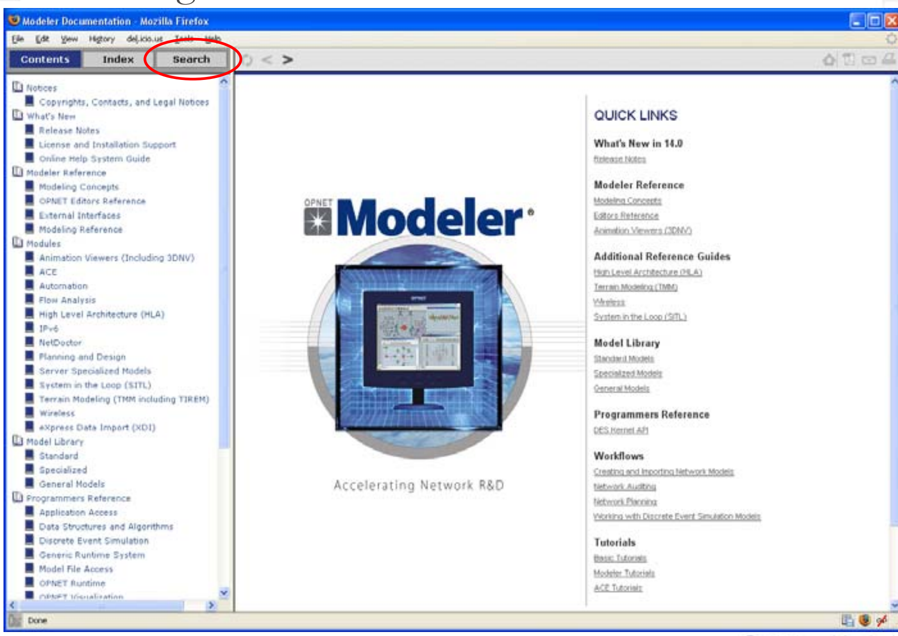


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Searching Product Documentation



OPNET Training

Lab: Using the Project Editor

- Purpose:
  - –"Play" with features of the Project Editor
  - –Use the "Device Creator" and model derivation to create new node models
  - –Modify attributes' values in order to model non-default behavior
  - –Search the product documentation

OPNET Training

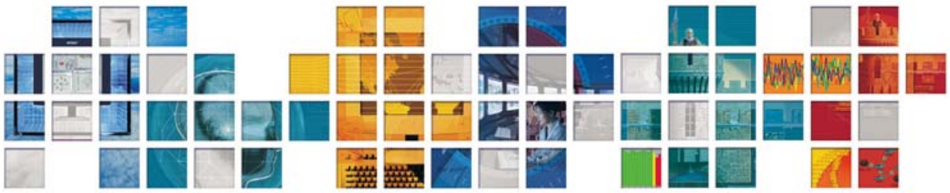
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Technologies, Inc.

Making Networks and Applications Perform™

DES Concept – Choosing Statistics

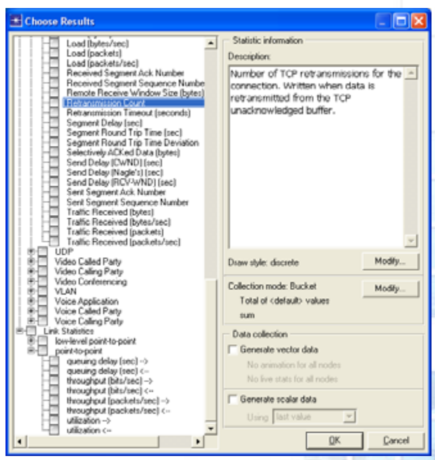


OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Choosing Statistics - Mechanics

- Choose statistics to collect:
  - –DES / Choose Individual Statistics
  - –List of statistics appears
- Types of statistics:
  - –Global: relate to network as a whole
  - –Node: collected on individual nodes
  - –Link: collected on individual links
- Common statistics:
  - –Global: application response times
  - –Node: delay, delay variation
  - –Link: utilization, throughput, queuing delay



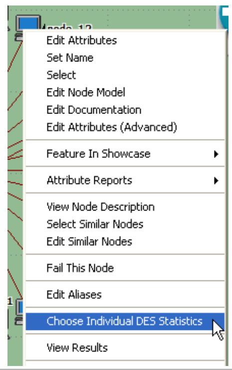
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

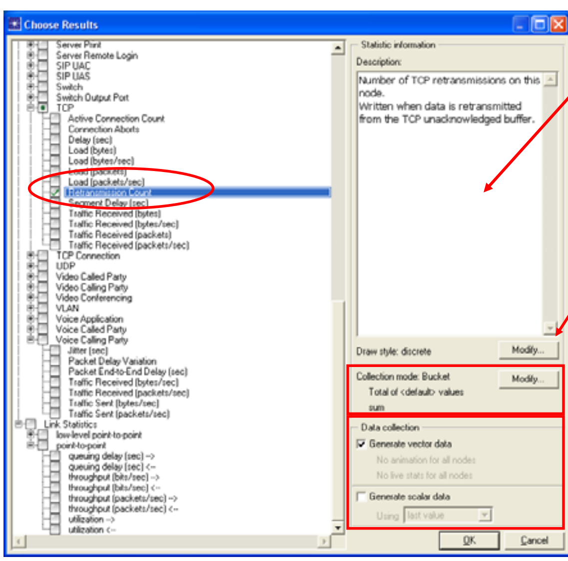
Choosing Statistics - Mechanics

- Collect statistics on a single node or link
  - –Right-click the object and **Choose Individual DES Statistics**
  - –This option is good choice for models with large number of nodes or links – cuts down on memory required for simulation
  - –Good for getting more detailed information on certain devices/links



OPNET Training

Anatomy of the "Choose Results" Dialog Box



Statistic description

Draw style (how the data is displayed in a graph)

Collection mode: How the data points are collected while the simulation is running

Vector vs. Scalar: data discussed later

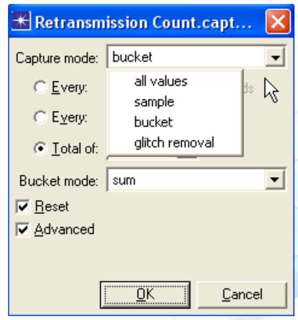
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Statistic Collection Modes

- There are four types of statistic collection modes
  - –All values mode: Every data point is collected from a statistic.
  - –Sample mode: The data is collected according to a user-defined time interval or sample count. For example, you could specify that data be collected
    - •Every 10th simulation second
    - •Every 25th data point.
  - –Bucket mode: All the data points are collected over the time interval or sample count and are processed according to a user-defined parameter-- max, min, sum, count, sample average or time average. (This is the default mode for most statistics)
  - –Glitch removal: Ignore repeated values at the same simulation time



Understanding Statistics

- To run simulations with useful results, it is essential to define the goals of the simulation and to understand the statistics that are needed to measure those goals.
- Browse available statistics and view their definitions.
- Understand the default collection mode to help interpret results.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Choosing Statistics - Advanced

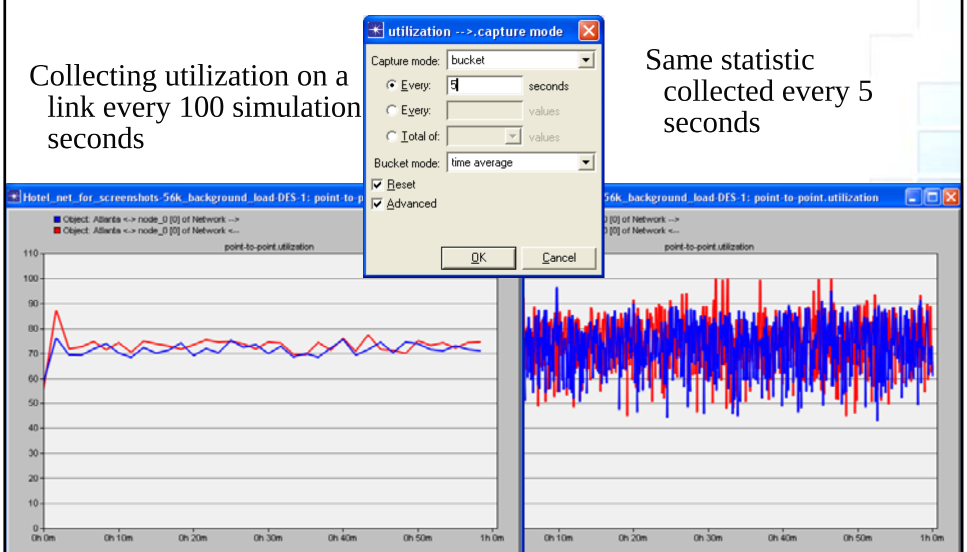
- Optional Step: Change statistic collection mode
  - –Default is bucket mode: collects up to 100 data points per simulation
  - –Simulation time divided into 100 segments
  - –Average value during each segment becomes one point on graph
- Solutions when results are not granular enough:
  - –Change collection mode
  - –Increase number of data points collected per statistic
  - –Shorten simulation time

OPNET Training

Modifying Granularity By Modifying Collection Mode

Collecting utilization on a link every 100 simulation seconds

Same statistic collected every 5 seconds

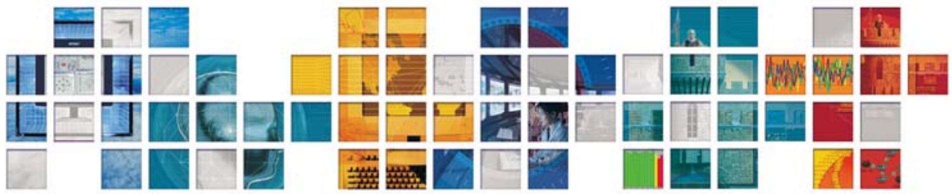


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

DES Concept – Running Simulations

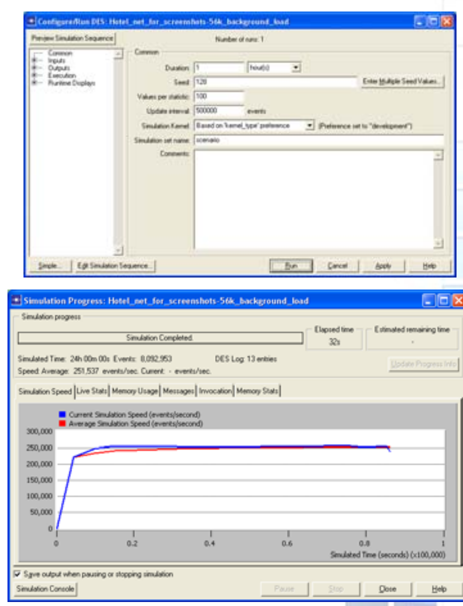


OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Simulation Mechanics

- Run simulation
  - –**DES / Configure/Run Discrete Event Simulation**
  - –Set simulation options
  - –Click **Run**
- While simulation is running, view stats on events processed per second and amount of RAM used by simulation
- After simulation completes, check simulation log for errors
  - –DES / Open DES Log



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

DES Log

- Contains errors generated during simulation
- Contains information about special events during simulation
- Your code can write to DES log – good way to output debugging information
- Always check DES log for errors before trusting results

OPNET Training

DES Log (cont.)

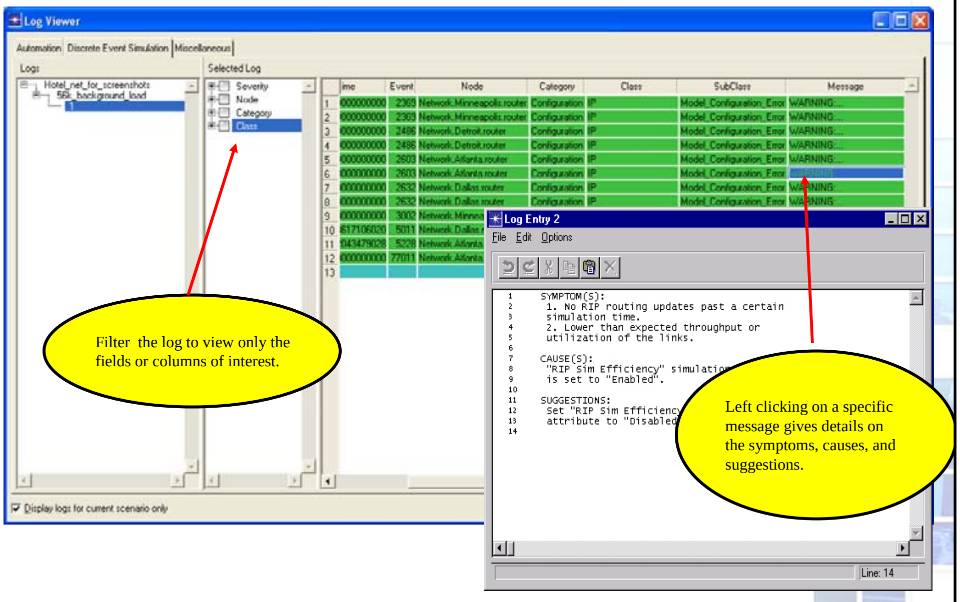
- To view the DES log:
  - –Right-click in the Project Editor workspace and choose "Open DES Log" from the pop-up menu. This will provide information about the entire simulation.
  - –Right click on a node or link and choose "Open DES Log". This will provide information from the simulation log that applies **only to this object**.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

DES Log (cont.)



Simulation Mechanics - Simulation Options

- Duration
  - –Time to simulate in the network model
  - –May be shorter or longer than real time
  - –Less discrete traffic = shorter real time for simulation to complete
- Global attributes
  - –Protocol specific
  - –Control behavior of every applicable object
  - –Sim efficiency modes: disable these for failure / recovery studies
  - –Tracer packet options: determine how quickly flows reroute in failure case
- Values Per Statistic
  - –Increase number to increase granularity of statistics that use default bucket mode

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

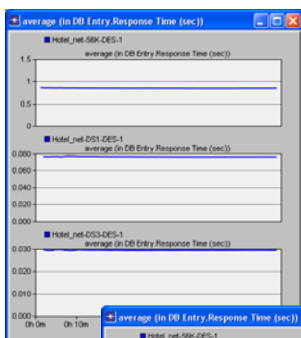
DES Concept – Viewing Results

- If simulation log reveals no errors that invalidate results, view results graphs on statistics you collected
  - –Right-click on workspace / View Results
- View list of node/link statistics in order from highest to lowest
  - –Ex: View most utilized links
  - –Right-click on workspace / Find Top Results
- Compare results between multiple scenarios on same graph

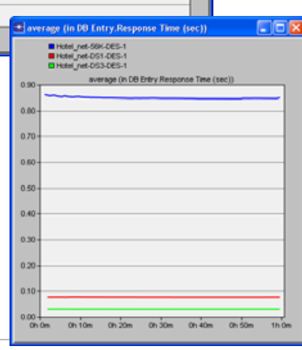
OPNET Training

Results Options

- Stacked Statistics



- Overlaid Statistics



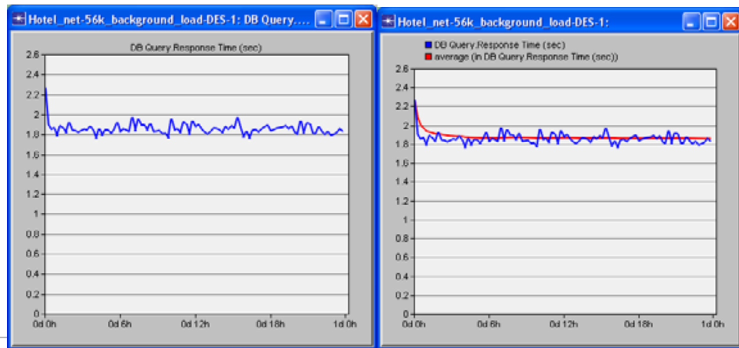
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

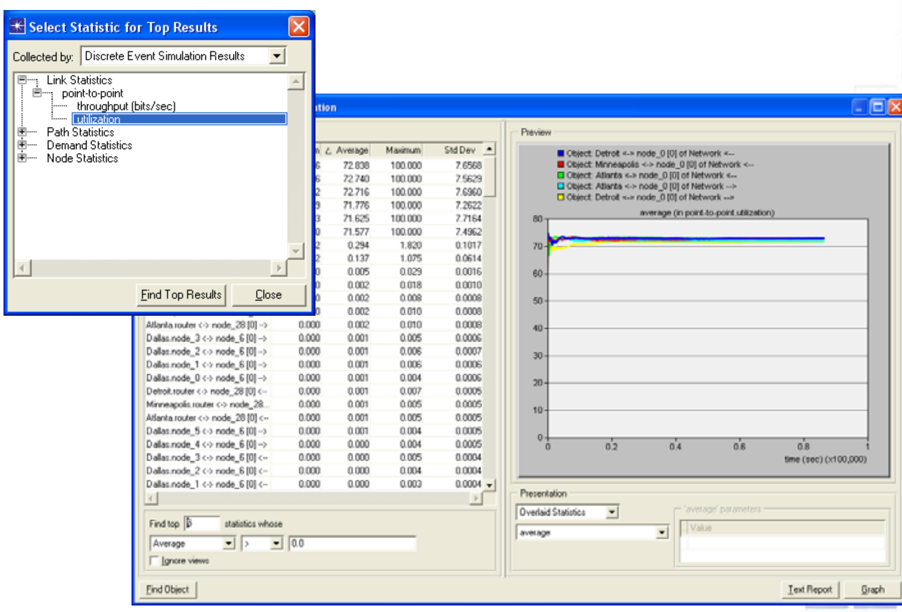
Results Options

- Filters:
  - –As is: every data point collected during sim
  - –Average: average of several data points together
- Use the **Add** button in the lower-right corner of the “View Results” dialog box to add another stat to an existing panel
  - –Ex: Add the **average** response time to a graph that shows **As is** response time



OPNET Training

Find Top Statistics



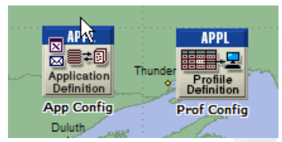
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Lab: Goals

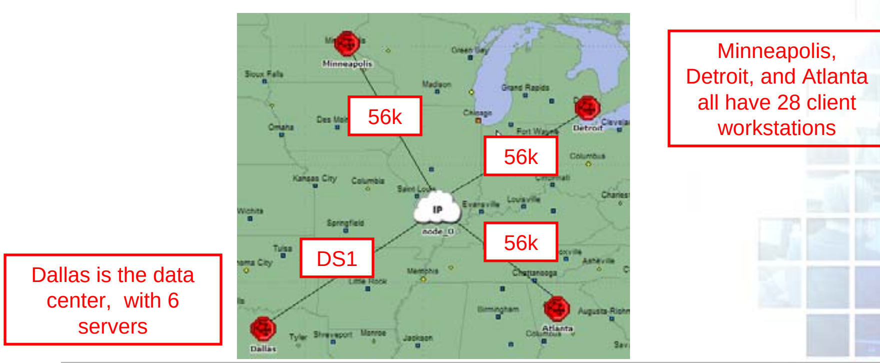
- Build your first network
  - –Review for anyone who has gone through the tutorials or IT Guru Quickstart
  - –Teaches fundamentals of modeling workflow
- Conceptual Goals in Project Editor
  - –Building Topologies
  - –Choosing Results
  - –Running Simulations
  - –Viewing Results
- Traffic Preconfigured in this lab
  - –The gray boxes you place in the workspace during the lab contain preconfigured application traffic. We will discuss application configuration later in the course.



83 OPNET Training

Lab: Building Your First Network

- Lansing Hotel Reservation Services is a hotel reservation company headquartered in Dallas, Texas.
- Lansing employees are experiencing high delays using the company’s proprietary hotel reservation application, and customers are threatening to use other services if the problem is not fixed.
- Your task is to identify the problem and propose a solution.



84 OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Summary

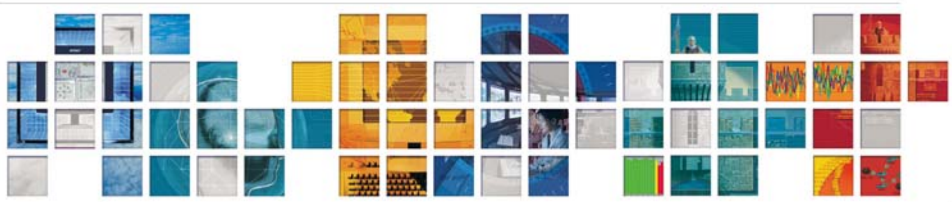
- One project can contain many scenarios
- The Startup Wizard can be used to quickly and easily create new scenarios.
- The Project workflow consists of:
  - –Creating a project
  - –Creating baseline scenario
  - –Choosing results to be collected
  - –Running simulation
  - –Viewing results
  - –Duplicating scenario
  - –Running simulation
  - –Comparing results

OPNET Training

OPNET Technologies, Inc.

Making Networks and Applications Perform™

Modeling Framework



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

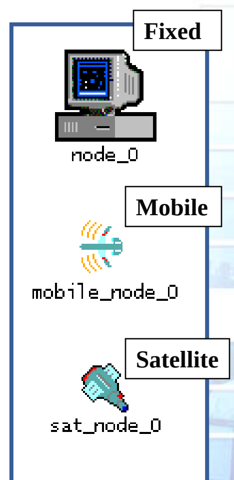
© Copyright 2007 OPNET Technologies, Inc.

Agenda

- Conceptual Goals
  - –Objects available in the modeling domains
  - –Data transfer between objects in a simulation
  - –Network, node, and process models
  - –Object attributes
  - –Object naming paradigm
  - –Role of packets in a simulation

Network Objects - Nodes

- Node objects are represented by icons. Different icons may represent the same underlying object. Icons shown are the default icons.
- A fixed node remains stationary during a simulation.
- A mobile node changes position during a simulation, following an assigned trajectory or using vector based mobility (ground speed, bearing, and ascent rate). Trajectories can easily be created graphically in Modeler, or by ASCII text files.
- A satellite node changes position during a simulation, following an assigned orbit.



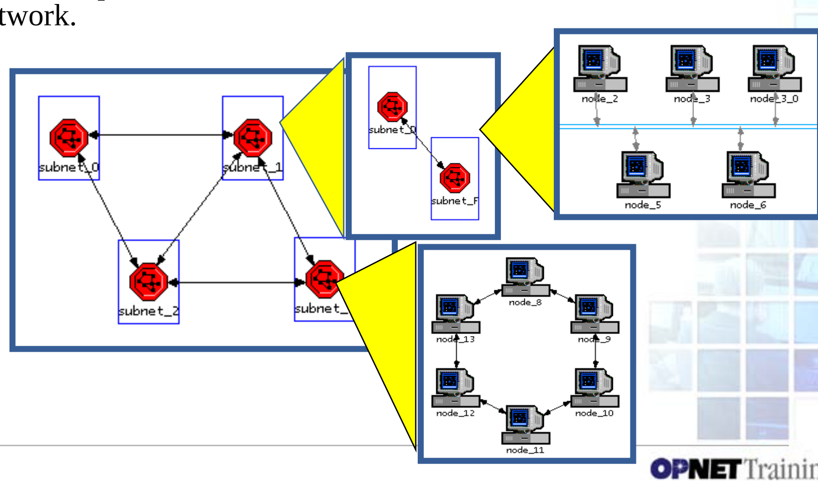
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Network Objects - Subnets

- A subnet object abstracts network components specified within it into a single object.
- Subnets represent identical constructs in an actual network.

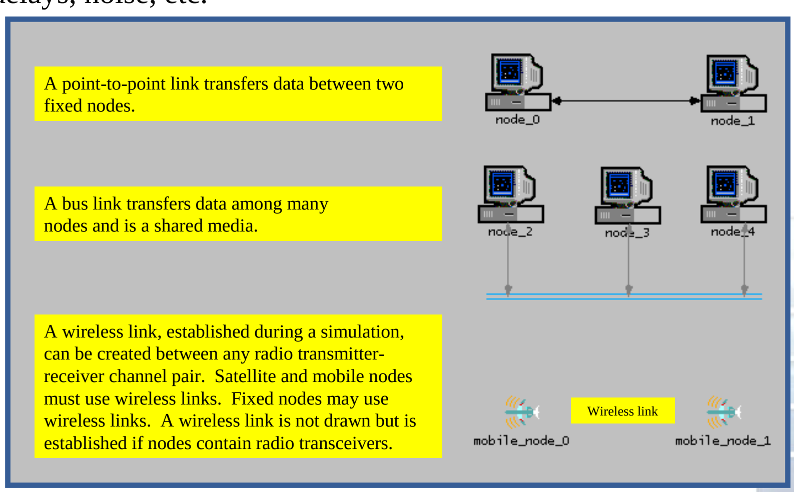
Subnetwork



OPNET Training

Network Objects - Links

- Link objects model physical layer effects between nodes, such as delays, noise, etc.



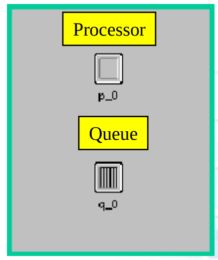
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Node Objects - Modules

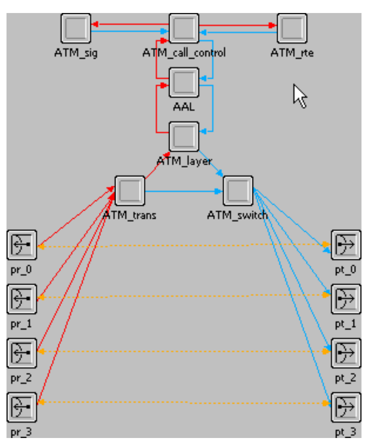
- Modules are the basic building blocks of node models. Modules include processors, queues, transceivers, and generators.
- Processors are the primary general purpose building blocks of node models, and are fully programmable.
- Queues offer all the functionality of processors, and can also buffer and manage a collection of data packets.



OPNET Training

Transmitters and Receivers

- Transmitters are the outbound interfaces between objects inside a node and communication links outside it.
- Receivers are the inbound interface.



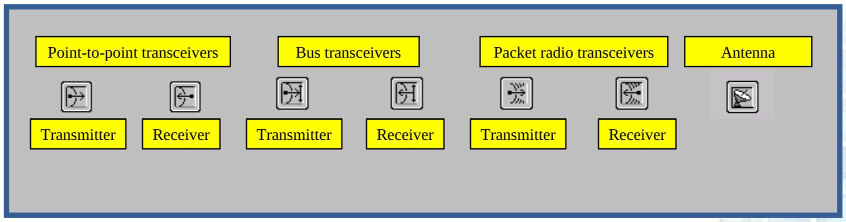
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Transmitters and Receivers

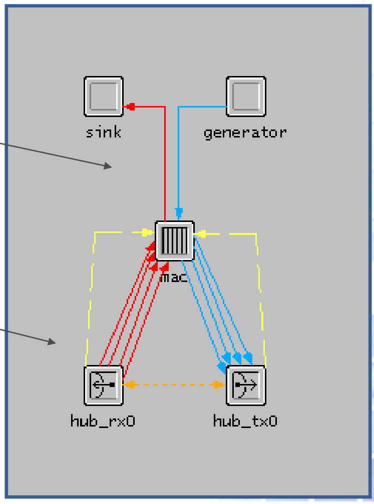
- Three types of transmitter and receiver modules correspond to different models of communication links.
- Antennas may be used with radio transceivers to specify antenna properties.



OPNET Training

Module Connections

- Packet streams carry data packets from a source to a destination module.
- Statistic wires carry a single data value from a source to a destination module. In this case, `hub_rx0` might report a packet reception to `mac`.



OPNET Training

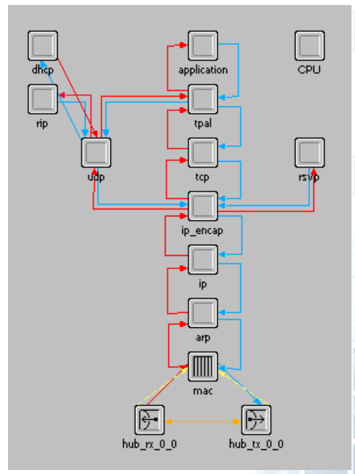
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Sample Node Model

- Node models support
  - –Layering of protocol functions
  - –Dynamic inter-module monitoring
  - –Arbitrary node architectures
  - –Definition of node classes through attribute promotion

ethernet_wkstn_adv Node Model

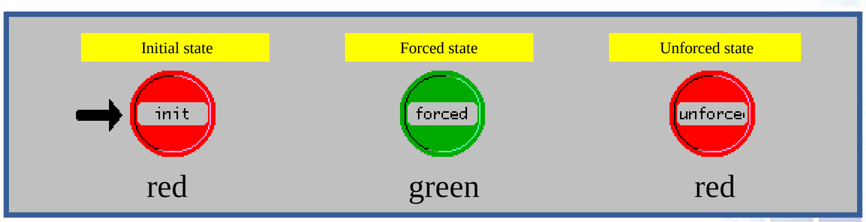


95

OPNET Training

Process Model Objects - States

- The *initial state* is the place where execution begins in a process.
- A *forced state* does not allow a pause during the process.
- An *unforced state* allows a pause during the process.
- Later chapters will fully discuss the differences between these types of states.



96

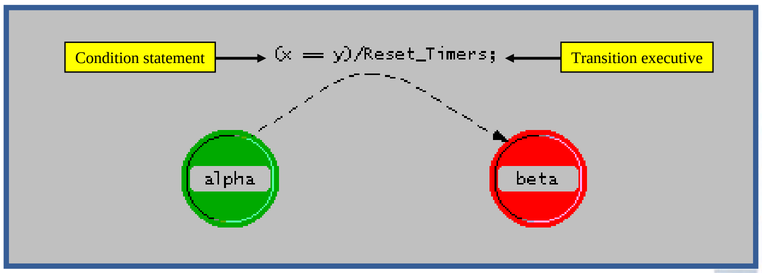
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

State Connections - Transitions

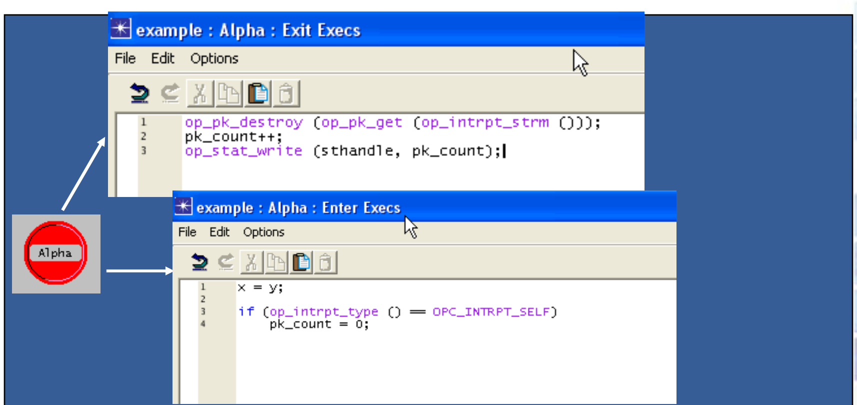
- Transitions describe the possible movement of a process from state to state and the conditions allowing such a change.
- Exactly one condition must evaluate to true.
- If the condition statement *(x == y)* is true, the transition executive *(Reset_Timers;)* is invoked.



OPNET Training

Executive Blocks

- Each state has two executive blocks
  - –*Enter executives* are invoked on entering a state.
  - –*Exit executives* are invoked before exiting a state.



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Kernel Procedures - Introduction

- KPs are pre-written functions that abstract difficult, tedious, or common operations. KPs free you from addressing memory management, data structure, handling event processing, etc.
- All KPs begin with prefix “op_”.
- KPs focus on communication modeling.

OPNET Training

Kernel Procedures

- Sample of commonly used KPs

Packet Package:
op_pk_create ()
op_pk_create_fmt ()
op_pk_copy ()
op_pk_get ()
op_pk_total_size_get ()
op_pk_nfd_set ()
op_pk_nfd_get ()
op_pk_send ()
op_pk_send_delayed ()
op_pk_destroy ()

Subq Package:
op_subq_pk_insert ()
op_subq_pk_remove ()

ID, Topo and Internal Model Access Packages:
op_id_self ()
op_topo_parent ()
op_topo_child ()
op_ima_obj_attr_get ()

Interrupt Package:
op_intrpt_schedule_self ()
op_intrpt_type ()
op_intrpt_strm ()
op_intrpt_code ()

Stat Package:
op_stat_reg ()
op_stat_write ()
op_stat_local_read ()
op_stat_scalar_write ()

Distribution Package:
op_dist_load ()
op_dist_outcome ()

Simulation and Event Packages:
op_ev_cancel ()
op_sim_time ()

- Naming convention for Kernel Procedures -
  - op_<family name describing object >_<action>
  - When using process model editor, press ctrl-H for list of commonly used KPs.

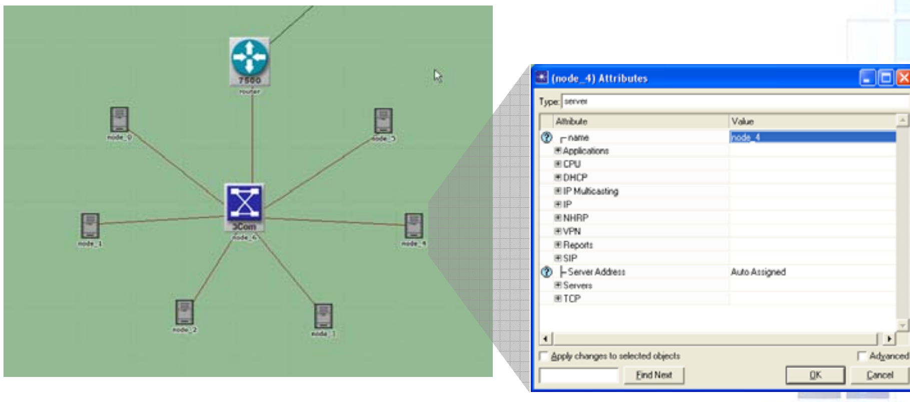
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

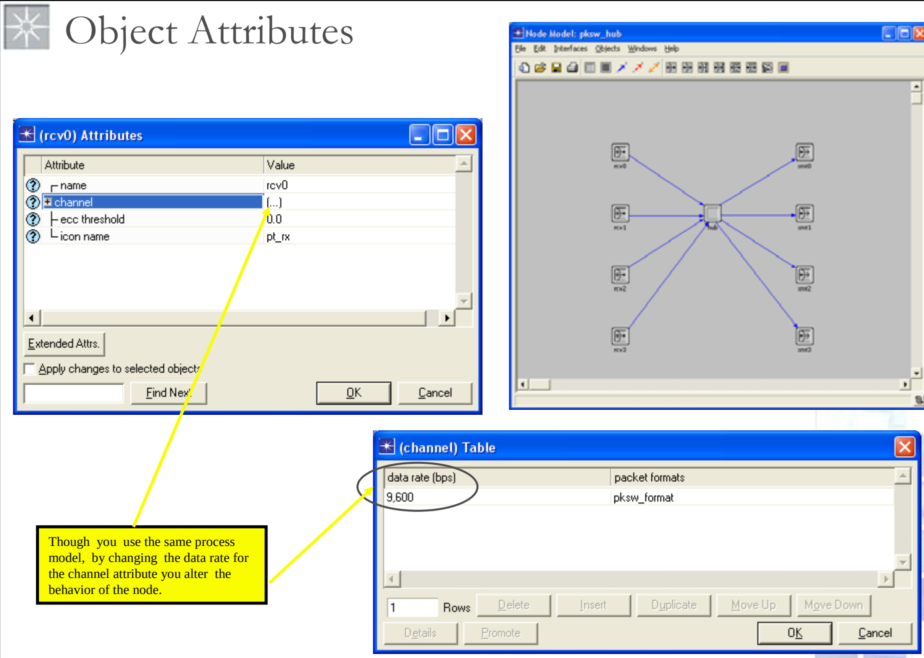
Object Attributes

- Attributes are parameters of an object that can configure its behavior.
- Attributes are dynamically changeable during simulation.
- Processes have access to all object attributes.
- Different attribute values allow objects of the same type to behave differently.



OPNET Training

Object Attributes



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Assigning Attribute Values

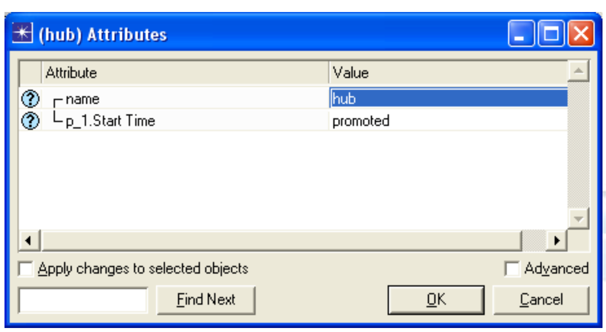
- You can assign attribute values by right-clicking on an object and selecting or specifying the attribute value.
- Attributes are of a certain type. Commonly used types are listed.

Type	Definition
Integer	Whole numbers: storage capacities; transmission window size
Double	Decimal numbers: processing speeds; timer values
String	General text info: statistic names, object names, options
Toggle	True/false condition: status flags, semaphores
Typed file	User defined file: routing tables, address mappings, script file
Compound	Nested, complex data: routing table, circuit table, subqueues

OPNET Training

Promoting Attribute Values

- You can "promote" an attribute. This means that you assign a value at a higher hierarchical level.
- Passing control of a lower-level object to a higher level provides more flexibility in how objects are used.
- You can leave an attribute unspecified at even the network level, and assign a value at run time.



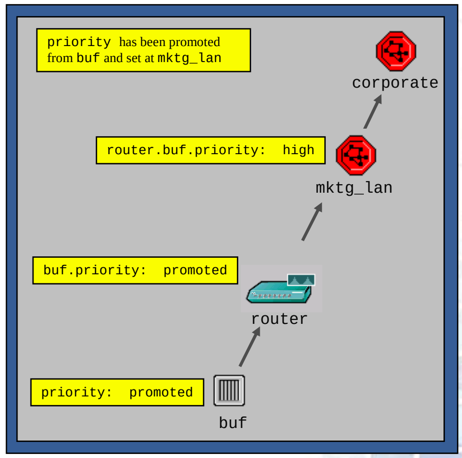
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Promoting Attributes Example

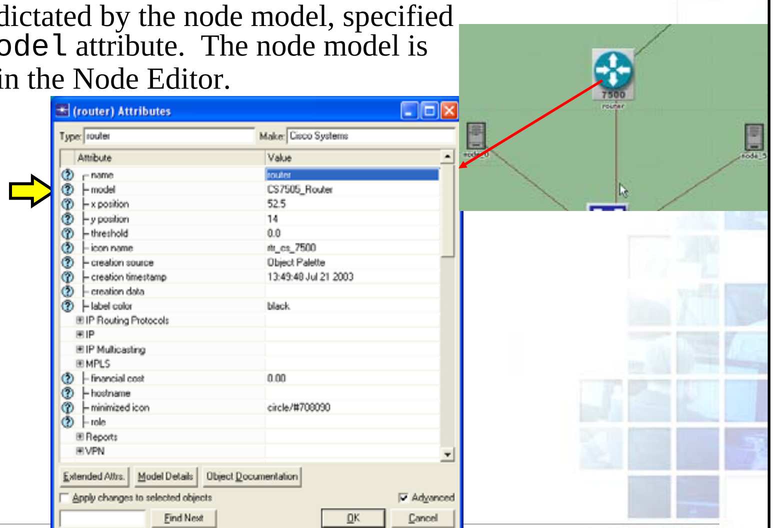
- When an attribute assignment is made, promotion stops. An attribute value was assigned at `mktg_lan,` so the attribute does not appear in the object `corporate`.
- Attribute names are used as prefixes at each new level of the object hierarchy.



OPNET Training

Model Hierarchy

- The internal structure and behavior of each node is dictated by the node model, specified in the `model` attribute. The node model is created in the Node Editor.



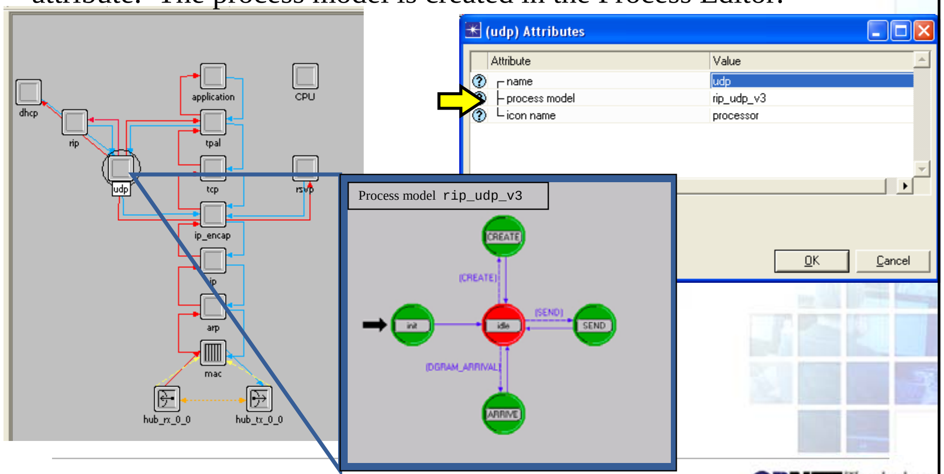
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Model Hierarchy

- The internal structure and behavior of each processor and queue is dictated by the process model specified in the `process model` attribute. The process model is created in the Process Editor.



OPNET Training

Object Naming

- Each object has a unique name that defines its place in the hierarchy.

- Format of name is:

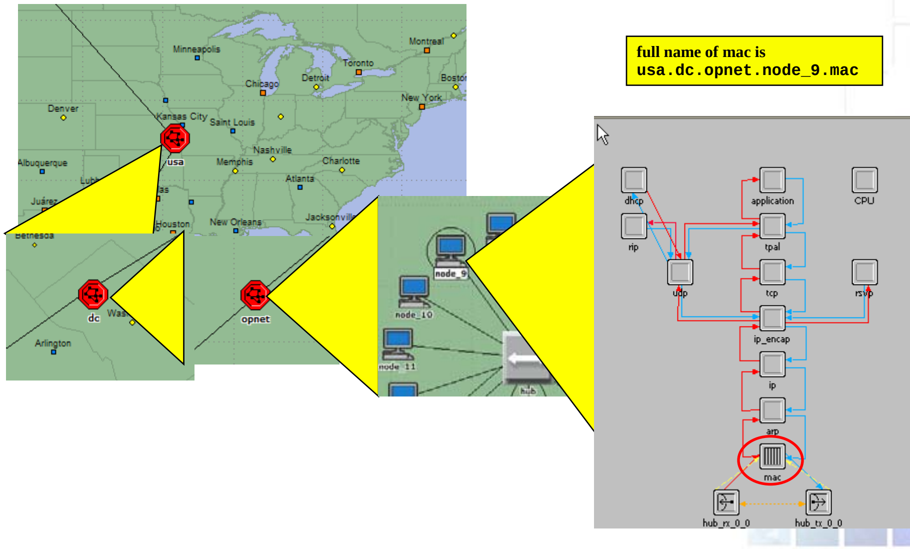
  `network_type.subneta.subnetb...subnetz.node_name.object_name`

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Object Naming



OPNET Training

Data Flow Among Objects

- Packets are the basic unit of information exchange in Modeler simulations.
- Information is exchanged among different objects via various communication mechanisms:
  - –Node to node: *Links*
  - –Module to module: *Packet streams and statistic wires*
  - –State to state: *Transitions*

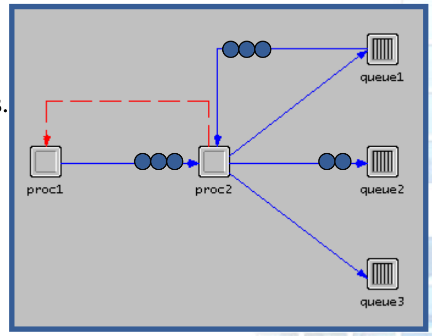
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Packets

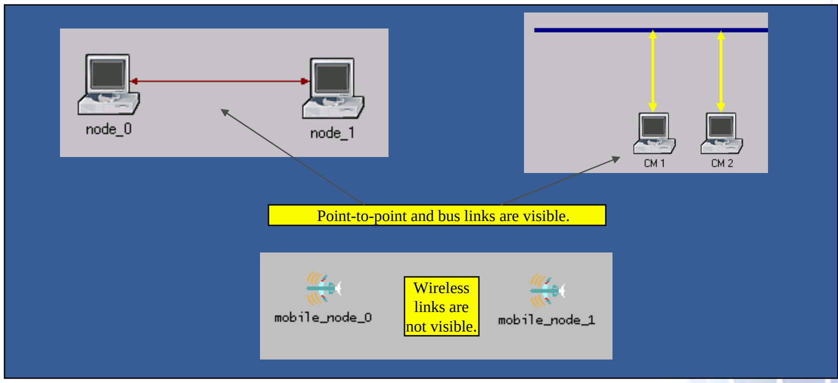
- Packets flow between modules via packet streams. At the end of each stream is a built-in packet buffer.
- Packets are
  - –The information-carrying entity that circulates among system components.
  - –General data structures, organized into fields of information you define.
  - –Dynamic simulation entities that are created and destroyed as the simulation progresses.
- A single system may rely on multiple types of packets with different formats.
- A statistic wire (statwire) communicates a single value that may cause an interrupt to occur at the destination module.



OPNET Training

Communication Mechanisms - Links

- In the network domain, packets flow between nodes via links.



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Summary

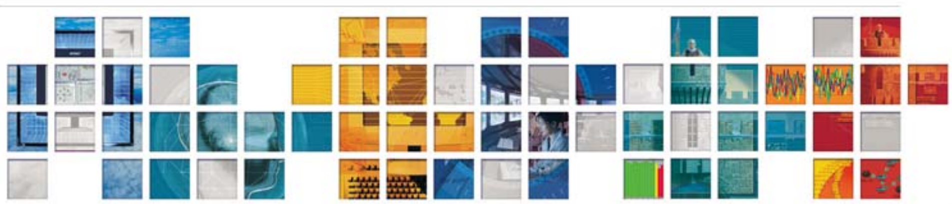
- **Network Objects**: Nodes (fixed, mobile, satellite), Subnets, and Links (point-to-point, bus, wireless).
- **Node Objects**: Modules (Processors, Queues, Transmitters, Receivers, Antennas) and Connections (Packet Streams and Statistic Wires).
- **Process Model Objects**: States (initial, forced, unforced) and Transitions.
- **Kernel Procedures**: Pre-written functions that abstract communication modeling operations.
- **Object Attributes**: Dynamic parameters that can configure the behavior of an object.
- **Packets** : Basic units of information exchange in OPNET simulation.

OPNET Training

OPNET Technologies, Inc.

Making Networks and Applications Perform™

Other Editors



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

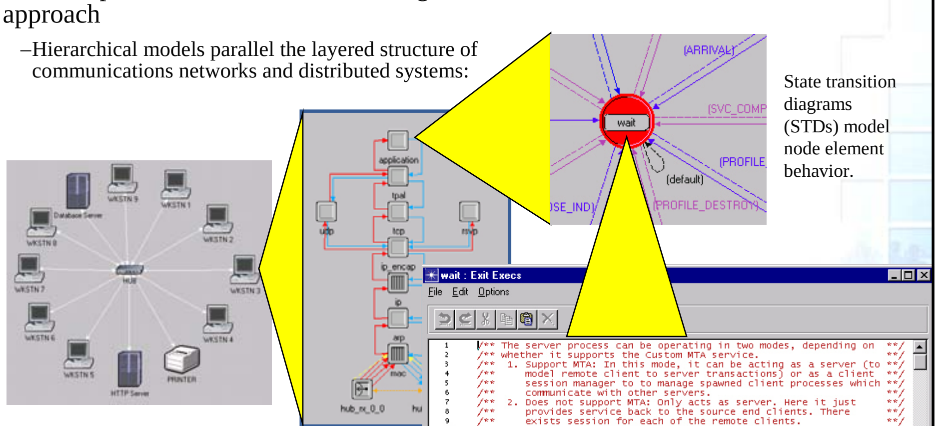
© Copyright 2007 OPNET Technologies, Inc.

Agenda

- Conceptual Goals
  - –Modeler’s architecture and philosophy.
  - –Brief look at all other Editors
  - –Hierarchy of modeling

Modeling Approach

- Modeler provides a structured modeling approach
  - –Hierarchical models parallel the layered structure of communications networks and distributed systems:



State transition diagrams (STDs) model node element behavior.

Network models consist of nodes and links.

Node models represent data flow between functional blocks.

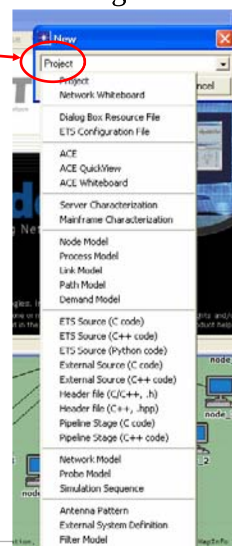
Each state of an STD can contain general logic expressed in C.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Modeler Editors

- A variety of editors allow you to view and configure different layers of the network structure.



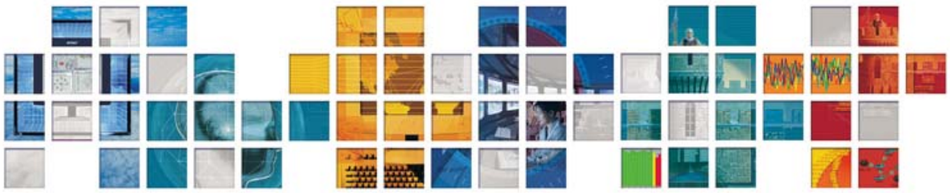
Lab: Using Other Editors

- Purpose: Brief introduction to some of the editors available in Modeler

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

General Simulation Methodology



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

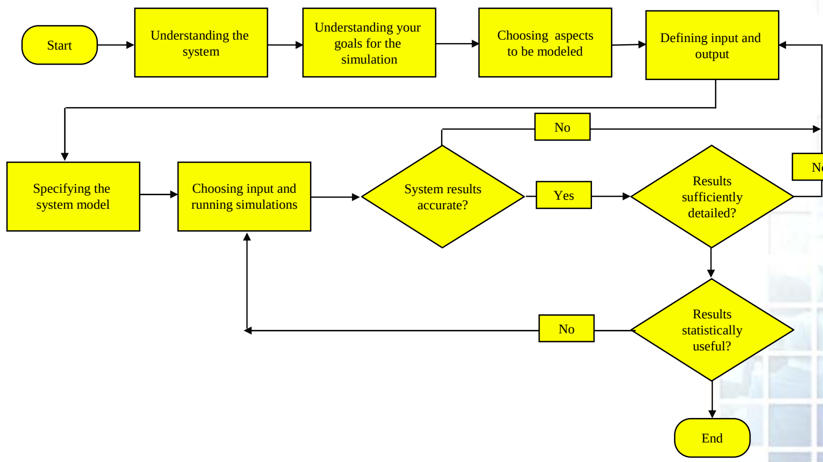
Agenda

- Topics
  - –Understanding what you are modeling
  - –Granularity
  - –Defining Data
  - –Results
- Conceptual Goals
  - –Use basic methodology for developing a network simulation study.
  - –Plan simulation studies using a consistent method.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Flow Diagram: A Simulation Study



OPNET Training

Understanding the Proposed System

- You must understand the proposed system to model it accurately.
- The model cannot be more accurate than your understanding.
- Recommendation: obtain a written specification of the system you will be modeling.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Understanding Your Goals

- What general questions do you want the simulation to answer?
- Sample questions:
  - –How much will replacing our Ethernet hub with an FDDI hub increase throughput?
  - –How much does a customized protocol improve efficiency in my cellular network?
  - –How much will adding 1000 users lengthen response time?

OPNET Training

Choosing Aspects to Be Modeled

- From the general questions, formulate specific research objectives, such as
  - –Quantifying the throughput at a specific receiver.
  - –Measuring the effect of changing a link's transmission rate.
  - –Finding the point where traffic load causes the system to become unstable.
- These objectives determine the "granularity" (level of detail) required in the model.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

What is Granularity?

- Granularity is the level of detail.

- Choose a granularity sufficient to answer all questions. You have:
  - –Coarse granularity (minimum detail) allows faster execution and simpler development.
  - –Fine granularity (more detail) allows greater precision, but simulations run slower and may require more complex development.

Choosing the Granularity Required

- Choose the coarsest granularity that still answers all questions accurately.

- Often, some portions of a model require fine granularity, and others much less.

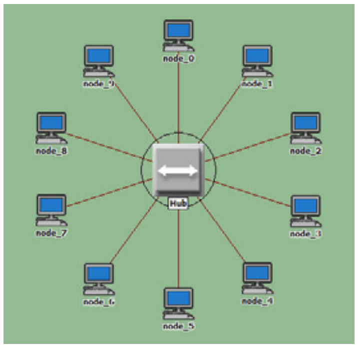
- OPNET is flexible -- it allows you to vary the granularity within a single model.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Granularity - Example 1

- Model and calculate application end-to-end delay (ETE delay) for 10 workstations connected via an Ethernet hub.
- Compare the delay measurements with those obtained using an FDDI hub instead.



OPNET Training

Determining Granularity for Example 1

- Important to model the full data flow of the TCP/IP/MAC layer protocol stacks within each workstation.
- Ethernet and FDDI protocols, especially, must be detailed and exact to precisely model ETE delay.
- Extensive detail produces accurate results for a variety of offered loads.
- Conclusion: This model requires fine granularity.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Granularity - Example 2

- Model the following scenario: 2,000 workstations in Chicago and another 4,000 in New York communicate via an OC12 link. How much will an OC48 link decrease response time?

OPNET Training

Determining Granularity for Example 2

- Abstract this scenario into a simple high-level model:
  - Model all 2,000 Chicago workstations as a single node and all 4,000 New York workstations as a second node.
  - Focus on the performance of the link between the two nodes.
- Conclusion: coarse granularity probably sufficient.
- Observation: modeling full detail might be prohibitively slow to develop and run.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Defining the Data Input and Output

- Choosing input is a critical step.
- If you
  - –Understand the system
  - –Understand your goals
  - –Have defined the specific aspects to be modeled
  - –Have clarified the questions to be addressed

You will be able to choose the proper input.

Defining the Data Input and Output

- Input may refer to a predefined aspect of the model (such as connectivity) or may be more parametric (such as traffic generation rate).
- To study effects on a system, keep most inputs constant and vary one or two over a range.
- Determine which output (throughput, ETE delay, etc.) you need and the best presentation method (graphs, tables, animations, etc.)

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

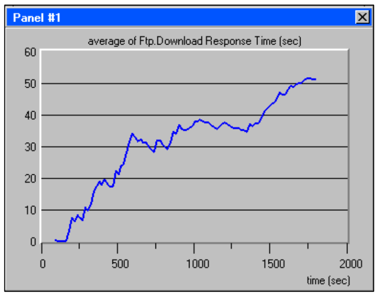
© Copyright 2007 OPNET Technologies, Inc.

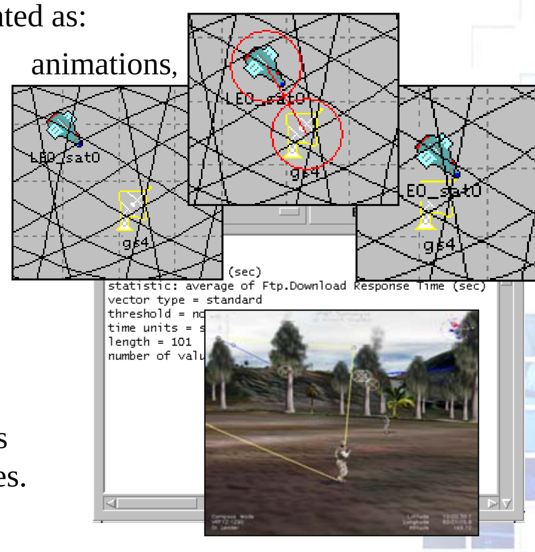
Defining the Data Input and Output

❖ Output may be presented as:

animations,

graphs,





tables, web reports
or numerical values.

OPNET Training

Specifying the System Model

- Simulation software can represent your system in many different ways. To use it effectively
  - –Understand the features your simulation software offers.
  - –Understand how those features can best describe your system.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

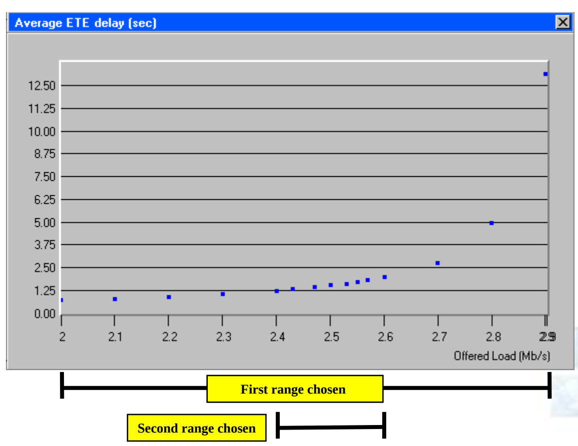
Choosing a Range of Input

- Keep most variables constant except for one or two that you vary over a range.
- Usually it is possible to analytically estimate upper and lower bounds for the range.
- Choose input values that span this range.
- After running initial simulations, you may decide to
  - –Extend the range.
  - –Focus on a narrow range to obtain finer detail.

OPNET Training

Choosing a Range of Input — Example

- First range chosen was 2.0 - 2.9 Mb/s with step 0.1 Mb/s.
- Second range chosen was 2.4 - 2.6 Mb/s with step 0.02 Mb/s.



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

System Results Accurate?

- Output should be "sanity-checked" for accuracy and robustness:
  - –Does all output make sense?
  - –Can it be justified?
  - –Does the model behave appropriately?
- One technique: predict output first, then compare predicted to actual output.

OPNET Training

Results Sufficiently Detailed?

- As needed, either widen the input range to get "the bigger picture" or narrow it to focus on a specific region.
- Should you
  - –Collect more or different statistics?
  - –Change the level of granularity?
  - –Model different aspects of your system?

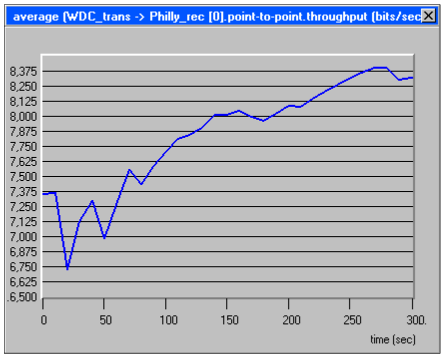
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Results Statistically Useful?

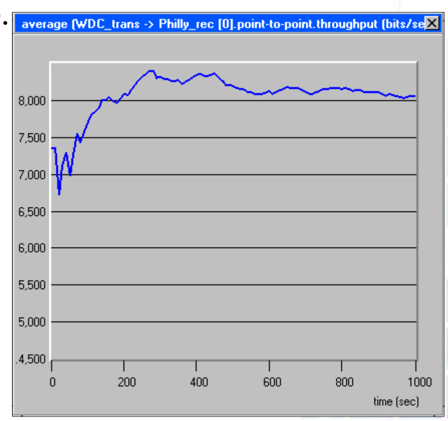
- Is the model operating in steady state?
  - –Expect initial transient behavior, converging to steady state.
  - –Look for rising averages, increasing queue lengths, or other output that is still varying. This might mean that the model is not in steady state.



OPNET Training

Results Statistically Useful?

- Run simulations for a longer period until outputs are stable.
- A model that never reaches steady state indicates an unstable system.
- Look at various statistical measures to ensure that the model is in steady state.



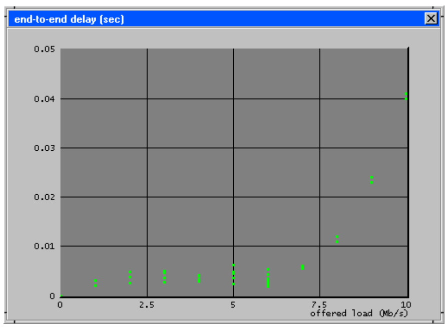
OPNET Training

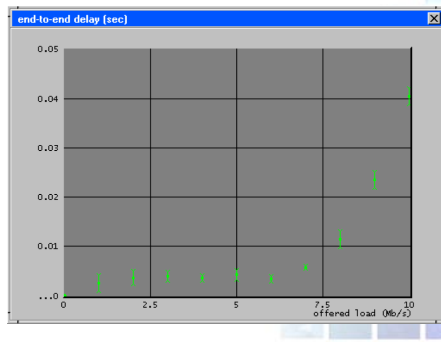
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Results Statistically Useful?

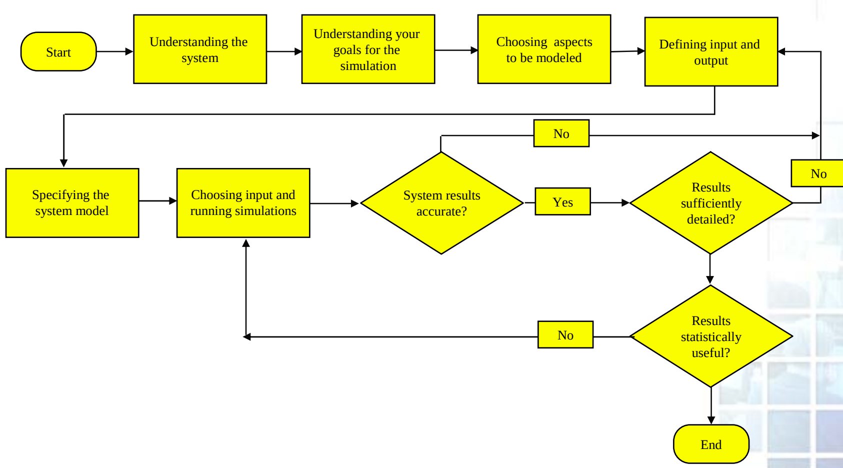
- Have you run enough simulations?
  - –Run numerous simulations, varying the random number seed and generating confidence intervals.
  - –Run simulations until confidence intervals are tight.





OPNET Training

Summary



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Technologies, Inc.

Making Networks and Applications Perform™

Events and Event List Concepts

OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Agenda

- Conceptual Goals
  - –Events
  - –Interrupts
  - –Event-driven simulation
  - –Event list and the simulation time clock
  - –Simulation Kernel
  - –Processes and interrupts

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Event-Driven Simulation

- An event is a request for a particular activity to occur at a certain time.
- OPNET simulations are event-driven. Time, in the simulation, advances when an event occurs.
- A different method might be to sample at regular intervals. Disadvantages are as follows:
  - –Accuracy of results is limited by the sampling resolution.
  - –Simulation is inefficient if nothing happens for long periods.

Event List Concepts

- An OPNET simulation maintains a single global event list.
- All objects access a shared simulation time clock.
- Events are scheduled on the list in time order. The first event on the list is the head.

Head →

Time	Event Type	Module
0.0	Initialize	src.gen
0.0	Initialize	src.rte
4.3	Timer expires	src.gen
4.3	Packet arrives	src.rte

- An event has data associated with it.
- When an event completes it is removed from the list.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Interrupts

- An event becomes an interrupt when it reaches the head of the event list and is delivered by the Simulation Kernel to the designated module.
- Data associated with the event can be obtained by the module when the interrupt occurs.
- Certain modules, processes, and queues can be selected to place initial interrupts on the event list.

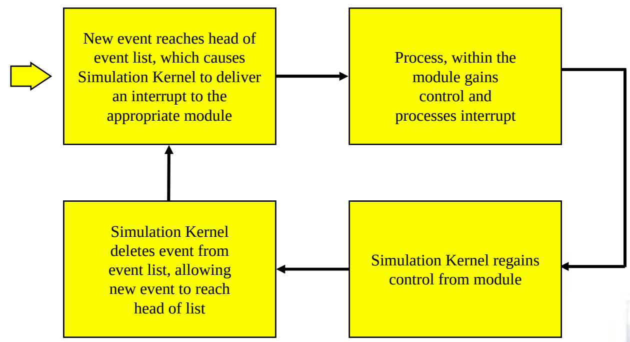
The Simulation Kernel

- An entity, the Simulation Kernel (SK), manages the event list.
- The SK delivers each event, in sequence, to the appropriate module.
- The SK receives requests from modules and inserts new events on the event list.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

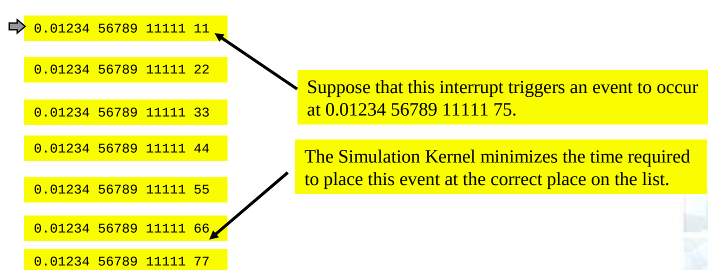
How Does the Event List Work?



OPNET Training

Event List Implementation

- The Simulation Kernel uses a proprietary, efficient algorithm to maintain the event list.
- Event times are expressed as double-precision, floating point numbers and are used to keep the event list sorted.



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Simultaneous Events

- What Happens When Two Events Are Scheduled for the Same Simulation Time?
- The events are actually simulated sequentially, though they appear to occur synchronously according to the time clock.
- The Simulation Kernel uses one of two methods to determine execution order:
  - "Natural order method": The event that reaches the list first is executed first.
  - "Priority factor method":
    - Modules and events are assigned priority factors.
    - Events with a higher priority or originating from a higher priority module are executed before events with a lower priority or from a lower priority module.

Event List Concepts Reviewed

- Some events must be entered on the event list at the start of a simulation.
  - A generator module enters an initial event.
  - A processor or queue module has the `begsim interrupt` attribute enabled.
- An event list typically has a few events – each event spawns another event or two that is placed on the list as the spawning event is deleted.
- The event list is always growing and shrinking.
- An event is pending until executed. A pending event can be cancelled.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Delivery of Interrupts

- When an interrupt is delivered to a module, control passes from the Simulation Kernel to the module.
- If the module is a queue or processor, the interrupt is delivered to the process running within the module.

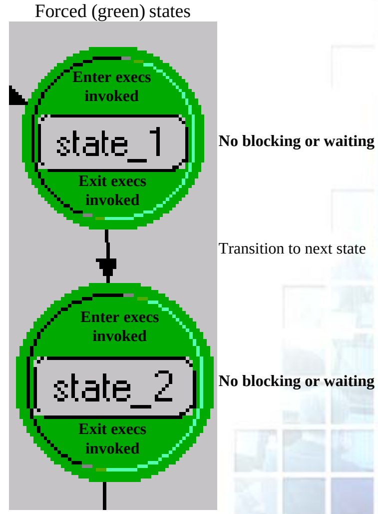
153

OPNET Training

Forced States

- Forced (green) and unforced (red) states differ significantly in execution timing.
- In a forced state, the process:
  - –Invokes the enter executives
  - –Invokes the exit executives
  - –Evaluates all condition statements
  - –If exactly one condition statement evaluates to true, the transition is traversed to the next state.

Forced (green) states



Transition to next state

154

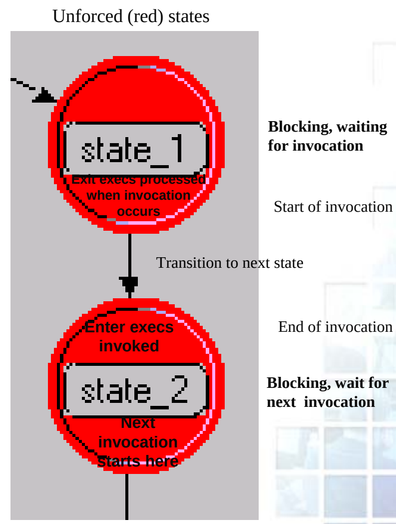
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Unforced States

- In an unforced state, the process
  - –Invokes the enter executives
  - –Places a marker at the middle of the state
  - –Releases control to the Simulation Kernel and becomes idle
  - –Resumes at the marker and processes the exit execs when next invoked



OPNET Training

Transitions Between States

- After completing the exit executives, the process evaluates the condition statements of all departing transitions from the state.
- One and only one condition statement must evaluate to true.
- The process traverses the transition associated with this condition statement.
- A transition with condition = “default” is true if and only if no other conditions are true.
- A transition with no condition set is termed *unconditional* and is always true.

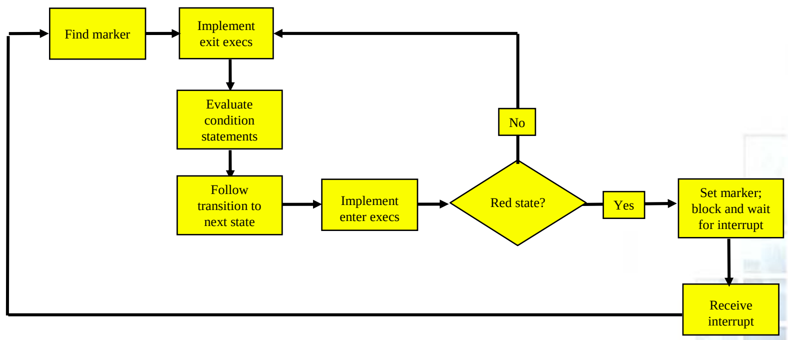
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

How a Process Handles an Interrupt

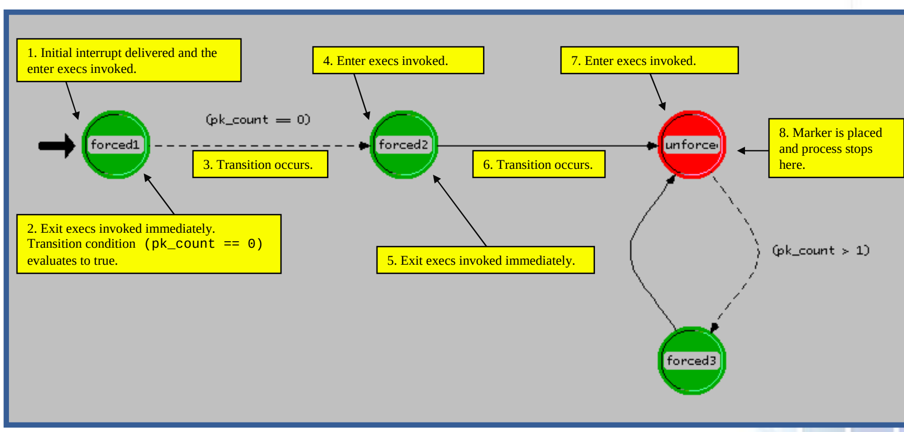
- Flow diagram showing how a process handles an interrupt **(except the initial interrupt)**



OPNET Training

Process Model Example

- Model with three forced states and one unforced state



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

The Simulation Kernel and Processes

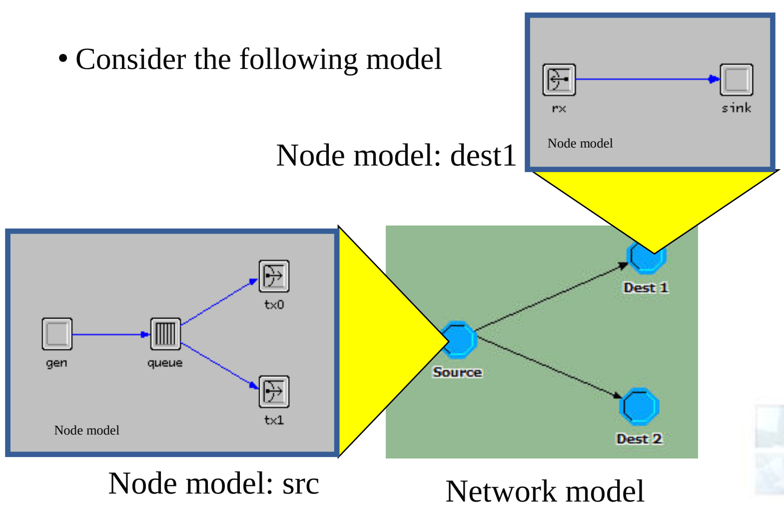
- Control passes between the Simulation Kernel (SK) and multiple processes (`pr1, pr2`, etc.), as described.

Description	In control	Idle
Read first event on event list. Deliver to appropriate process (`pr1`).	SK	All processes
Invoke enter and exit execs of initial state (`forced1`). Evaluate condition statements and transition to next state (`forced2`). Invoke enter and exit execs of `forced2`. Transition to next state (`unforced`). Invoke enter execs of next state (`unforced`). Release control to SK. Become idle.	`pr1`	SK; other processes
Remove first event from event list. Advance next event to head of list. Deliver interrupt to `pr2`.	SK	All processes
Invoke enter and exit execs of initial state (this process model is not shown). Evaluate condition statements and transition to next state ...	`pr2`	SK ; other processes

OPNET Training

Event List Example

- Consider the following model



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Event List Example

- The network model has three nodes (`src, dest1, dest2`) relying on two node models (both dest nodes use the same node model).
- In the `src` node model, packets are generated at `gen` and sent by `queue` to either transmitter `(tx0 / tx1)`.
- Packets then flow across a link to a destination node (`dest1, dest2`) where they are received (`rx`) and thrown out (`sink`).
- Three modules (`gen, queue,` and `sink`), have process models associated with them.

OPNET Training

The BEGSIM Interrupt

- BEGSIM is a special type of interrupt that occurs at simulation time 0.0, before any other type of interrupt.
- A BEGSIM interrupt usually initializes a module and schedules future events.
- Any processor or queue can have its `begsim intrpt` attribute enabled, resulting in an event being placed on the event list for time 0.0.

OPNET Training

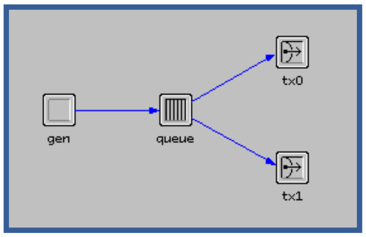
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Event List Example

- The `begsim intrpt` attribute for `gen` and `queue` is enabled; this places two events on the event list.

Time	Event Type	Module
0.0	BEGSIM	src.gen
0.0	BEGSIM	src.queue

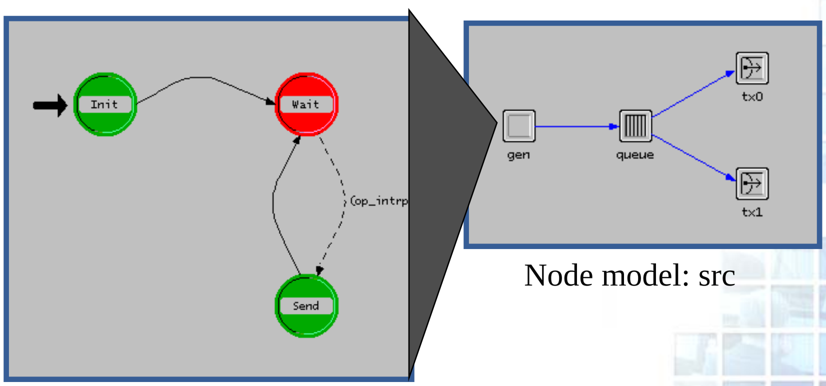


Node model

OPNET Training

Processing the First Interrupt

- Consider the process model specified by the `gen` module.



Node model: src

Process model: gen

OPNET Training

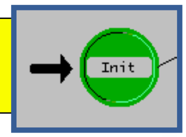
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

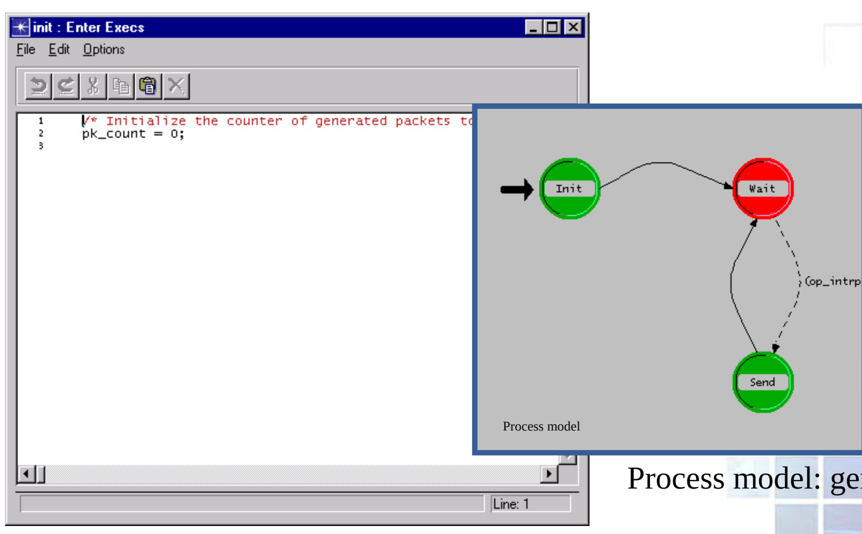
Starting the Simulation

1. To start the simulation, the head of the event list is processed and a BEGSIM interrupt is delivered to the process in `src.gen`.
2. Because this is the first interrupt, the process begins execution at the initial state, marked with a black arrow.
3. Because this state is being entered, the enter execs are executed.

Time	Event Type	Module
0.0	BEGSIM	src.gen
0.0	BEGSIM	src.queue



Processing the First Interrupt at gen



Process model: gen

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Processing the First Interrupt at gen

4. Process invokes and completes the enter execs of Init.

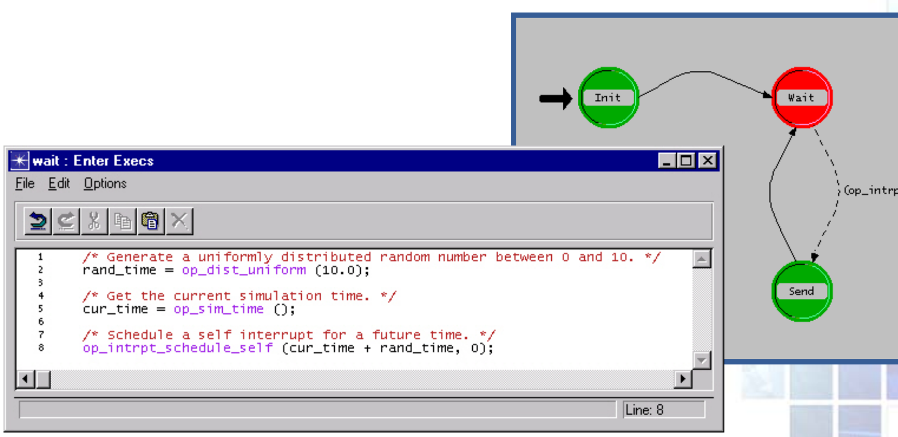
5. Because Init is a forced (green) state, process immediately invokes and completes the exit execs.

6. Process evaluates all condition statements. This state has only one departing transition which evaluates to true.

7. Process transitions to Wait.

Processing the First Interrupt at gen

8. Process invokes and completes the enter execs of `Wait`.



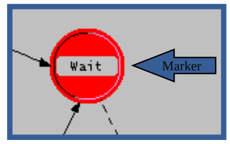
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Processing the First Interrupt at gen

9. One action in the enter execs (line 8) is to schedule a self interrupt by means of a KP. This adds an event to the event list.
10. Process places a marker at the middle of `Wait`.
11. Process becomes idle.

Time	Event Type	Module
0.0	BEGSIM	src.gen
0.0	BEGSIM	src.queue
4.3	SELF	src.gen



Processing the Second Interrupt at queue

12. Simulation Kernel removes the first event and advances the next event to the head of the event list. The simulation time remains 0.0.
13. Simulation Kernel delivers a BEGSIM interrupt to `src.queue`.
14. Process in `queue` module gains control. It executes until it reaches an unforced (red) state, places a marker, and then becomes idle. (This model is not shown.)

First event is removed

Time	Event Type	Module
~~0.0~~	BEGSIM	src.queue
4.3	SELF	src.gen

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Processing the Next Interrupt at gen

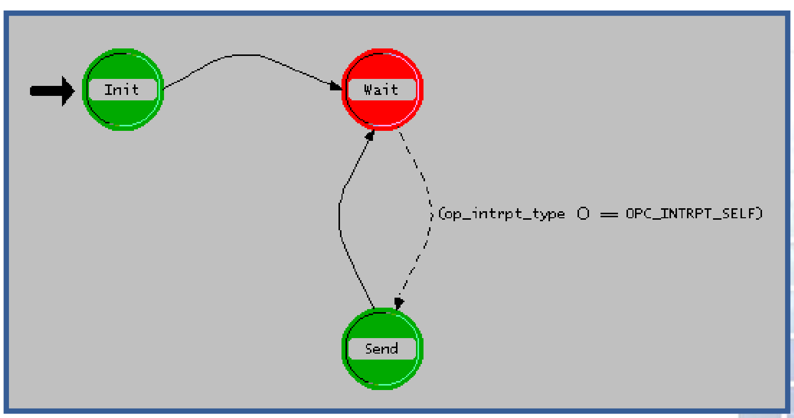
15. The Simulation Kernel removes the previous event and advances the next event to the head of the event list. The simulation time becomes 4.3 seconds.
16. The event at the head of the list causes a SELF interrupt to be delivered to the `src.gen` process. The process resumes at the marker in the middle of `Wait`.
17. Process invokes and completes the exit execs of `Wait`.

Time	Event Type	Module
4.3	SELF	src.gen

OPNET Training

Continuing the Process at gen

18. To leave `Wait`, the process evaluates all condition statements. This state has one outgoing transition and its condition statement (shown) evaluates to true.



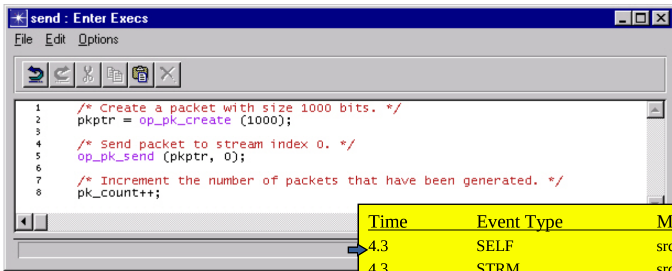
OPNET Training

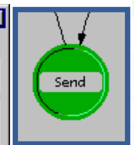
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Continuing the Process at gen

19. The process transitions to `Send`.
20. The process invokes the enter execs of `Send` and calls ***op_pk_send ( )*** to send a packet. This results in an event of type STRM being placed on the event list.





Time	Event Type	Module
4.3	SELF	src.gen
4.3	STRM	src.queue

Continuing the Process at gen

21. Because `Send` is a forced (green) state, the process immediately invokes and completes the exit execs.
22. Process evaluates all possible transitions. One evaluates to true.
23. Process transitions to `Wait`.
24. Process invokes enter execs of `Wait` , schedules another SELF interrupt in the event list and becomes idle.
25. Simulation Kernel takes control and processes the next event on the list.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

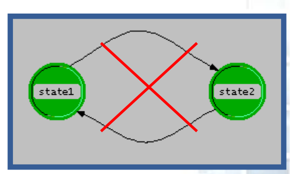
© Copyright 2007 OPNET Technologies, Inc.

Simulation Termination

- Simulations terminate in one of four ways
  - –The event list is emptied.
  - –Simulation attribute `duration` expires.
  - –A process calls for termination, using the KP ***op_sim_end().***
  - –A fatal error occurs.

How Does Time Advance?

- Simulation time advances only when an event with a later time is taken from the event list.
- No simulation time occurs during an invocation of a process model.
- No time elapses during transitions between states.
  - –A process model must always end in a red state so time can advance.
  - –Avoid endless looping between forced (green) states.



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Summary

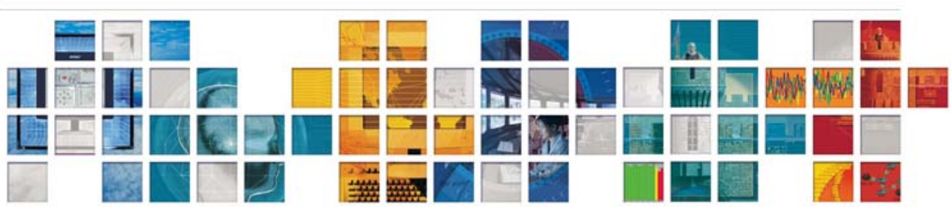
- Forced and unforced states differ in when they execute their instructions.
- Any processor or queue can have the attribute `begsim intrpt` enabled, scheduling an event for time 0.0.
- Control passes dynamically between the Simulation Kernel and individual processes.

OPNET Training

OPNET Technologies, Inc.

Making Networks and Applications Perform™

Node Modeling



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET® Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Agenda

- Conceptual Goals
  - –Creating and configuring nodes
  - –Configuring transceivers
  - –Configuring traffic generators
  - –Testing predicted behavior
  - –Deriving new models
- Tools
  - –Node Editor
  - –Probe Editor

Sequence of Events

Follow these steps

1. Understand the problem:
   - –Understand the network or system to be modeled.
   - –Understand the questions to be answered by the model.
2. Create models:
   - –Create the node models first.
   - –Create the network model.
3. Choose which statistics to collect
4. Configure simulation.
5. Determine expected output.
6. Run the simulation.
7. Analyze output.
8. Compare actual results to expected output. Explain any differences.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

General Hints

- Create descriptive names for nodes, modules, network models and node models.
- Create node models first, and only then the network model.
- To determine a particular button's function, place the mouse on it and read the explanatory text in the display areas.
- Check link consistency before exiting the Network Editor.
- To avoid confusion, you may want to exit each editor as you finish using it.

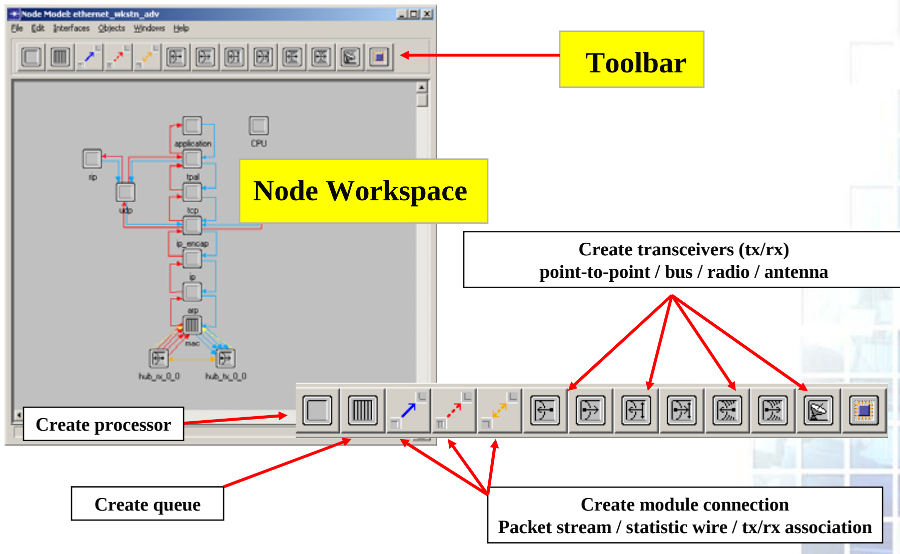
Node Editor

- The Node Editor provides the resources necessary to model the internal functions of nodes.
- You have access to different modules which are used to model internal aspects of node behavior.
- Modules represent the internal capabilities of a node such as:
  - –Data creation
  - –Transmission
  - –Reception
  - –Storage
  - –Internal routing
  - –Queuing

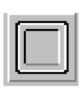
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

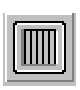
Node Editor



Node Editor - Toolbar



Processor. A module that represents the most general building block of node models. You can specify the behavior of a processor. It can be connected to other modules to share information or packets.



Queue module. A module that provides a superset of the functionality of processor modules. Queue modules can execute an arbitrary process model that describes the behavior of a particular process or protocol, and can be connected via packet streams to other modules.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Node Editor - Toolbar

Packet stream. A connection between modules that carries data packets from a source module to a destination module. They represent the flow of data across the hardware and software interfaces within a communications node

Statistic wire. A connection between modules that conveys numeric values between devices or processes in the same node. Statistic wires are primarily used to allow processes to monitor changes in state and performance of the devices that make up a node, and to create a simple signaling mechanism between processes.

Logical association. A connection used to indicate that a relationship exists between two modules in a node model, for example, between a receiver and transmitter used as a pair. Logical associations do not carry any data.

OPNET Training

Node Editor - Toolbar

Transmitters: the outbound interface between packet streams inside a node and communications links outside the node.

- Point-to-point
- Bus
- Radio

Receivers: the inbound interface between communications links outside a node and packet streams inside a node.

- Point-to-point
- Bus
- Radio

Antenna: A module that is used to specify the antenna properties for radio transmitter or receiver modules.

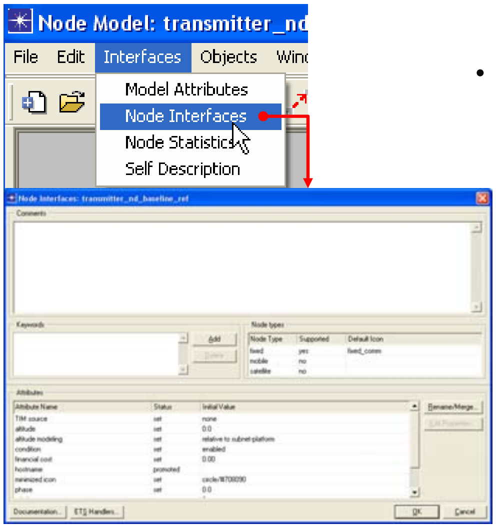
- Antenna

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Specifying Node Interfaces



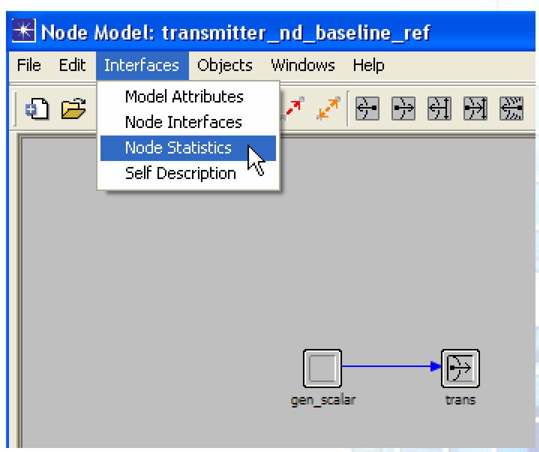
- The "Node Interfaces" option allows you to specify various characteristics of the node.

  * Node types (fixed, mobile, satellite)
  * Keywords
  * Attributes
  * Node Documentation
  * Comments

OPNET Training

Specifying Available Node Statistics

- Node Statistics allows you to select which statistics can be chosen for collection from within the Project Editor.



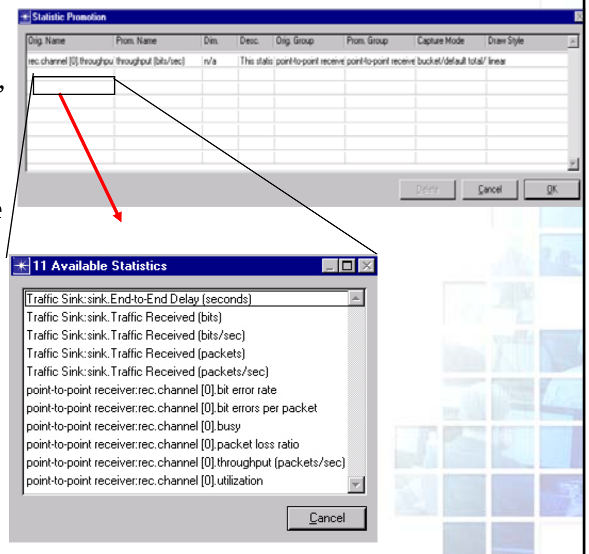
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Specifying Available Node Statistics

- By selecting an empty field in the "Orig. Name" column, a table of available statistics appears.
- Selecting a statistic from the "Available Statistics" table adds the statistic to the "Statistic Promotion" table.
- If a statistic is not promoted, a user can still collect it using the "Probe Editor".



Understanding the Question

- Preview: This lab models the flow of bank transactions (represented as packets) from Washington, D.C. to Philadelphia.
- In order to measure the performance of this simple network in a meaningful manner, you must define specific questions that your model is designed to answer.
- Design the lab to answer the following questions:
  - –Does the *queue size* of the WDC transmitter steadily increase?
  - –What is the *throughput (in bits/second)* at the WDC transmitter?
  - –What is the *throughput (in bits/second)* at the Philadelphia receiver?
  - –What is the *utilization* of the DC to Philadelphia link?

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Guidelines for Setting Probes

- Follow this procedure
  - –Review the questions to be answered.
  - –Determine which module or link should be monitored for each question.
  - –Review the different types of statistics available, and choose the appropriate ones etc)

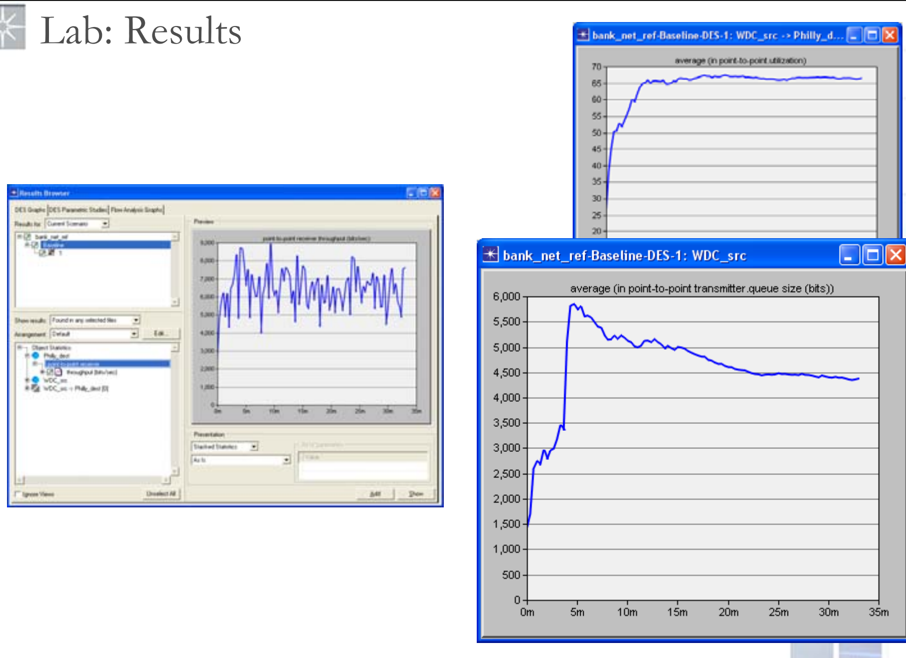
Lab: Node Modeling

- Lab Book: Lab: Node Modeling
- Bank transactions originate in Washington, D.C. (WDC) and are routed to Philadelphia via a telephone line and modem capable of transmitting **9,600 bits/second**.
- The size of a transaction varies according to a normal distribution with a **mean size of 3,200 bits and a variance of 400 bits**.
- Transactions are modeled as exponential interarrivals, with a **mean interarrival time of 0.5 sec/trans**.
- The goal is to analyze performance of a system in steady state.
- Create a transaction packet with 2 fields, source node, and destination node. Each field has a size of 64 bits.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Lab: Results



OPNET Training

Summary

- The node editor gives you complete control over the design of a node's architecture to reflect the behavior of a device based on specifications.
- The probe editor expands your capability to customize the selection of statistics.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Technologies, Inc.

Making Networks and Applications Perform

Process Modeling

OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Agenda

- Conceptual goals
  - –Process models
  - –Custom statistics
  - –Kernel Procedures
- Tools
  - –Process Editor

OPNET Training

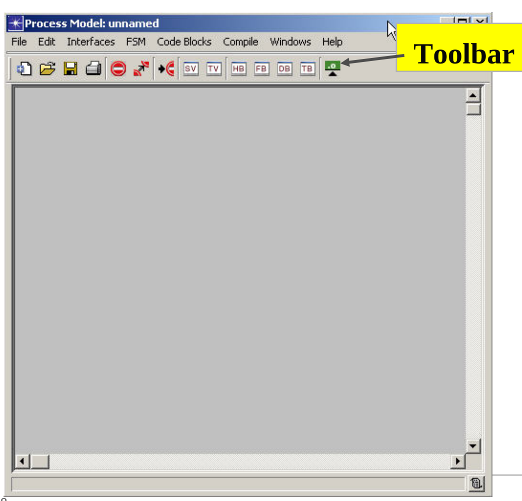
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Process Models

- A process in the context of computer systems and communications networks can be viewed as a series of logical operations performed on data, and the conditions that cause these operations.
- Processes may be implemented in terms of both hardware or software components.
- OPNET *process models* describe the logic of real-world processes, such as:
  - –Communications protocols and algorithms
  - –Shared resource managers
  - –Queuing disciplines
  - –Specialized traffic generators
  - –Statistic collection mechanisms
  - –Operating systems
- The Process Editor provides the necessary features for specifying process models, which consist of both graphical and textual components.

Process Model Editor



1. Create state
2. Create transition
3. Set initial state
4. Edit state variable
5. Edit temporary variable
6. Edit header block
7. Edit function block
8. Edit diagnostic block
9. Edit termination block
10. Compile process model

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Process Editor Toolbar

Create State: Creates a new state within the process model.

> ***State:*** One of the components of a finite state machine. The behavior of a state is defined by its state executives, which are executed upon entry into and exit from the state.

Create Transition: Creates transitions between states.

> ***Transition:*** The path a process takes between states in a finite state machine. Transitions contain attributes that can be used to specify conditions that must be met before the transition takes place.

Set Initial State: Sets the selected state in the process to be the initial one.

Process Editor Toolbar

State Variables Block: Defines variables that retain their value from one process invocation to the next.

Temporary Variables Block: Defines variables that retain their value only during the span of a single process invocation.

Header Block: Defines constants, macro expressions, include files, global variables, data structures, data types, and function declarations for the process. Also declares whether the process model will be in C/C++

Function Block: Defines C/C++ functions that are associated with the process.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Process Editor Toolbar

Diagnostic Block: Defines C/C++ statements that send diagnostic information to the standard output device.

Termination Block: Defines C/C++ statements that execute just before a process is destroyed.

Compile Code: Generates the C/C++ source file and object code for the process model.

Creating Process Models

Follow these steps:

1. Understand the questions to be answered.
2. Create a new process model or modify an existing process model.
3. Edit the node model to use the new/modified process model.
4. Reselect statistics if necessary..
6. Determine the expected output.
7. Run simulations.
8. Analyze output and post-process it to answer questions.
9. Compare actual results to expected output. Explain any differences.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Lab: Process Modeling

- Lab Book: Lab: Process Modeling
- This lab uses the project **Bank_net**, to model changes in the transaction rate and measure the effects of those changes.
- In this lab, we will consider the following questions:
  - What is the average end-to-end delay (ETE delay) for all transactions when the generation rate is 0.5 seconds/transaction (or 2 transactions/second)?
  - For this generation rate, what percentage of the packets incurred an ETE delay of less than 1 second?

203 OPNET Training

Summary

- Process modeling enable you to generate customized models and statistics.
- Built-in Kernel Procedures facilitate the coding of process models.
- In this Chapter, we learned how to create process model from scratch and incorporate it into existing node model.

204 OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Parametric Studies

OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET® Training

Agenda

- Conceptual goals
  - –Creation of an attribute of a process model
  - –Collect scalar statistics
- Tool goals
  - –View scalar analysis results

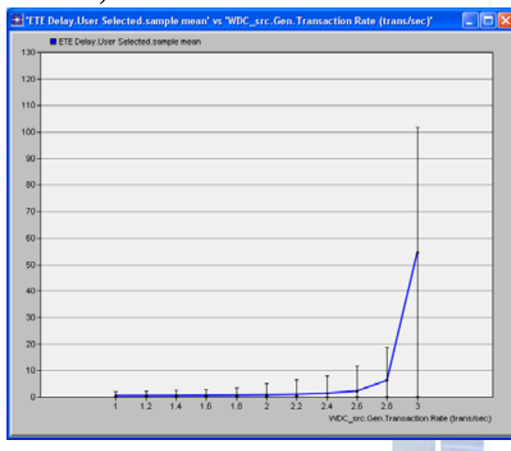
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

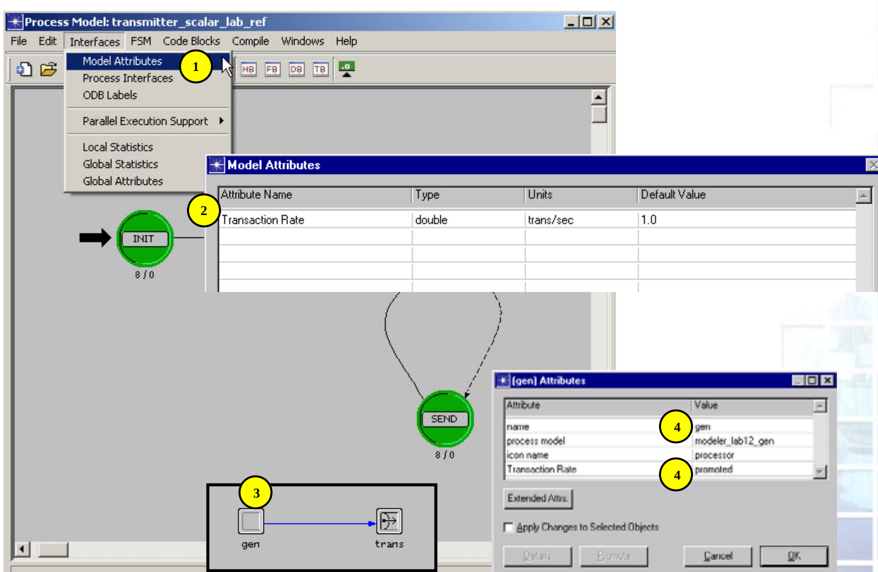
Parametric Studies

- A scalar statistic is a single value representing a statistic computed over an interval, typically the entire simulation (example: average end-to-end delay for all received packets in a simulation).



- Scalar statistics:
  - –Must be plotted versus another scalar statistic
  - –Allows for parametric studies
  - –Requires many simulation runs because each run produces only one data point
  - –Each simulation will add a new data point

Creating an Attribute for the Process Model

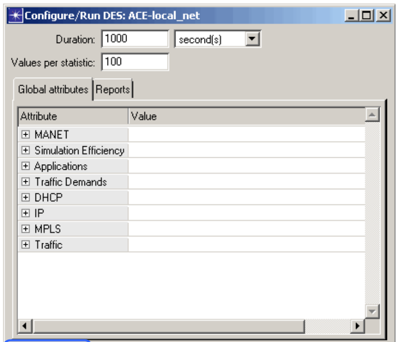


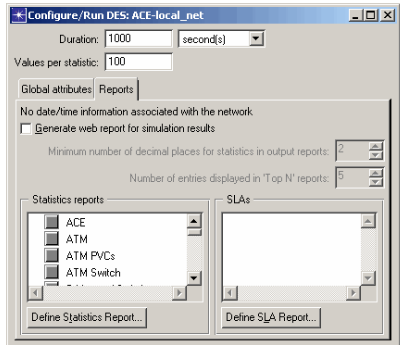
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

"Simple" Configure/Run DES

- Very small set of configurable options
- No parametric support
- Default mode for IT Guru and SP Guru



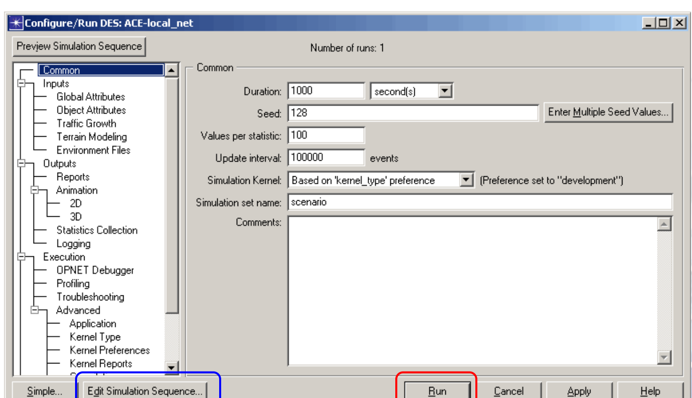


Switch to Detailed version of the Configure/Run dialog box

OPNET Training

"Detailed" Configure/Run DES

- Similar to 11.5 dialog box
- Default for Modeler



Switch to Advanced version

Execute DES

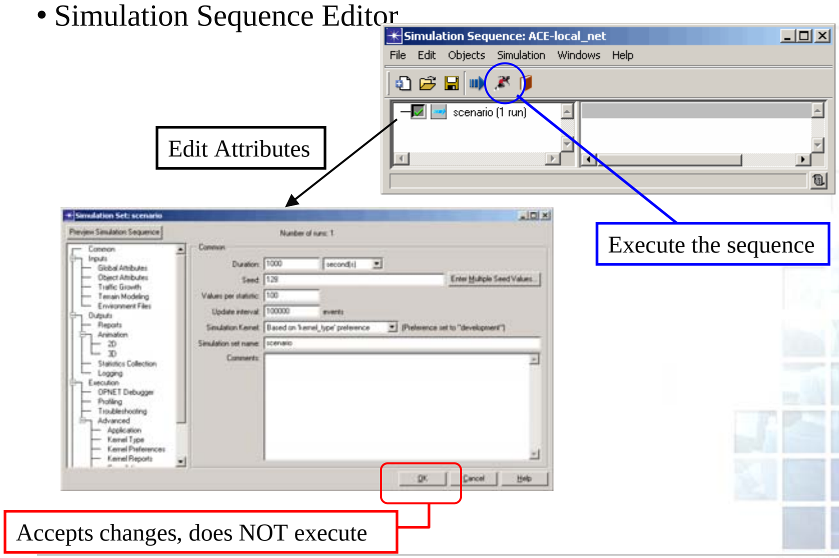
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

“Advanced” Configure/Run DES

- Simulation Sequence Editor



OPNET Training

Parametric Studies Workflow

- Select a model to study
- Choose the parameter(s): any attribute, preference, or property of model
  - –Parameter(s) will vary over a range of values
- Select statistic(s) of interest
- Execute group of related simulations by varying the parameter(s) over their respective value ranges
- Plot parametric graphs showing data evolution as a function of parameter values

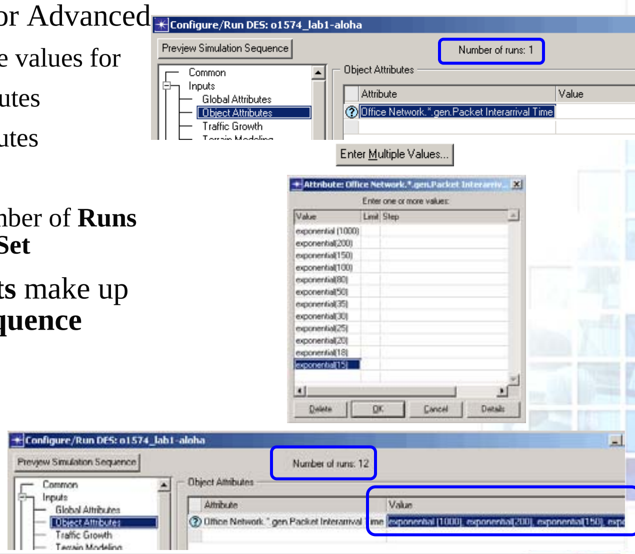
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Parameters in Configure/Run DES

- Using Detailed or Advanced
  - –Specify multiple values for
    - Global Attributes
    - Object Attributes
    - Seed
  - –Defines the number of **Runs** in a simulation **Set**
- One or more **Sets** make up a simulation **Sequence**

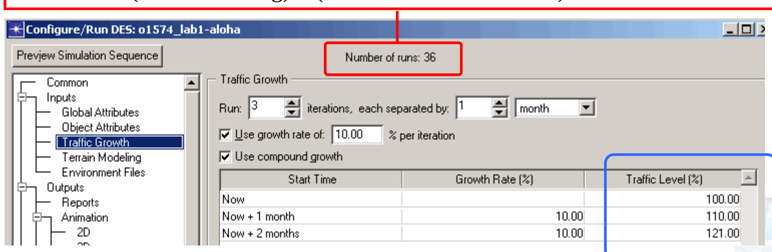


OPNET Training

"Traffic Growth" Parameter

- Specifies values for 'Traffic Scaling Factor' global attribute
  - –Attribute does not appear in the 'Global Attributes' panel

Number of Runs is product of each parameter's value count
In this case (Traffic Scaling) x (Packet Interarrival Time) = 3 x 12 = 36



Traffic Scaling Factor set to 1, 1.1, and 1.21

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

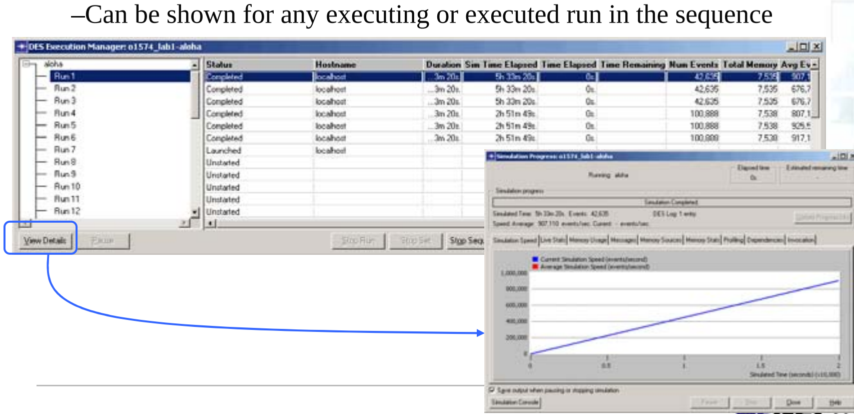
© Copyright 2007 OPNET Technologies, Inc.

Parameters Across Multiple Scenarios

- Execute simulation(s) for several scenarios in one or more projects
  - –Vary anything you want in these scenarios
  - –Require some scheme to produce appropriate scalar value for each scenario
    - •Number of nodes, QoS arbitrary “index” value, etc.

DES Execution Manager

- Displayed whenever there is more than one run to execute
  - –Summary info for each run as it is executed
- Simulation progress
  - –Only dialog box shown if there is only one run
  - –Can be shown for any executing or executed run in the sequence

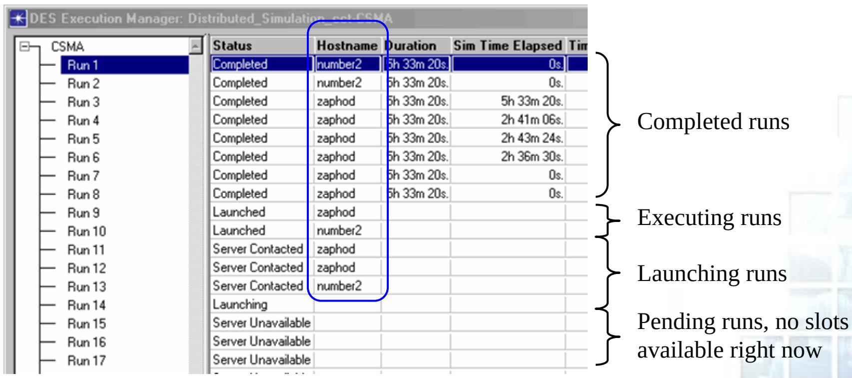


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Distributed Execution of Simulation Runs

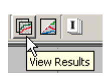
- New in 12.0: Ability to launch runs on remote machines

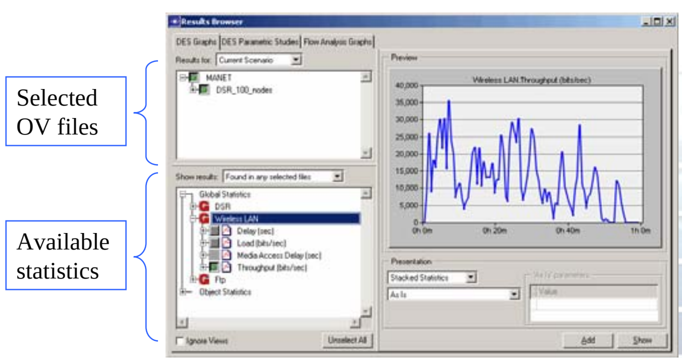


OPNET Training

Results Browser

- Enhanced version of previous vector data viewer
  - –Right-click in Project Editor and select 'View Results'
  - –Choose DES > Results > View Results
  - –Click on View Results toolbar button





OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

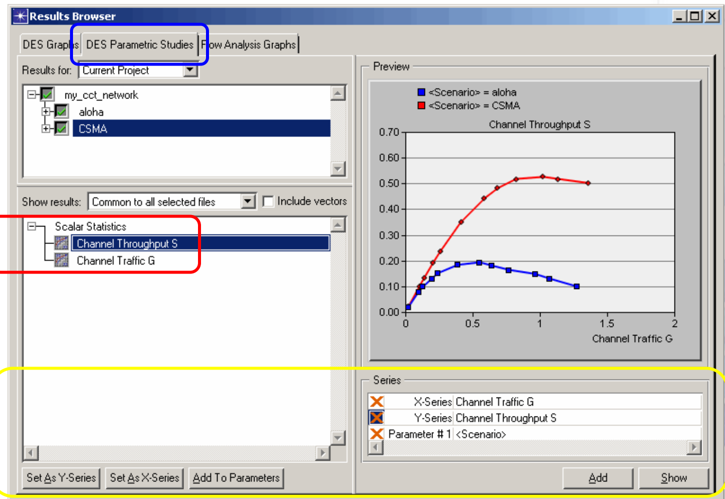
Selecting Recorded Scalar Data

• Open Results Browser and select ‘DES Parametric Studies’ tab

12.0 .ov
Pre-12.0 .os

Recorded
scalar data

Presented in
more details
in the lab



OPNET Training

Viewing Vector’s Scalar Data

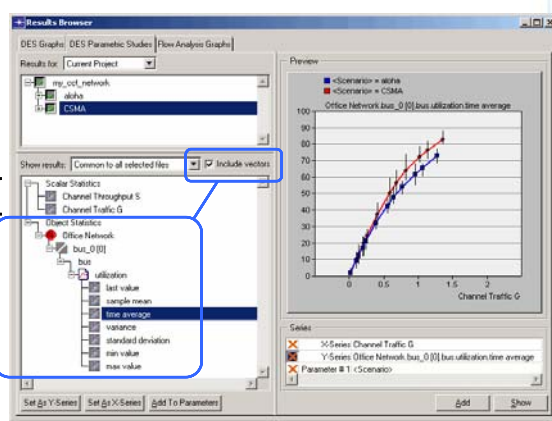
• Scalar probe generates scalar value during simulation from raw stat

• GUI can generate scalar values from collected vector data

–Data based on any vector processing (buckets, glitch removal)

Data from scalar probes
Generated during simulation

Scalar from OV data
Computed on-the-fly



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

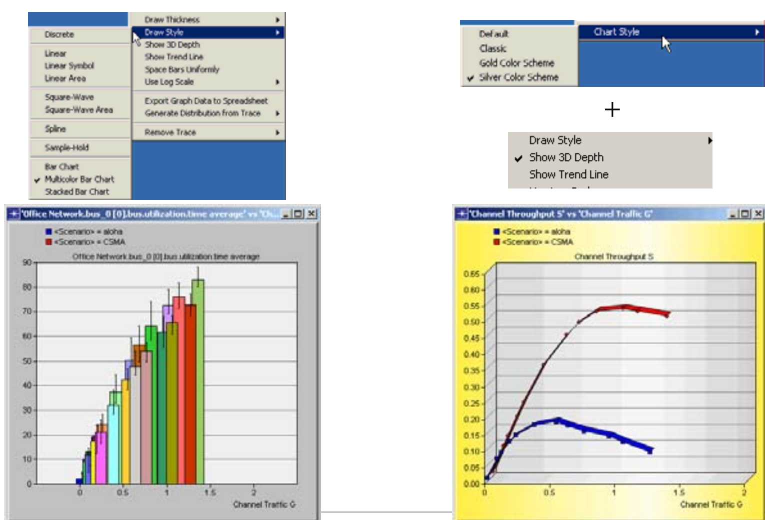
Enhanced Graphic Options

- Exert more control over your outputs with new charting features in release 12.0
  - –New line styles like linear symbol, area, multi-color bar
  - –Log scale for X or Y axes
  - –Optional 3D look
  - –Background schemes

OPNET Training

Enhanced Graphing Options

- Of course, it depends on your idea of the “perfect” look



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

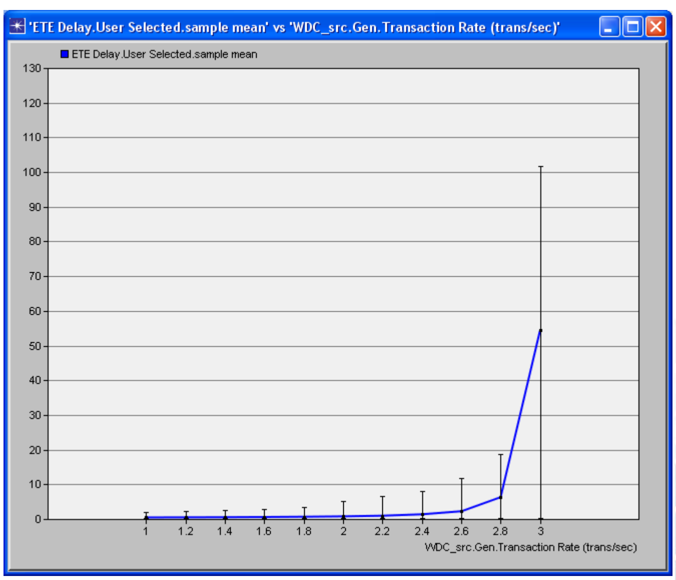
© Copyright 2007 OPNET Technologies, Inc.

Lab: Parametric Studies

- Lab Book: Lab: Parametric Studies
- Modify the Bank_net project to include a custom generator that has an attribute `transaction_rate`, the transaction rate in transactions per second.
- Answer the following questions:
  - –Are your results the same with this customized generator as with the ideal generator? *Hint: run the simulation with the same settings as in the previous lab. Verify that results are the same.*
  - –What is the maximum generation rate such that the average ETE delay for all transactions is less than 5 seconds? *Hint: run multiple simulations with different transaction rates.*

OPNET Training

Lab: Results

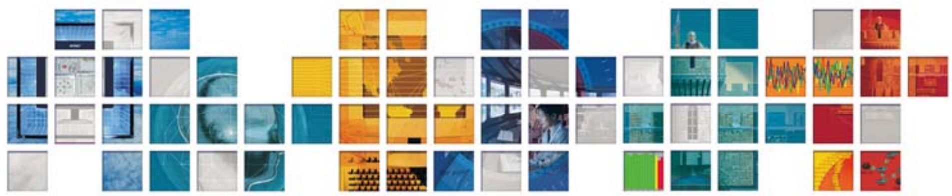


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Modeling Large Networks



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Agenda

- Conceptual goals
  - –Importing network topologies
  - –Choosing aggregation levels
  - –Using LAN models
  - –Importing traffic
  - –Using routed background traffic
  - –Managing projects and scenarios
  - –Navigating Large Topologies
  - –Controlling Simulation Runtime
  - –Effectively Using Background Traffic
- Software goals
  - –Import topology
  - –Question and answer browser
  - –Import traffic
  - –Network browser
  - –Traffic browser
  - –Efficiency Modes
  - –Static Background Traffic
  - –Event Speed Parameter
  - –Specifying Traffic Growth
  - –Scheduling Automated Simulations
  - –Viewing and Editing Scheduled Simulations Log

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Overview – Large networks

- The concept of a “large” network does not necessarily refer to the number of nodes or the geographic extent of a network.
- A large network is any model that will generate a significant amount of events. This translates into longer simulations and memory limitation issues.
- We will explore methods for increasing the efficiency of your models when the economies of scale become an issue.

Common Pitfalls

- OPNET is powerful software that tempts users with “grand designs”.
- Common Pitfall: “I will model everything”
- Developing models without clear direction and a solid understanding of the questions that need answering leads to inefficiency, and results that aren’t useful.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

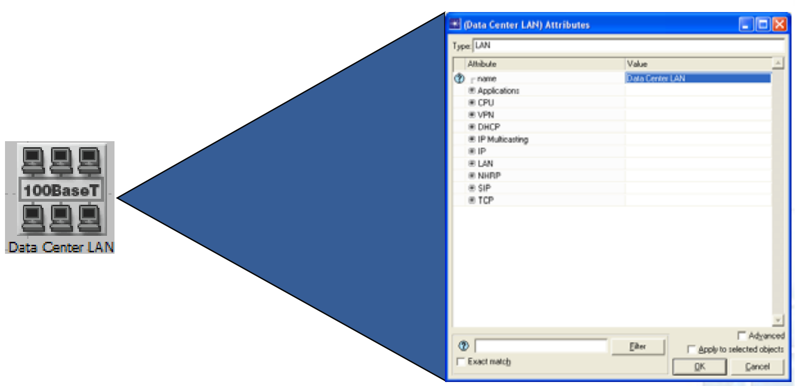
© Copyright 2007 OPNET Technologies, Inc.

Questions Should Drive Model Design

- It essential to first determine *which* questions to answer.
- Design the model to answer those questions. Don't try to answer everything.
- A key factor in effective modeling is flexibility. Realize that over the course of the modeling process that the questions may change.

LAN Models

- Single nodes may be used to model entire LANs.

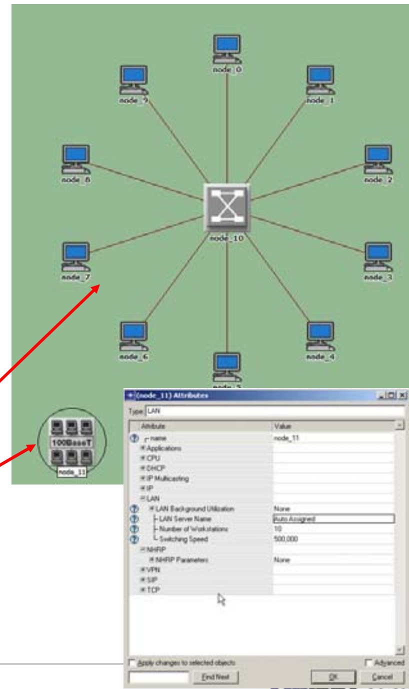


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Why Use LAN Models?

- LAN models are a powerful mechanism for abstracting complex local area networks that may contain multiple nodes with the same configurations.
- LAN models reduce clutter in the workspace.
- LAN models use less memory.
- LAN models generate fewer simulation events allowing simulations to run faster.



These models represent the same network.

Additional Information About LAN Models

- LAN Models behave as if they were specific nodes modeled using OPNET.
- The key difference is that you can collect statistics on the LAN models but not on the specific nodes and links within the model.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Modeling Techniques

- Design Accurate, Efficient Models
- Two Techniques
  - –Approximation
  - –Reduction

Approximation

- Tradeoff: Precision vs. Performance
  - –Loss in precision means higher variance
  - –Overall results are still accurate
- Approximation Approach
  - –Apply a combination of analytically and explicitly modeled traffic
  - –Allows you to limit the number of variables, while retaining the proper traffic behavior for the model

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

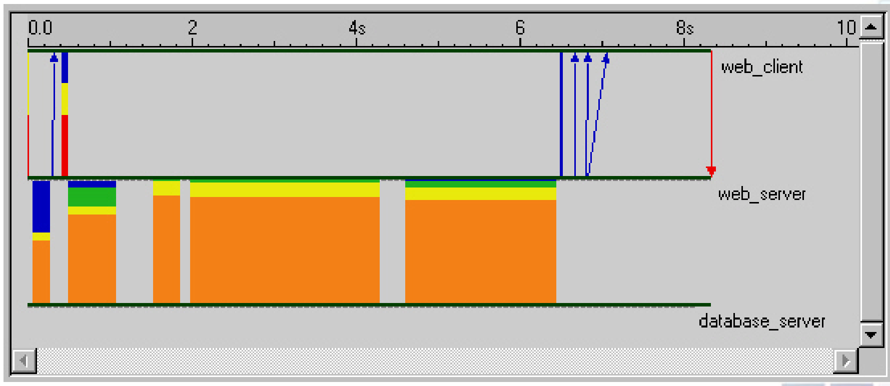
Traffic Types

- Three different types of traffic
  - –Loads
    - Also called "Device/Link Loads"
    - Represent percentage of capacity being used
    - Can be imported or manually created
  - –Flows
    - Visualized as demand objects
    - Can be imported or manually created
  - –Packets
    - Provide detailed performance statistics
      - –Delay, response time, jitter
    - Represent different types of application traffic
    - Can be imported from optional ACE module or manually created
- All three can exist simultaneously in the network model

OPNET Training

Importing and Characterizing Applications

- With the Application Characterization Environment (ACE) module, you can model any networked application by capturing the packets associated with that application, then importing them into Modeler.



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Link Loads

- Represent utilization of individual links
- Values are represented in percentage utilization for each time period
  - Utilization A->B
  - Utilization B->A
- To set link loads manually:
  - Right-click a link, edit attributes
  - Expand “Background Utilization” attribute
  - Set utilization percentages for multiple time periods
- To import data from management platforms:
  - Traffic / Import Device/Link Loads
  - MRTG
  - InfoVista
  - Concord eHealth-Network
  - Text Files

OPNET Training

Flows

- Represent flow of network traffic between sources and destinations in network
- Flows are represented in bits/sec and packets/sec
- Flow data can be imported from management platforms
  - Cisco NetFlow (AS Aggregation or No Aggregation)
  - Cflowd
  - NetScout
  - NetScout Ngenius
  - NAI Distributed Sniffer / Sniffer Pro
  - Agilent NetMetrix
  - Spreadsheet / text files
- You can add new flows manually by choosing: Protocols / IP / Demands / Configure Traffic Demands Among Selected Nodes

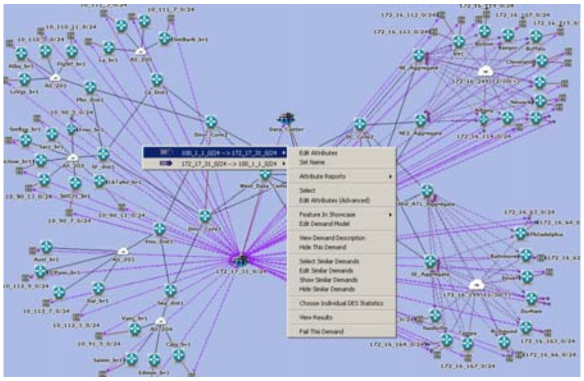
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Flows

- Flows are a type of traffic demand. Demands appear as dotted blue lines from source to destination.
- Right-click a traffic demand to edit its attributes or hide the demand
- Use **View / Demand Objects** to hide/show all demands
- Use **Traffic / Open Flows Browser** to quickly examine many flows



OPNET Training

Packets

- Only used for discrete event simulation (DES)
- Represent individual packets in network
  - –Application traffic
  - –LSAs
- Application Configuration node used to globally define application traffic
- LSAs automatically sent by routers during simulation
- Capture the transient behavior of the network
  - –Convergence
  - –Latency
  - –Queue depth
  - –Protocol effects
- Verify Service Level Agreement compliance

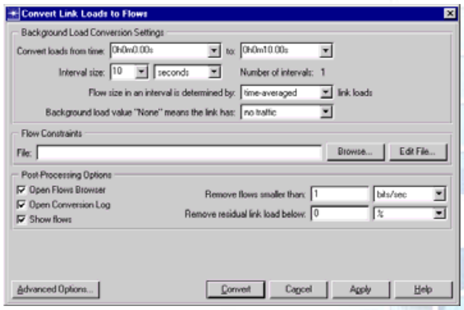
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Converting Loads to Flows

- –Import load traffic
- –Gather information about traffic characteristics in real network
- –Traffic / Convert Link Loads to Flows…
- –Give Modeler constraints:
  - List of possible endpoints of traffic flows
  - Max/min amounts of traffic per endpoint
- –Modeler performs conversion
- –Caveat: This is an approximation!
- –More constraints = more realistic flows
- –Flow traffic can be used for failure, routing, and QoS studies



OPNET Training

Summary: Traffic Modeling

- Different traffic types available for different types of studies
- Three types available for import or creation:
  - –Flows
  - –Loads
  - –Packets
- Flows are best choice for failure studies
- Loads are often easiest to obtain and best for viewing utilization of current network
- Packets are only used in discrete event simulation. Packets give the most detailed results and are the best choice for QoS studies.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Object Reduction

- Each and every object in a model uses memory.
- Reducing the number of objects in a simulation allows for more efficient simulations, possibly allowing the workstation to overcome memory limitations (reduce swapping).
- Modeler provides LANs, clouds and devices with varying interfaces to allow you to keep the number of objects in the network model to a minimum without sacrificing precision.

Object Reduction

- LANs allow you to represent any number of workstations and a server as a single object.
- IP Clouds abstract numerous IP devices into a single cloud object.
- Each interface in a device requires memory. Creating devices that have the minimum amount of interfaces required for a model reduces memory needed to run simulation.
  - Example-- If all the hubs in a model will only have 8 connections, it is more efficient to use an ethernet8_hub than an ethernet64_hub. Even if this doesn't represent your actual hardware, it will not affect the precision of the simulation.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Alternative Solutions

- There are other solutions that do not involve model design.
- These alternative solutions can be characterized as "brute force" methods.
- The advantage to alternative solutions is that it gives you more latitude in your model designs.
- Increasing physical memory is always an effective way to increase workstation performance.
- Increasing the swap space is another effective method to get a better performance from a computer.

OPNET Training

Controlling Simulation Runtime

- Every protocol in Modeler has attributes associated with them. These attributes represent real-world application of a protocols procedures and techniques.
- In many cases, although representing a protocol's actual behavior, certain attributes may not add value to a user's model. In fact they will generate events that are not helpful to the results of the model and will actually increase simulation run-time.
- Modeler accounts for this by enabling certain efficiency modes in order to decrease the number of unimportant events and accelerate simulation run-time.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

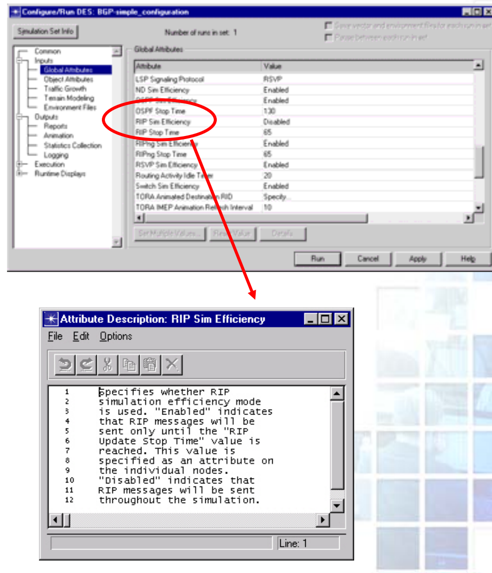
© Copyright 2007 OPNET Technologies, Inc.

Controlling Simulation Runtime

- When configuring a simulation, you will be able to view and edit a list of simulation attributes. These attributes will change based on the protocols being modeled.

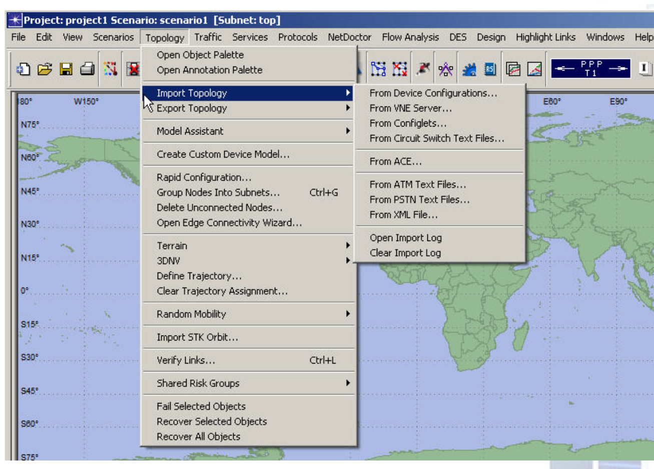
- Among these attributes are a variety of efficiency modes that are enabled.

- Right-clicking on a simulation attribute will provide you with a complete description of that attribute.



Importing Topology Information

- OPNET can import topologies from
  - –Device Configurations (XDI Module)
  - –Circuit Switch Text Files
  - –ACE
  - –ATM Text files
  - –XML files
  - –VNE Server



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

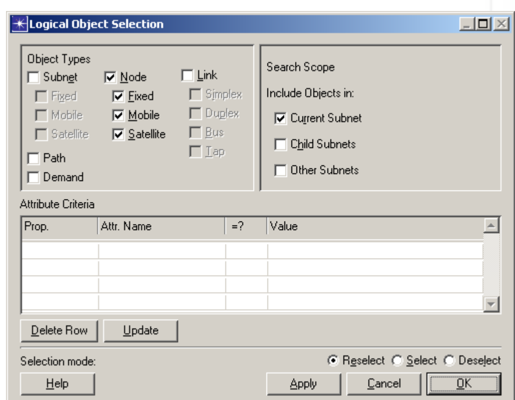
© Copyright 2007 OPNET Technologies, Inc.

Navigating Large Topologies

- OPNET provides you with various tools to facilitate the locating and selecting of nodes in a large network:
  - –Select Objects Logically
  - –Network Browser

Edit / Select Objects

- Choose **Edit / Select Objects** to select certain objects throughout network based on the value of attributes.
- Ex: Select all links with a data rate greater than 1.544 Mbps

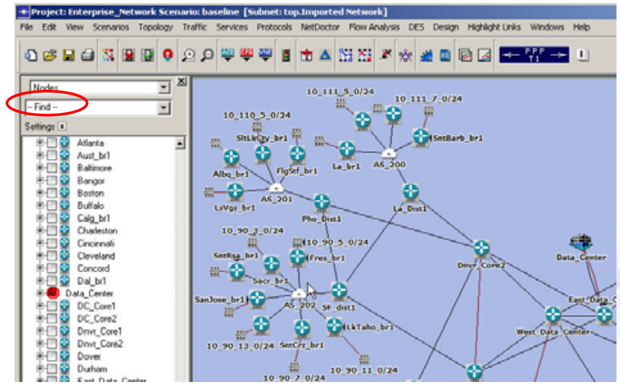


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Network Browser

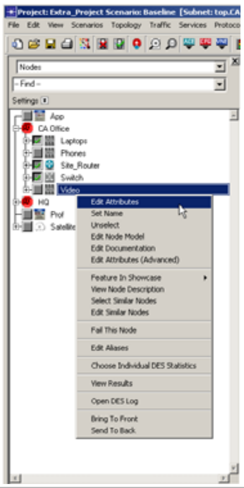
- The Network Browser can be used to locate objects in your network.
- Choose **View / Show Network Browser**
- Type a search string into the “Find” box to show objects with that string in their names.



OPNET Training

Network Browser

- To view an object’s attributes, right-click the object in the workspace or the list and select “Edit Attributes.”



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

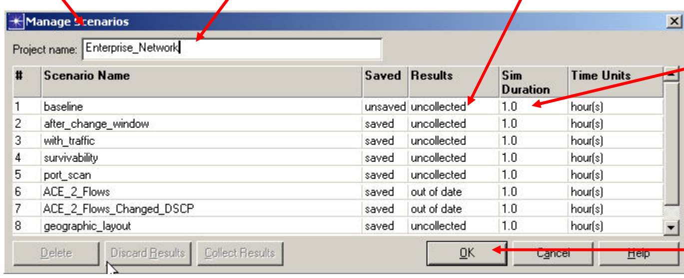
Managing Projects and Scenarios

- Managing scenarios allows you to examine the status of scenario results, change project/scenario names and run multiple simulations sequentially.

Change the project name here.

Change the scenario name here.

Collect results on multiple scenarios by changing the results column to <recollect>. Clicking OK will start a simulation run for each scenario with <recollect> set.



Change the simulation duration.

Ok will write all of the changes to the project.

Summary

- Networks can be imported in a variety of manners.
- Modeler's power as modeling software becomes evident with the analysis of large networks. As the size of the model increases so does the importance of efficient modeling.
- Avoiding common pitfalls and understanding the efficiency techniques inherent in Modeler leads to useful results and efficient models.
- By skillfully using the combination of the three types of traffic, you can achieve a comfortable balance of performance and precision.

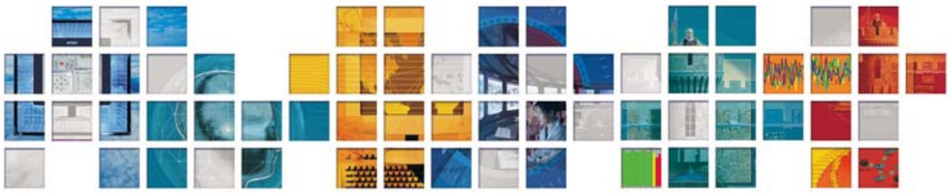
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Technologies, Inc.

Making Networks and Applications Perform

Wireless Modeling



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Agenda

- Conceptual Goals
  - –Introduction to wireless modeling
  - –Introduction to Transceiver Pipeline Stages
  - –Mobile/Satellite Nodes
  - –Custom Packets
  - –Custom Antennas
  - –Link Models
  - –Custom PDFs
- Tool Goals
  - –Define Trajectory
  - –Packet Format Editor
  - –Link Model Editor
  - –Antenna Pattern Editor
  - –PDF Editor

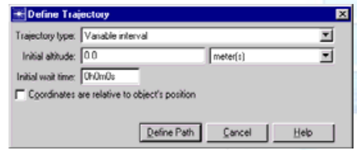
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Wireless Modeling

- The Wireless Module allows models to utilize mobile nodes and mobile subnets using dynamic RF links.
- Node and subnet positions are updated (lazy evaluation) based on user specifications such as time step, altitude and a defined path that is anchored on absolute or relative coordinates.



- Wireless transceivers must be used that employ a 14 stage link budget model called the Transceiver Pipeline

OPNET Training

Wireless Modeling

- Wireless is a broadcast technology and depends on dynamically changing parameters. The simulation must evaluate the possible connectivity between a transmitter channel and every receiver channel for each transmission.
- The network level characteristics factored into these calculations are the locations of the source and destination nodes, the distance between the nodes, and the direction the wireless signal travels from the source node to the destination node.
- If the nodes are mobile or satellite nodes, these position-related parameters may change during simulation.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

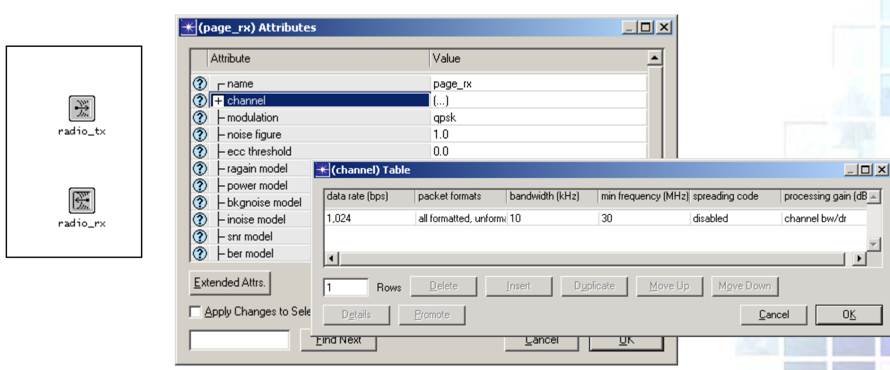
© Copyright 2007 OPNET Technologies, Inc.

Wireless Links

- A wireless link is not statically represented by an object, as are point-to-point and bus links.
- A wireless link can exist between any radio transmitter–receiver channel pair and is dynamically established during simulation.
- The possibility of a wireless link between a transmitter channel and a receiver channel depends on many physical characteristics of the components involved, as well as time-varying parameters, which are modeled in the *Transceiver Pipeline Stages.*

Wireless Links

- In OPNET simulations, parameters such as frequency band, modulation type, transmitter power, distance, and antenna directionality are common factors that determine whether a wireless link exists at a particular time or can ever exist

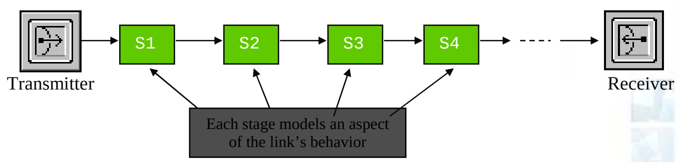


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Transceiver Pipeline

- Models the *transmission* of packets across a *communications channel* (link)
- Implements the *"physical layer"* characteristics
- Divided into *multiple stages*, each modeling a particular aspect of the channel
- Determines whether or not a packet can be *received* at the link's destination



OPNET Training

Radio Transceiver Attributes for Specifying Pipeline Stages

Radio Transmitter

- Receiver Group
- Transmission Delay
- Link Closure (LOS)
- Channel Match
- Tx Antenna Gain
- Propagation Delay

6 Stages (0-5) Associated with Radio Transmitter

Radio Receiver

- Rx Antenna Gain
- Received Power
- Background Noise
- Interference Noise
- Signal-to-Noise Ratio
- Bit Error Rate
- Error Allocation
- Error Correction

8 Stages (6-13) Associated with Radio Receiver

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Radio Transmitter and Receiver Attributes

- **Modulation** - name of modulation table used to look up the bit error rate (BER) as a function of effective signal to noise ratio
- **Channel** - specifies the number and attributes of the channels in the transceiver
- **Noise figure** - represents the effect of thermal noise on wireless transmission (receiver only)
- **Ecc threshold (err/bits)** - specifies the highest proportion of bit errors allowed in a packet in order for the packet to be accepted by a receiver (receiver only)
- **... model** - these attributes specify the various *Transceiver Pipeline Stage* models used

Wireless Channel Attributes

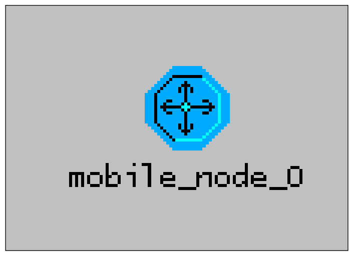
- **Data rate (bps)** - rate at which data may be transmitted or received
- **Packet formats** - determines the types of packets a channel can transmit or receive
- **Bandwidth (KHz)** - specifies bandwidth of channel
- **Min frequency (MHz)** - specifies the base frequency of the channel
- **Spreading code** - used to specify a user-assigned code for the channel
- **Power (W)** - transmission power of packets transmitted through this channel (transmitter only)

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Mobile Nodes

- Mobile nodes model terrestrial network elements whose positions vary with time, such as automobiles, aircraft, and ships.
- Mobile nodes cannot be connected to point-to-point and bus links, because they move relative to the earth.
- Mobile nodes may change position during a simulation
  - –An Assigned Trajectory
  - –Random Movement
  - –Directly Changing Node’s Position Attributes



OPNET Training

Mobile Node Trajectories

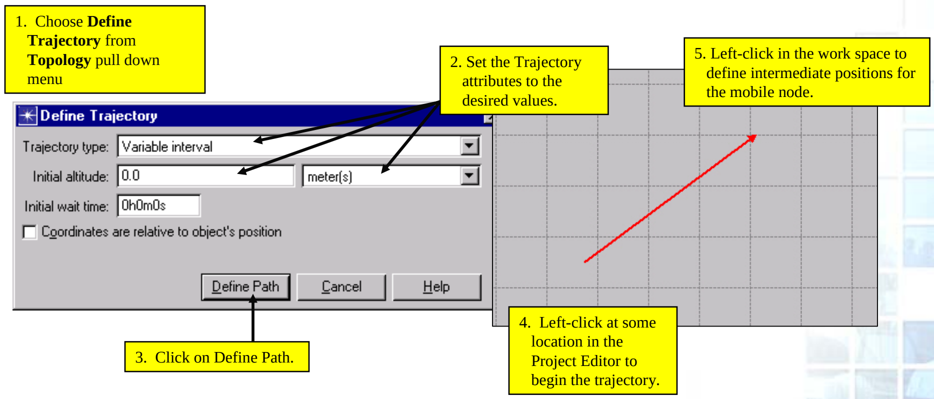
- A trajectory is the path a mobile node moves along during simulation.
- Trajectories can be created using the “Define Trajectory” selection in the “Topology” pull down menu.
- During simulation, the mobile node follows the trajectory by traveling in a straight line from one defined position to the next.
- Once the simulation time exceeds the last specified time in the trajectory file, the mobile node remains at its final position.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

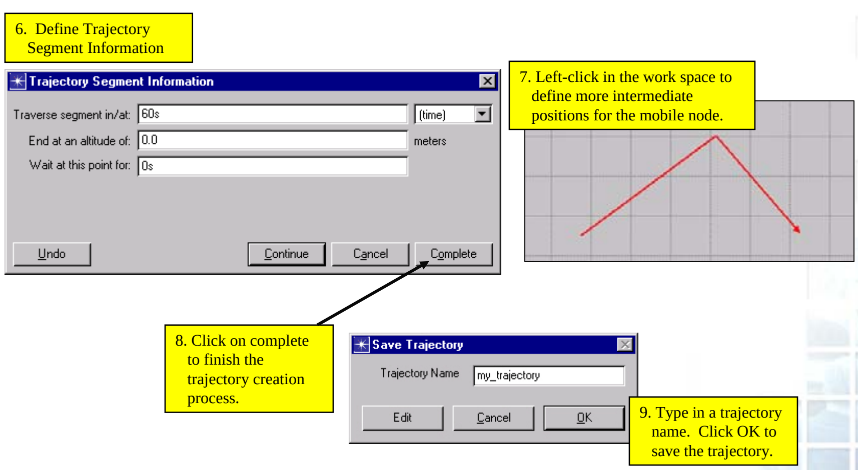
© Copyright 2007 OPNET Technologies, Inc.

Defining a Trajectory



- Mobile nodes support an attribute called trajectory which specifies the name of an ASCII text file defining the path of the mobile node during the simulation.

Defining a Trajectory Continued

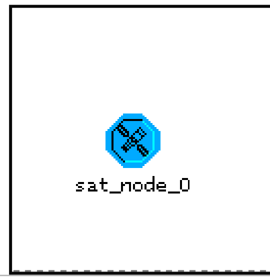


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Satellite Nodes

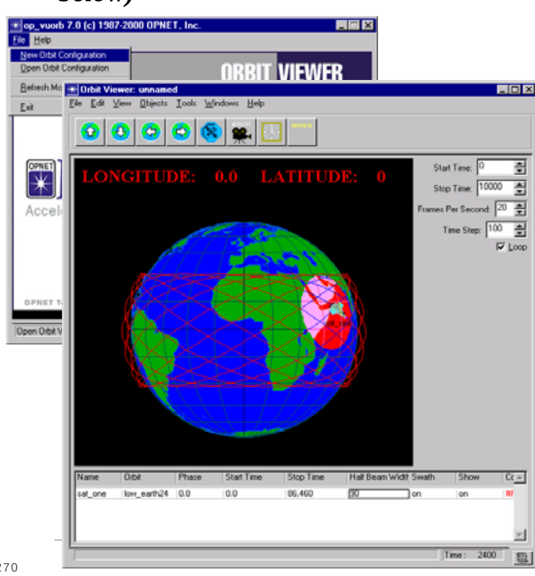
- Satellite nodes model network elements in orbit around the earth, such as satellites and spacecraft.
- Like mobile nodes, satellite nodes cannot be connected by point-to-point and bus links because they change positions during a simulation.
- Satellite nodes change position based on an assigned orbit or by direct changes to the node's position attributes.

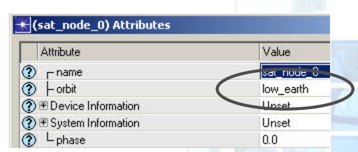


OPNET Training

Orbits

- Orbits can be imported from **STK** and viewed using ***op_vuorb (shown below)***





OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

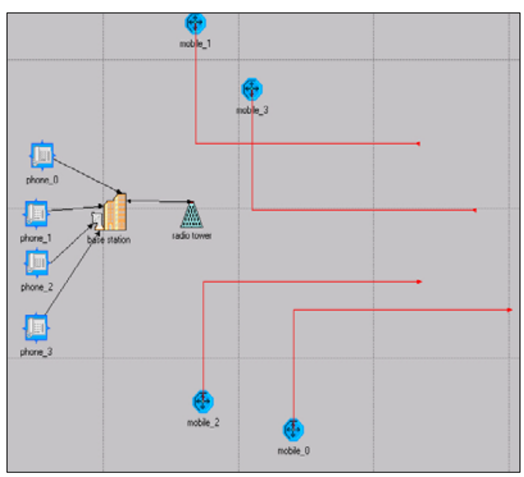
© Copyright 2007 OPNET Technologies, Inc.

Wireless Features

- Simulation Efficiencies
  - –Dynamic Receiver Group
  - –Parallel Processing
- Add-on Modules
  - –TMM: (Terrain Modeling Module)
    - The Terrain Modeling Module allows you to take into account terrain when modeling wireless networks.
    - With TMM, you can import elevation maps that contain terrain data.
    - Supported map formats are DTED and DEM.
    - TMM also allows you to select and compare signal loss from various propagation models.
    - Requires Wireless Module
  - –TIREM (TIREM propagation modeling)
    - Adds the ability to use the *TIREM* propagation calculations in addition to the built-in *Free Space* and *Longley-Rice* propagation modeling
    - Requires Wireless and TMM Modules

Lab: Wireless Modeling

- This lab will model a mobile paging system.



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Lab: Overview

- Parameters:
  - –Telephones initiate pages addressed to a specific mobile pager.
  - –The pages are relayed through a central base station.
  - –A radio tower broadcasts the pages.
  - –If the appropriate mobile pager receives the page, an ack is transmitted back to the radio tower.
  - –The radio tower relays the ack back to the base station and a *successful page* statistic is recorded.

OPNET Training

Summary

- Wireless models depend on dynamically changing parameters, which the simulation evaluates to determine the possible connectivity between all radio transmitters and receivers.
- Modeler simulate the movement of mobile nodes along trajectories and satellite nodes in orbit.
- The antenna, PDF, link and packet editors further expands your capability to define and customize parameters that affect the behavior of the model.

OPNET Training

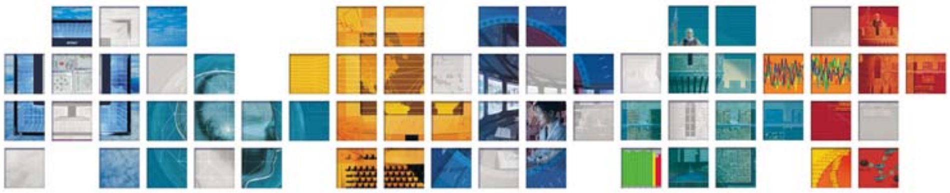
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

OPNET Technologies, Inc.

Making Networks and Applications Perform

Debugging Techniques



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

Goals

- Debug sensibly
- Detect problems early
- Diagnose problems quickly
- Introduce debugging aids
  - –Graphical OPNET Debugger (ODB)
  - –Animation
  - –Memory tracking
  - –Live Statistics
- Correct simulation errors

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Agenda

- OPNET Simulation Debugger
- Exploring simulation details in ODB
- Debugging packets
- Investigating memory bugs
- Error reporting in OPNET simulations
- Advanced ODB features

Debugging Strategy

- Systematic approach
  - –Limit amount of time “scanning” source code
- Looking for clues
  - –Compiler warnings
  - –DES Log
  - –Error Log
  - –OPNET Debugger (ODB)
  - –Memory tracking
  - –Animation and live statistics
- Narrow down the problem
  - –Time (simulation time, event number, packet number)
  - –Location (node, module, process, code segment)

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Bug Classifications

- Configuration
  - –Topology and attributes are not set up correctly
  - –“User error” or “usability error”
- Structural
  - –Simulation does not complete successfully
- Behavioral
  - –Does not accurately represent the developer’s intent
  - –Incorrect results, no results
- Memory
  - –Excessive memory usage
  - –Inefficient modeling approach
  - –Memory “leaks”

Simulation Kernels

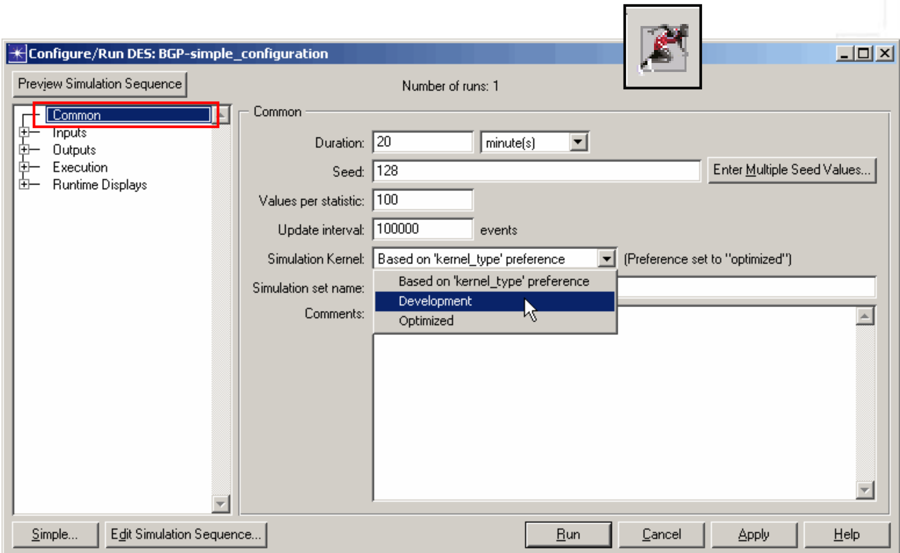
- Development Kernels: Use for development and debugging
  - –Same DES events
  - –Same results
  - –**OPNET Debugger (ODB)**
  - –Function call stack included in Error Log
  - –OPNET Profiler
  - –Slower simulation execution
  - –Slightly higher memory usage

- Optimized Kernels: Use for production results
  - –Same DES events
  - –Same results
  - –No function call stack in Error Log
  - –Error messages with less detail

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Choosing the Kernel Type



OPNET Training

OPNET Debugger (ODB) Capabilities

- Graphical display of simulation internals
- Interactive control of simulation
- Issue commands to obtain detailed information about events or objects
- Same events are executed in debug and non-debug simulations

OPNET Training

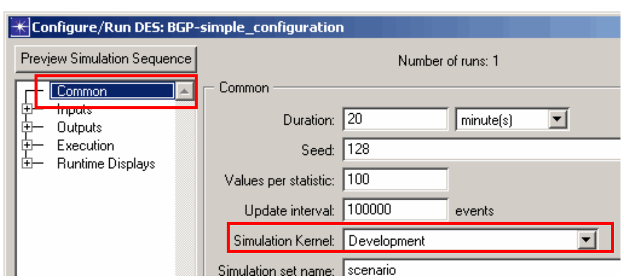
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

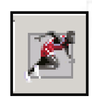
© Copyright 2007 OPNET Technologies, Inc.

ODB Features

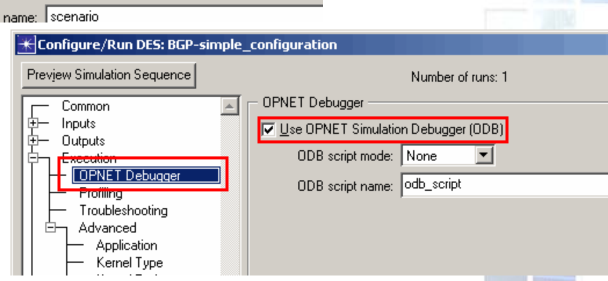
- Graphical exploration
  - –Network, node, process topology
  - –Packets, Events, Object Attributes details
  - –Breakpoints
  - –Animation
- Execute single and multiple events
- Set breakpoints for specific
  - –Events, Simulation Times
  - –Modules, Processes
  - –Packets
- Trace Kernel Procedures (KPs) as they are executed
- Print out current status of simulation entities
- Print out memory usage statistics
- Pass control back and forth with a C/C++ symbolic debugger

Simulations with ODB in GUI





Command line:
`op_runsim -debug`



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Refresher: Simulation Entities

- Each entity type has a unique ID in its domain
  - –Starting with zero
- Objects
  - –Cannot be created or destroyed at runtime
  - –E.g., nodes, links, subnets, demands…
  - –E.g., processors, queues, transmitters, streams…
- Events
- Packets
  - –Packet ID – unique for each packet
  - –Packets also have a tree ID, which marks copies of packets
- Interface Control Information (ICI)
- Processes
  - –Root processes may not be destroyed
  - –Dynamic (child) processes are created and destroyed at runtime

Refresher: Object References

- ID – unique integer assigned at the beginning of simulation
- Hierarchical name
  - –top.<subnet>.<subnet>.<subnet>…
  - –<node> or <link>
  - –<module>
- Path and demands are not hierarchical
- Examples

```
–top.usa.kansas.kansas_city.router.ip
–top.canada.alberta.edmonton.workstation2
–top.enterprise network.server <--> switch
```

Subnets Nodes Links Modules

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Getting Help in ODB

- **`help`**
  - –Lists available topics
- **`help <topic>`**
  - –Lists available commands for topic
- **`help <command>`**
  - –Lists usage of command

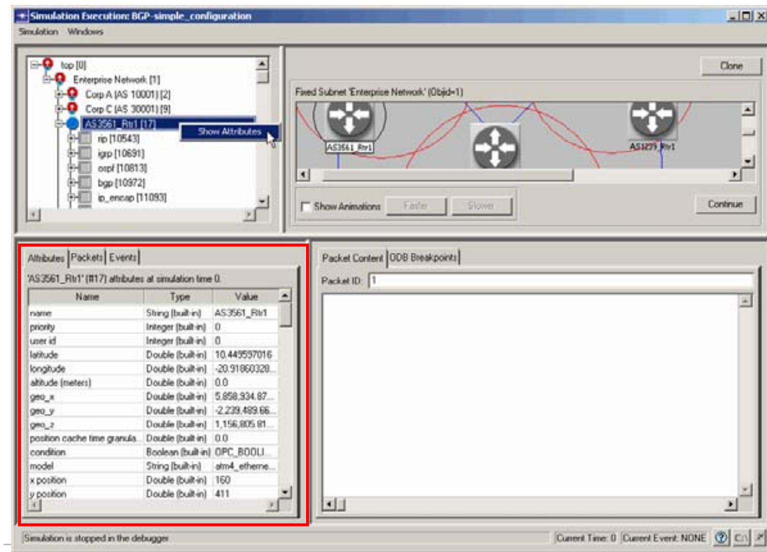
Lab: ODB and Simulation Execution

- Explore features of ODB and graphical simulation execution

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

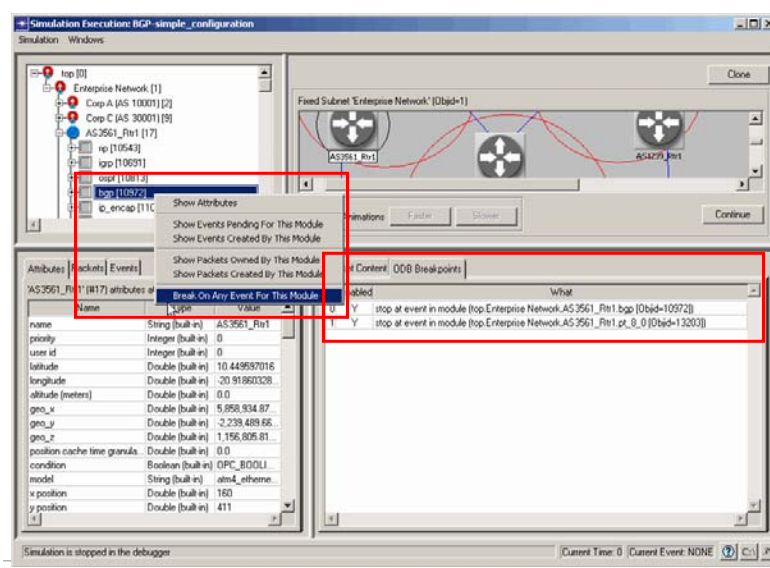
© Copyright 2007 OPNET Technologies, Inc.

Advanced ODB GUI: Attributes



OPNET Training

Advanced ODB GUI: Breakpoints

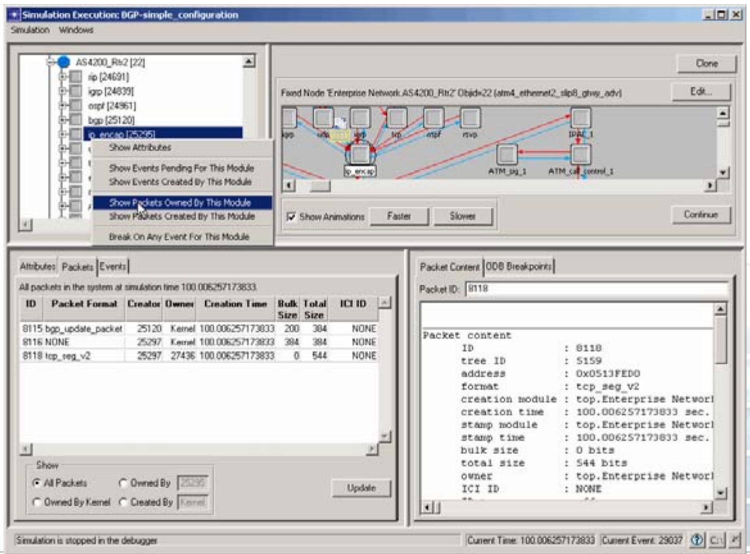


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

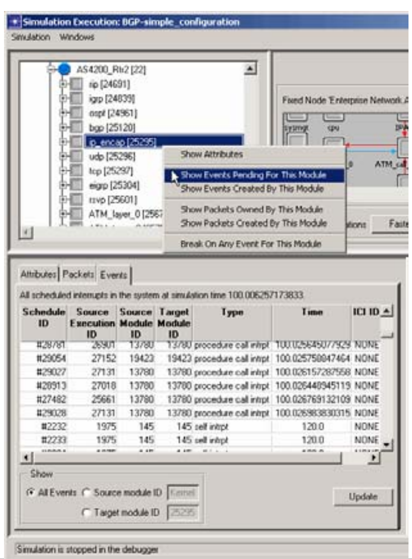
© Copyright 2007 OPNET Technologies, Inc.

Advanced ODB GUI: Packets



OPNET Training

Advanced ODB GUI: Events



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Lab:
Advanced ODB GUI

- Diagnose packet problem in simulation
- Utilize ODB animation

Traces

- Prints detailed information about event execution
- Modules, Processes, Packets, and Packet Trees
- Kernel traces display details of KPs
  - –KP name
  - –Argument names and values
  - –Return value
- Custom traces contain whatever you want
- Once activated, a trace remains active until deactivated

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Traces: Interpreting Trace Output

```
+- mobility_random_process_intrpt (mobile_node_ptr, code)
| +- mobility_random_waypoint_process_update (mobile_node_ptr)
| |
| | +- op_ima_obj_pos_get (node_objid, lat_ptr, long_ptr, alt_ptr, x_ptr, y_ptr, z_ptr)
| | |       object id         (3)
| | |       completion code   (success)
| | |       geodetic (lat, long, alt) pos  [0 0' 5. 9" (0.00), -0 0' 5.74" (-0.00), 0.0m]
| | |       geocentric (x, y, z) pos       [6,378,136.996m, -177.706m, 156.452m]
| | +-------
| | +- op_intrpt_schedule_call (time, code, proc, state_ptr)
| | |       time              (36)
| | |       code              (2)
| | |       procedure         (0x032BBE0B)
| | |       state ptr         (0x02CB7ED8)
| | |       event ID          (1589)
| | +-------
| |
| +-------
|
+-------
```


OPNET Training

Traces: Argument Information in Custom Functions

- KP Package **op_trace_** . . .
- Record argument values
- Just like KP traces
- No-op when not in ODB

```
void
my_func (int a, double b)
{
FIN (my_func (a, b));
op_trace_int32 ("a", a);
op_trace_double ("b", b);

. . .
op_stat_write (a_sh, (double) a);
. . .

FOUT
}
```

```
+- my_func (a, b)
|  a          (47)
|  b          (22.7)
|
|  . . .
|
| +- op_stat_write (…)
| |  stathandle (0x83de)
| |  value      (47.0)
| +----
|  . . .
+-------
```


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

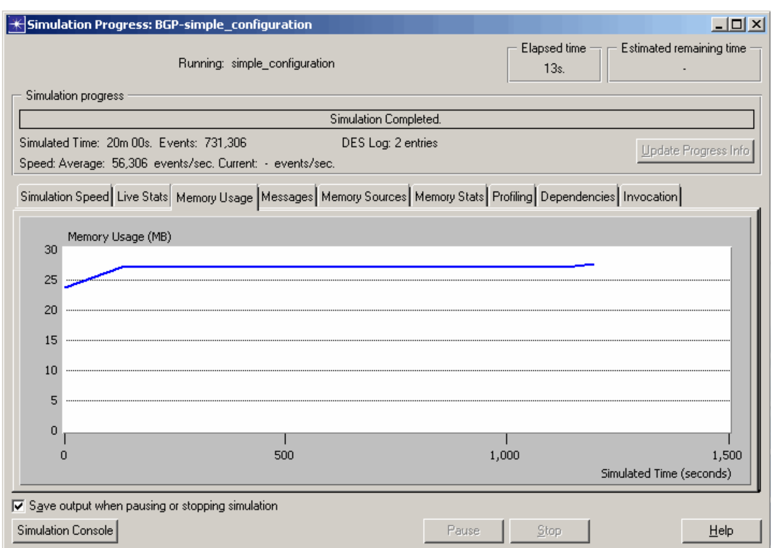
© Copyright 2007 OPNET Technologies, Inc.

Optional Lab: Packet Tracing in ODB

- Use packet tracing in ODB to investigate packet problem from Lab 2

OPNET Training

Overall Memory Usage

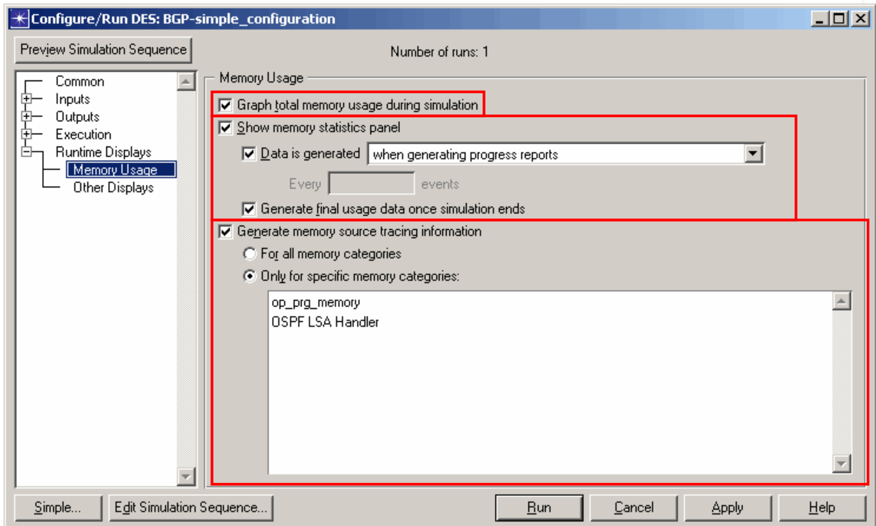


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Memory Usage Tools



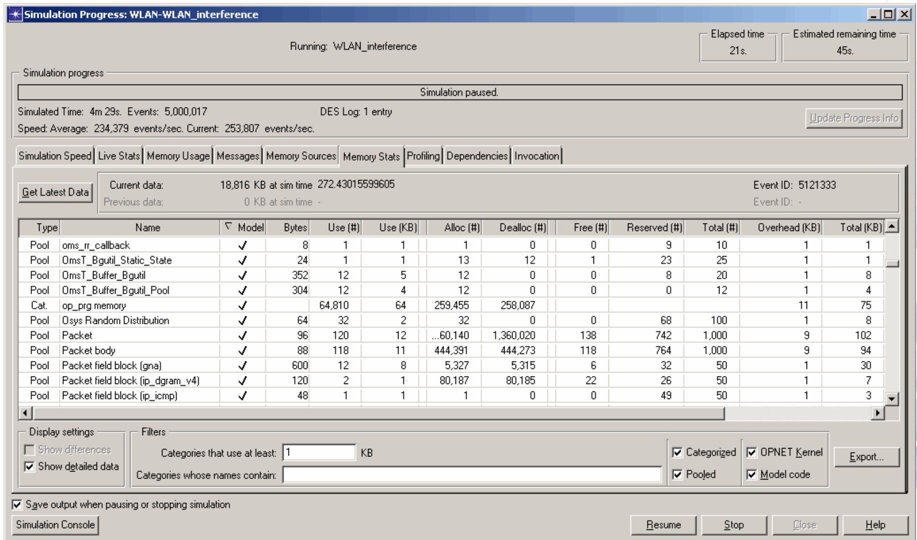
Memory Statistics

- Simulation Kernel tracks memory allocation by category
- Identify areas of memory buildup
- Leaks
  - –Example: Packets received but never destroyed
  - –Example: Structures allocated but never freed
- Non-Leaks
  - –Example: Modeling more traffic than the network can handle
  - –Example: Modeling large queues in a congested network
- Disable Memory Management Preferences
  - –**`mem_opt.compact_pools`**
    - If TRUE, do not have separate pools for categories of same size
  - –**`mem_opt.pool_small_blocks`**
    - If TRUE, use Pooled Memory for small blocks of dynamically allocated memory

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

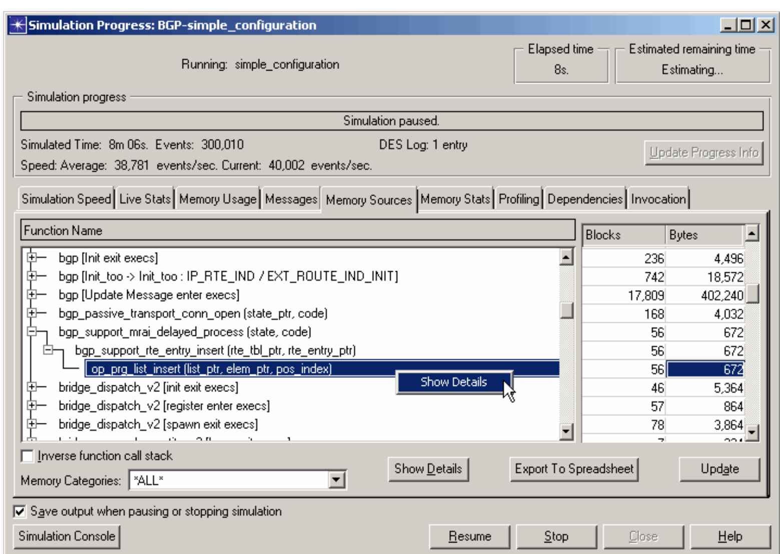
© Copyright 2007 OPNET Technologies, Inc.

Memory Statistics: Information



OPNET Training

Memory Sources: Tree View



OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

DES Log: What It Does

- Every scenario has its own DES Log
  - –Recreated after each simulation run
- Records notable events
  - –Not necessarily errors
  - –Describes configuration problems
  - –Contains model-related messages
  - –Suggests solutions
  - –Gives informational messages

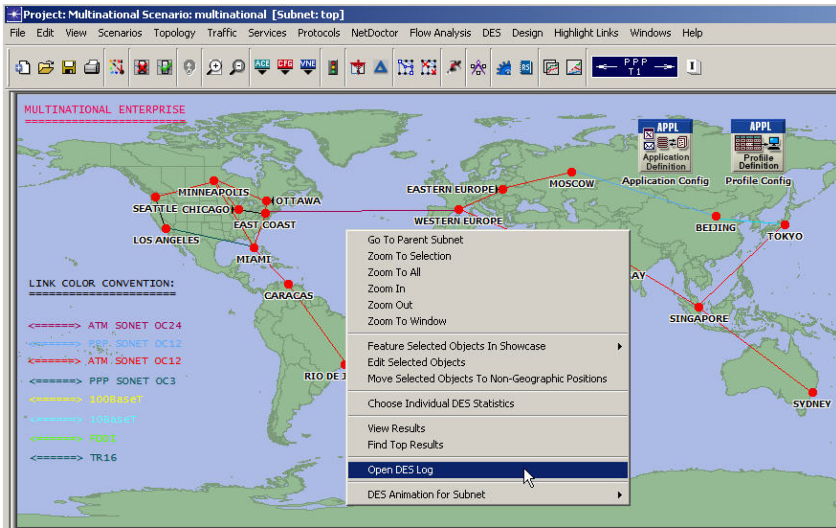
DES Log: Customizing

- Insert your own entries
  - –**`op_prg_log_handle_create ()`**
  - –**`op_prg_log_entry_write ()`**
  - –**`op_prg_log_entry_write_t ()`**
  - –**`op_prg_log_handle_severity_set ()`**
- Messages from these KPs will appear in the DES Log, too
  - –**`op_sim_error ()`**
  - –**`op_sim_end ()`**
- Many examples in OPNET Model Library

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

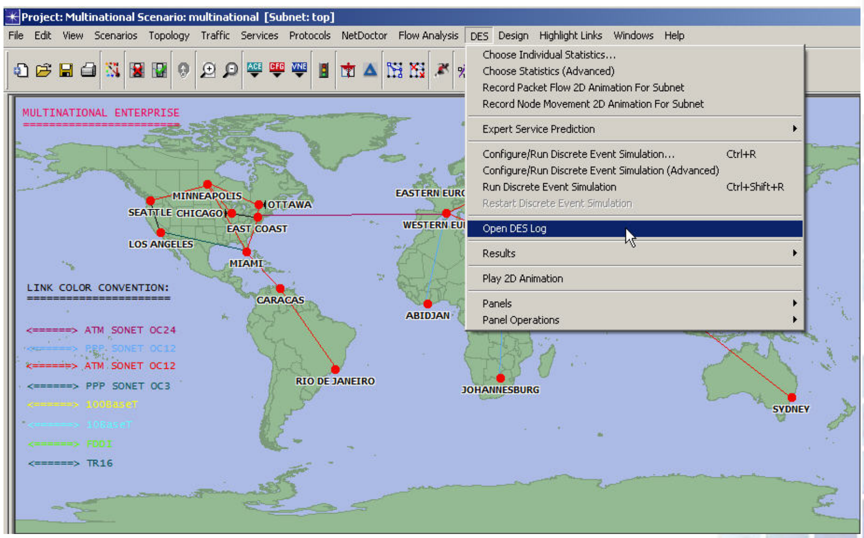
© Copyright 2007 OPNET Technologies, Inc.

Opening the DES Log



OPNET Training

Opening the DES Log (cont.)



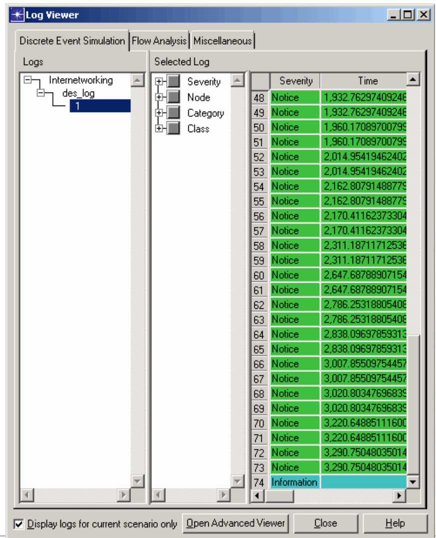
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

`printf()` vs. DES Log

- **`printf ()`**
  - –Not recorded permanently
  - –Best for “surgical” debugging
- DES Log
  - –Severity
  - –Simulation time
  - –Event ID
  - –Node
  - –Class/subclass
  - –Category
  - –Detailed message



OPNET Training

`printf()` vs. `op_prg_odb_print()`

- **`printf()`**
  - –Need to change code or configuration before finalizing models
  - –No easy way to turn desired output on or off
- **`op_prg_odb_print_<major/minor> ()`**
  - –No need to change code or recompile
  - –Automatically forces an event banner to be printed
  - –ODB-related KPs are no-ops in optimized kernels

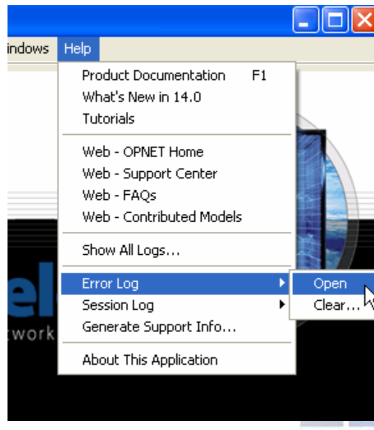
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Error Log

- Contains errors from OPNET kernel and all products
- Not project/scenario specific
- Use **op_sim_error ()** to add your own
- More details than console errors
- Includes error location
- Includes error time
- New entries at end of file
- Clear it out occasionally (**Help** / **Error Log** / **Clear**)
- **<home>\op_admin\err_log**
- **op_vuerr** utility



OPNET Training

Error Types

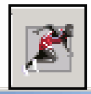
- Program Fault, Program Abort
  - –Stops the simulation
- Warning or Recoverable Error
  - –Does not stop the simulation, but may cancel current operation
  - –Preferences can be used to hide these errors (not recommended)
    - **rec_err_suppress**
    - **warning_suppress**
- Diagnostic
  - –Minor problem
  - –Does not cancel the current operation
  - –Hidden by default
  - –Visible only when **diag_enable** preference is set to **TRUE**

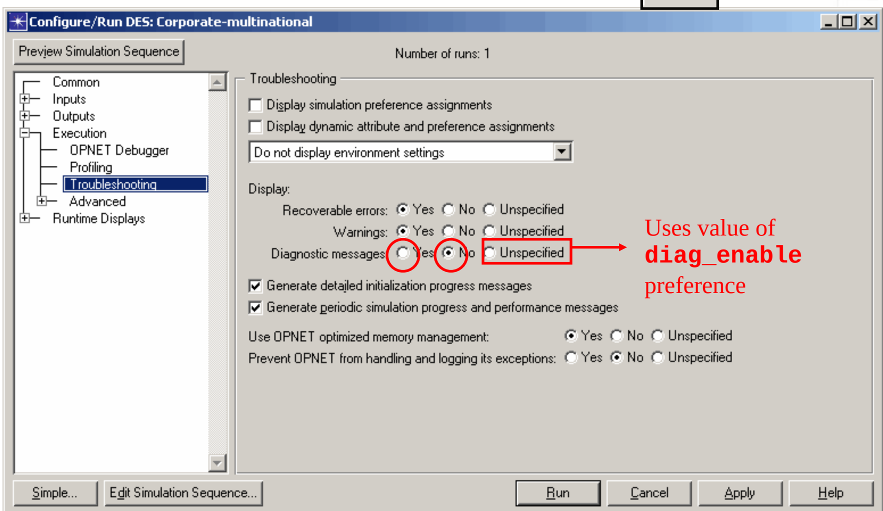
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Viewing Diagnostic Errors





Uses value of **diag_enable** preference

OPNET Training

Error Messages – Short View

- Messages tab of Simulation Sequence dialog box
- Console

```
|-------------------------------------------------------------|
| This node does not have any valid IP interfaces             |
| Assign a valid address to at least one interface            |
| T (0), EV (1593), MOD (top.DCI Network.FR_Cloud.ip),        |
| PROC (ip_dispatch_intf_table_create (total_interfaces))     |
|-------------------------------------------------------------|
```


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Function Call Stack (FCS)

```
<<< Recoverable Error >>>
  * Time:      10:45:50 Tue Jul 19 2006
  * Product:   modeler
  * Program:   op_runsim (Version 12.0.A PL0 Build 4701)
  * System:    Windows NT 5.0 Build 2195
  * Package:   process (ip_dispatch) at module (top.DCI Network.FR_Cloud.ip)
  * Function:  op_sim_error (gravity, line1, line2)
  * Error:     This node does not have any valid IP interfaces
               Assign a valid address to at least one interface
               T (0), EV (1593), MOD (top.DCI Network.FR_Cloud.ip),
               PROC (ip_dispatch_intf_table_create (total_interfaces))
```


OPNET Training

Function Call Stack (cont.)

```
* Function call stack: (builds down)
  ------------------------------------------------------------
        Call   Block
        Count  Line#  Function
  ------------------------------------------------------------
  0)       1    152  0xd84a6f08 [name not available]
  1)       1   2211  0x00004c00 [name not available]
  2)       1   1358  0x0000c400 [name not available]
  3)       1    291  m3_main
  4)       1   1106  sim_main
  5)       1   2706  sim_ev_loop
  6)    1565    531  sim_obj_qps_intrpt
  7)      69     15  ip_dispatch [wait -> cmn_rte_tbl :
                     SELF_NOTIFICATION / ip_dispatch_init_phase_2
   ()]
  8)       9   1791  ip_dispatch_init_phase_2 ()
  9)       9   7364  ip_dispatch_intf_table_create
   (total_interfaces)
 10)       1    649  op_sim_error (gravity, line1, line2)
  ------------------------------------------------------------
```


OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

FIN, FOUT, and FRET Macros

- How does OPNET know what to put in the function call stack?
  - –Process model states – automatic
  - –Need to add a macro for complete functions
    - Function Block
    - External C/C++ code
    - Pipeline Stages
- **`FIN`** - Macro used to label a function
  - –C: Place after local variables
  - –C++: First statement in the function
- When using **`FIN`** macro, replace return keyword with
  - –**`FOUT`** replaces **`return`**`;`
  - –**`FRET`**`(value);` replaces **`return`**`(value);`
- Using **`FIN`** without **`FOUT`** / **`FRET`** results in **Standard Function Stack Imbalance** error message
- No overhead when models compiled with **`comp_trace_info`** preference set to FALSE
- Macros required if using state variables

Block Line Numbers Refer to Last Curly Brace or `FIN` Passed in Code *Execution*

```
21 static void
22 ip_encap_pk_destroy (Packet* pkptr)
23        {
24        Ici*                          intf_iciptr;
25        IpT_Rte_Ind_Ici_Fields*       intf_ici_struct_ptr;
26
27        /** Destroys the IP datagram received from lower  **/
28        /** layer and the associated ip_rte_ind_v4 ICI    **/
29        FIN (ip_encap_pk_destory (pkptr));
30
31        /* Get the ICI associated with the packet. */
32        intf_iciptr = op_pk_ici_get (pkptr);
33
34        /* Destroy the ICI and its fields       */
35        if (intf_iciptr != OPC_NIL)
36                  {
37                  op_ici_attr_get (intf_iciptr,
38                            "rte_info_fields", &intf_ici_struct_ptr);
39                  ip_rte_ind_ici_fdstruct_destroy (intf_ici_struct_ptr);
40                  op_ici_destroy (intf_iciptr);
41                  }
42
43        /* Destroy the packet. */
44        op_pk_destroy (pkptr);
45
46        FOUT;
47        }
```


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

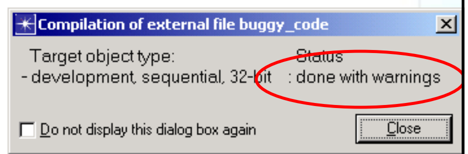
© Copyright 2007 OPNET Technologies, Inc.

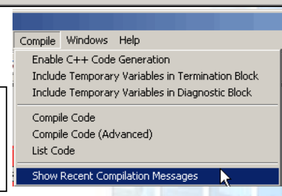
Compiler Warnings

- Results of kept in:
  - **<home>/op_admin/tmp/cc_err_<compiler process id>**

Compiler warnings can identify:

- Uninitialized variables
- Non-prototyped functions
- **'='** in test, or **'=='** not in test
- Statements with no effect
- **printf** type mismatches…






```
char * char_ptr;
strcpy (char_ptr, "hello world");
```

```
warning C4700: local variable 'char_ptr' used without having been initialized
```


OPNET Training

Lab:
Warnings and Errors

- Utilize the DES Log to identify simulation errors

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

ODB Labels

- Restrict traces to classes of activity
- Established by process model code
- Used in coordination with print statements…
  - –Values of state, temporary variables
  - –Results of statements
  - –Descriptions of protocol behavior… so that humans can read it!
- User-defined trace statements may be added to a process model
- OPNET Standard Model Library makes extensive use of labels

Labeled Traces: ODB Commands

• `acttrace`	reactivate trace
• `deltrace`	delete specified trace(s)
• `encaptrace`	enable/disable automatic tracing of encapsulating packets
• `exectrace`	enable/disable execution trace
• `fulltrace`	enable/disable global trace
• `interfacetrace`	set trace for a specified esys interface
• **`ltrace`**	**set trace for specified label**
• **`lmap`**	**display matching label information**
• **`mltrace`**	**set trace at module for specified label**
• `mtrace`	set trace at module
• `pktrace`	set trace for specified packet
• **`proltrace`**	**set trace for specified process and label**
• `protrace`	set trace for specified process
• `pttrace`	set trace for specified packet tree
• `status`	list breakpoints and traces
• `susptrace`	deactivate trace
• `tracefmt`	set trace output format

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Labeled Breakpoints: ODB Commands

- `actstop` reactivate breakpoint
- `cont` continue event execution until breakpoint
- `delstop` delete specified breakpoint(s)
- `evstop` set breakpoint for specified event
- `interfacestop` set breakpoint for a specified esys interface
- `intstop` set breakpoint at module for specified interrupt
- **`lstop` set breakpoint for specified label**
- **`mlstop` set breakpoint at module for specified label**
- `mstop` set breakpoint at module
- `pkstop` set breakpoint for specified packet
- **`prolstop` set breakpoint for specified process and label**
- `prostop` set breakpoint for specified process
- `status` list breakpoints and traces
- `suspstop` deactivate breakpoint
- `tstop` set breakpoint for specified time

Labels: Examples of a Labeled Trace

- Model code

```
if (op_prg_odb_ltrace_active (“tcp”))
  {
  op_prg_odb_print_major (
    "TCP process was unable to open a new connection.",
    "The Active Connections Threshold has been exceeded.", OPC_NIL);
  }
```

- In ODB, activate the trace

```
odb> ltrace tcp
odb> next
____________________________ (ODB 12.0.A: Event) ____________________________

    * Time   :  106.434795585 sec, [00d 00h 01m 46s . 434ms 795us 584ns 652ps]
    * Event  :  execution ID (3218), schedule ID (#3322), type (stream intrpt)
    * Source :  execution ID (3217), top.subnet.client.ip_encap (processor)
    * Data   :  instrm (0), packet ID (1179), ICI ID (627)
    > Module :  top.Logical Network client.tcp (processor) [process id: 129]
                |               TCP process was unable to open a new connection.
                |               The Active Connections Threshold has been exceeded.
```


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Labels:
Limit Traces to Objects of Interest

```
odb> lmap ping
ping    : <Diagnostic: ip_icmp>Prints the Ping configuration on the node.

odb> ltrace ping
```

Print *all* ping-related traces

```
odb> mltrace top.INET_CLOUD.RTR1.ip ping
```

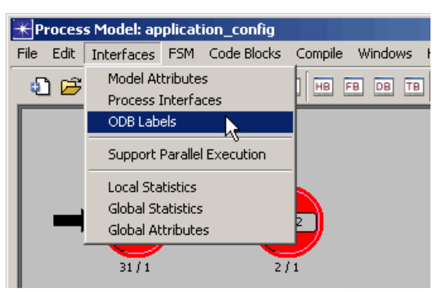
Print ping-related traces for *this module*

```
odb> promap top.INET_CLOUD.RTR1.ip

______________________________________________________
Active Processes for Module (698)
        process ID     process model       process tag
        ----------     -------------       -----------
        14             ip_dispatch         (none)

odb> proltrace 14 ping
```

Print ping-related traces for this *process*

Labels:
Discovering Defined Labels

- Easy way
  - –Process Editor : **Interfaces** / **ODB Labels**
  - –Up to model developer to fill this in
  - –ODB command: **`lmap`**



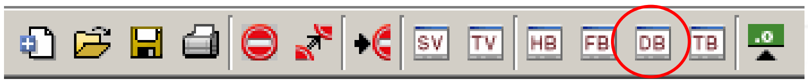
- Manual method: search for **`op_prg_odb_ltrace_active`**
  - •UNIX: **`grep`**
  - •Windows: Find in files or the **`find`** command

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Diagnostic Block

- Written in process model
  - –**DB** button
  - –**Code Blocks** / **Diagnostic Block**



- Invoked in ODB
  - –ODB is the only way to invoke it
  - –Can invoke for any process, not just active ones
- Statements that display state information of interest
  - –Values of state variables
  - –Contents of key data structures
  - –Whatever you want!

Diagnostic Block: ODB Commands

•**`prodiag`**	execute diagnostic block of process
•**`proldiag`**	execute diagnostic block with trace on label

```
if (op_prg_odb_ltrace_active (“my_process_svars”))
    {
    char str1 [100], str2 [100];

    op_prg_odb_print_major (“My state variables’ values are:”, OPC_NIL);

    sprintf (str1, “Left-hand value:\t(%s)”, lh_val);
    sprintf (str2, “Right-hand value:\t(%s)”, rh_val);

    op_prg_odb_print_minor (str1, str2, OPC_NIL);
    }
```


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Diagnostic Block: Workflow

1. Find object ID of module containing the process of interest
   - –Use **`objmap <type> <name>`** to obtain a list of selected objects
   - –E.g., **`objmap proc rip`** gives the list of all processors with the name "rip"
   - –Locate desired node and note module ID of interest
2. Find object ID of the process of interest
   - –Use **`promap <objid>`**, where objid is the module of interest
   - –E.g., **`promap 104`**
3. Run the simulation in debug mode; stop it at some time
4. Print the information from the diagnostic block
   - –E.g., **`proldiag 42 rip_route_table`**

OPNET Training

Starting ODB from Command Line

- Example use:

```
op_runsim -net_name project1-scenario1 –debug
```

- Why run from command line?
  - –Memory not taken up by Modeler
  - –Create your own scripts/batch files
  - –Control-C interrupt to go to ODB prompt after event completes

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Take-Away Points

- Debug sensibly
- Detect problems early
- Diagnose problems quickly
- Use debugging aids
  - –Compiler warnings
  - –DES Log
  - –Error Log
  - –OPNET Debugger (ODB)
  - –Memory tracking
  - –Animation and live statistics
- Goal: Correct simulation behavior

OPNET Training

OPNET Technologies, Inc.

Making Networks and Applications Perform

Publishing and Animation

OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Agenda

- Conceptual Goals
  - –Exporting Data
  - –Desktop Publishing
  - –Animation
  - –Preferences
- Tool Goals
  - –Print Reports
  - –Print Graphics
  - –Capture Screen Images
  - –op_vuanim Utility
  - –Editing Preferences

Publishing Results

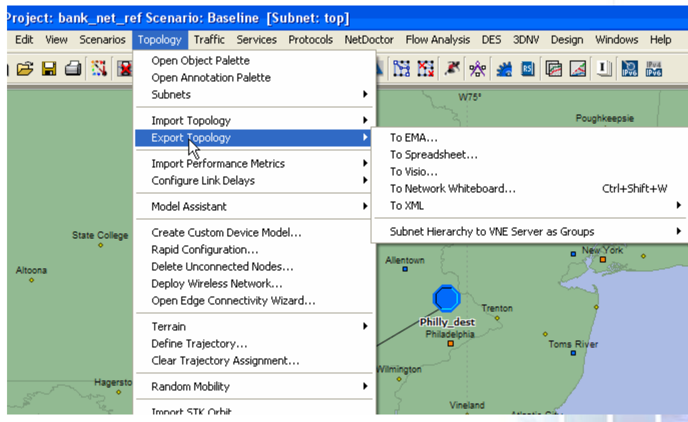
- After building a model, running simulations, and analyzing results, you will often have to report the project to others.
- It is important to organize, document, and archive the evolution and results of a continually changing model.
- Modeler has several output options so you can create graphics, detailed reports, and output files.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

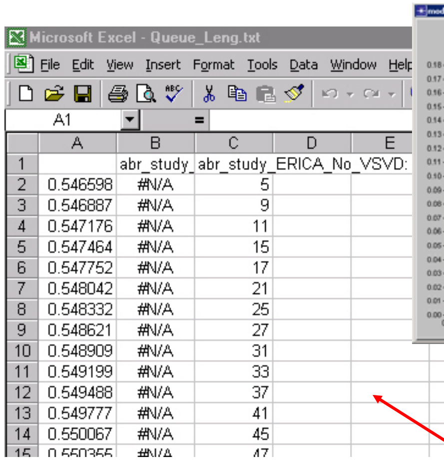
Exporting Data

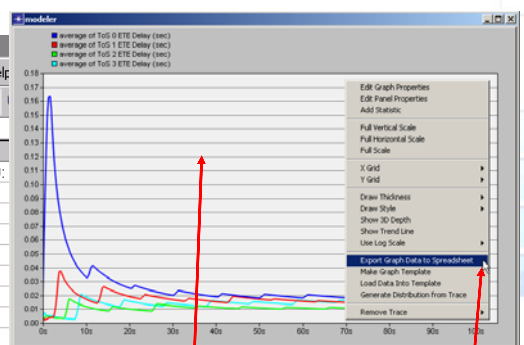
- Modeler allows you to export the network topology in a variety of different methods.
  - –EMA
  - –Spreadsheet
  - –Visio
  - –XML
  - –HTML
  - –Bitmap



OPNET Training

Exporting Data - Spreadsheets





- User right clicks inside graph on desired panel.
- Selects “Export Graph Data to Spreadsheet”.
- Modeler launches the appropriate program with data.

OPNET Training

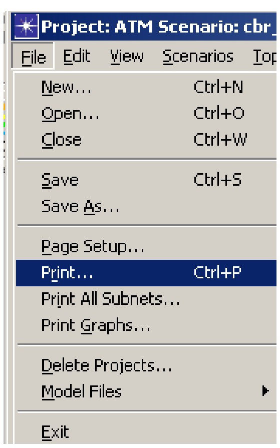
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

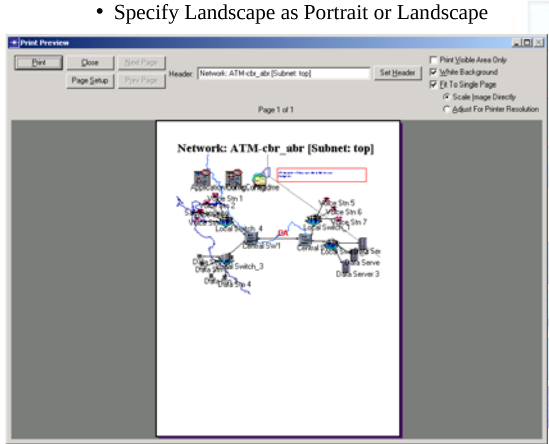
Desktop Publishing

- There are four types of operations for publishing OPNET information:
  - –Printing model graphics
  - –Capturing bitmap graphics
  - –Export to HTML
  - –Printing text files and contents of edit pads

Printing Model Graphics



- Left-click on “File”, “Print…” or “Print All Subnets…”.
- Specify Landscape as Portrait or Landscape

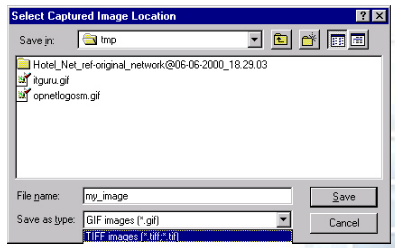


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

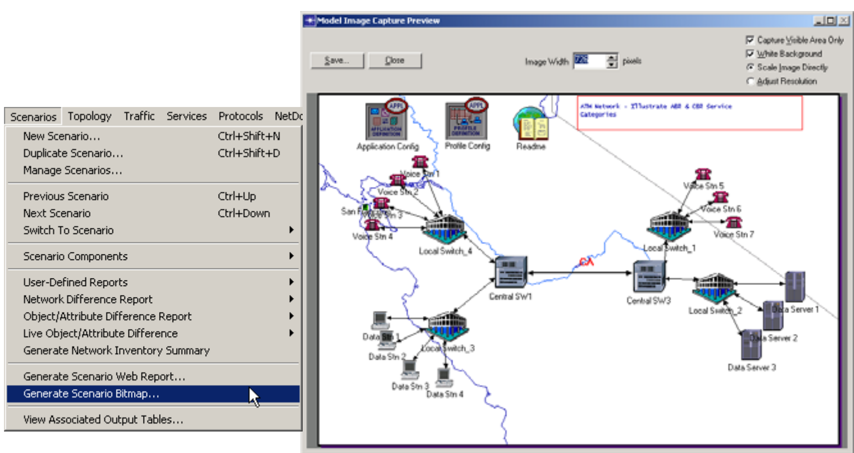
Capturing Bitmap Graphics

- Left-click on “Topology”.
- Select “Export Topology / To Bitmap …”.
- Files are saved in <HOME>\op_admin\tmp.
- Files can be saved as either:
  - –GIF images (*.gif)
  - –TIFF images (*.tiff, *.tif)



- An example is shown on the following slides.

Capturing Bitmap Graphics



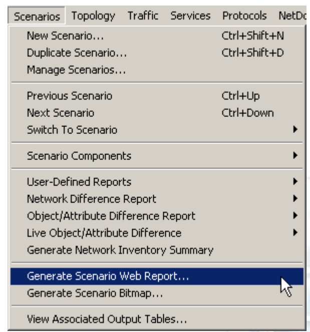
CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Exporting to HTML

- Exporting to HTML will create screen captures of your network model and allow you to navigate the topology in a web browser.

- Save the HTML files to the desired location.

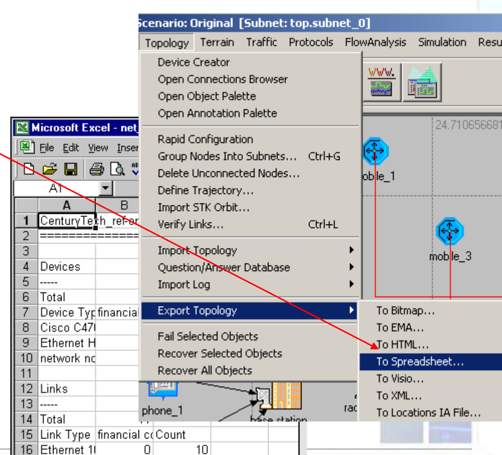


Exporting to Spreadsheet

- Exporting your network to a spreadsheet will provide a network summary that includes the type and number of links and nodes in your network model, and financial cost information.

- Under the "Topology" menu, select "Export Topology", then choose "To Spreadsheet…".

- Your default spreadsheet program will be launched with network summary data.



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

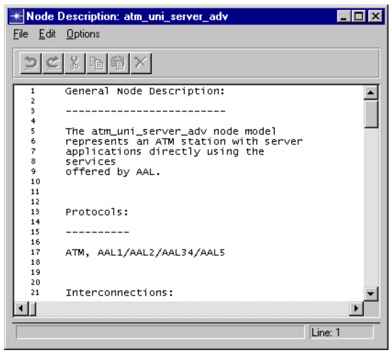
© Copyright 2007 OPNET Technologies, Inc.

Printing Text Files

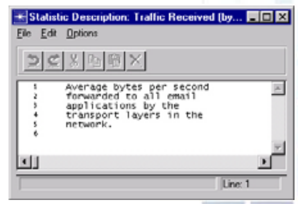
- The description windows in OPNET can be saved to file and printed.
  - –Select **File / Print** to print a description.
- Examples:

The information in these windows can be written to a user named text file or sent directly to a printer.

Node description:



Statistic description:



Animation

- An animation can be viewed using the OPNET utility, op_vuanim.
- This capability is valuable in better understanding the behavior of the model and it is an excellent tool for debugging models.
- There are two types of animation in OPNET.
  - –Automatic
  - –Statistic

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

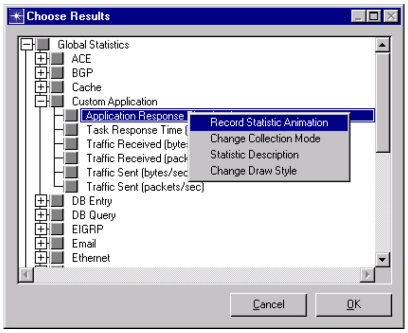
© Copyright 2007 OPNET Technologies, Inc.

Animation - Automatic

- Automatic animation displays node and packet movement.
- This is an effective way to graphically depict the movement of traffic throughout your network.
- To collect packet flow animation:
  - –Open the desired project-scenario.
  - –Enter the desired subnet.
  - –Right-click on the project workspace and select “Record Animation.”
  - –Run the simulation.

Animation - Statistic

- Statistic animation allows you to view the statistics in a graph format as they are being collected.
- In “Choose Statistics”, right-click the Statistic and select “Enable Animation”.



CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

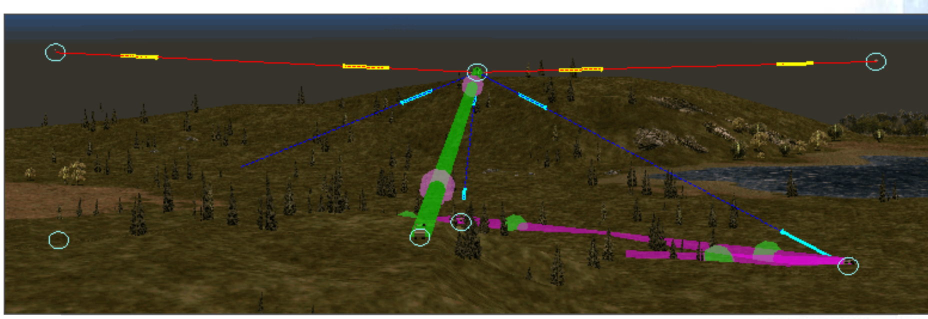
© Copyright 2007 OPNET Technologies, Inc.

Animation

- Open project, “anim_lab”.
- Right click the project workspace and select “Record Animation.”
- Configure the simulation for 50 seconds.
- Run a simulation
- From the “Results” pull-down menu, select “Play Animation” and wait for the Animation viewer to come up.

Animation: 3DNV

- 3D Network Visualizer (3DNV)
- Separate OPNET product
- Visualizes
  - –Network Topology
  - –Communications Effects
  - –Node Movement

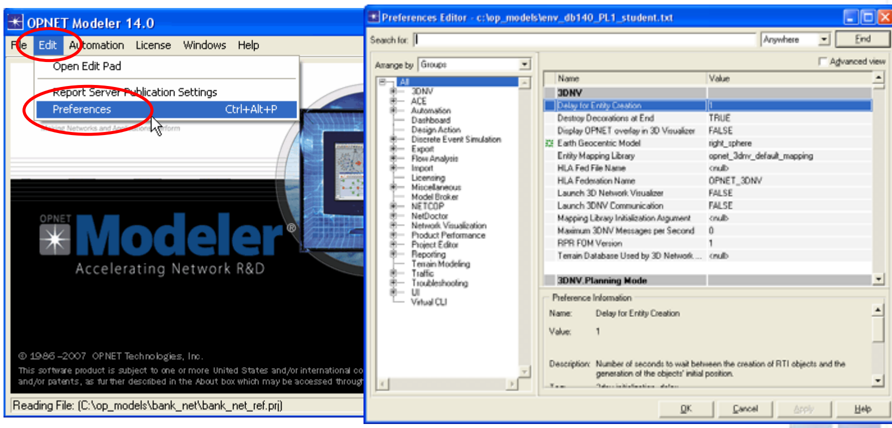


CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Edit preferences

- Instructor demo: Instructor will demonstrate the edit preferences utility in OPNET.



Summary

- Modeler provides you with various utilities for documenting and presenting the results of models.

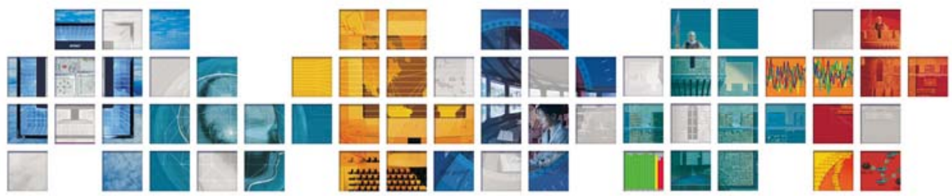
- Animation allows you to see a dynamic representation of the behavior of a model and acts as a helpful tool in debugging simulations.

- Preferences allow you to customize OPNET to meet specific needs.

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Product Modules



OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET® Training

Objectives

- In this chapter you will learn
  - MVI Module :
    - Import Topologies and Traffic from a Variety of Sources
  - –Wireless Module:
    - Transceiver Pipeline
  - –Terrain Modeling Module:
    - Contour Elevations Maps
    - Selecting Propagation Models
  - –ACE Module

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

eXpress Data Import Module (XDI)

- Additional Modeler feature purchased separately
- Allows import of topology and traffic information from other sources
- Topology import source
  - –Device configurations
- Traffic import sources
  - –NetScout (analytically modeled traffic)
  - –Cisco NetFlow
  - –Network Associates' Sniffer
    - •ExpertSniffer Traffic Archives (analytically modeled traffic)
  - –Concord Network Health
    - •Background Utilization values
  - –MRTG

OPNET Training

Wireless Module

- The Wireless Module is an add-on component for Modeler that allows you to build wireless network models.

- Additional node types included with the Wireless Module are mobile and satellite nodes.

  - –Mobile nodes can be used to represent vehicles or people that move during the course of a simulation.
  - –You can assign a trajectory to a mobile node, or enter values for a bearing, ground speed, and ascent rate.
  - –Satellite nodes represent communications satellites that follow a specified orbit.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Transceiver Pipeline

- Wireless communications can have complex and dynamically changing parameters that can affect the transmission and reception of a packet.
- Dynamically changing parameters may include node location, interference, antenna power, etc.
- OPNET models the complexity of communication links as multiple computation stages, called the Transceiver Pipeline, that determine whether a transmitted packet will be received.
  - The Wireless Transceiver Pipeline consists of 14 stages, and each stage computes a specific element of the connection between a given transmitter and an eligible receiver.
  - The various stages calculate such values as transmission delay, propagation delay, transmitter antenna gain, received power, interference, etc.

Terrain Modeling Module

- The Terrain Modeling Module is an add-on component for OPNET Modeler that allows you to take into account terrain and environmental effects when modeling wireless networks.
- With TMM, you can import elevation maps that contain terrain data.
- Supported map formats are DTED and DEM.
- TMM also allows you to select and compare signal loss from various propagation models
  - Free Space
  - Longley-Rice
  - TIREM (requires TIREM Module)

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

ACE (Application Characterization Environment)

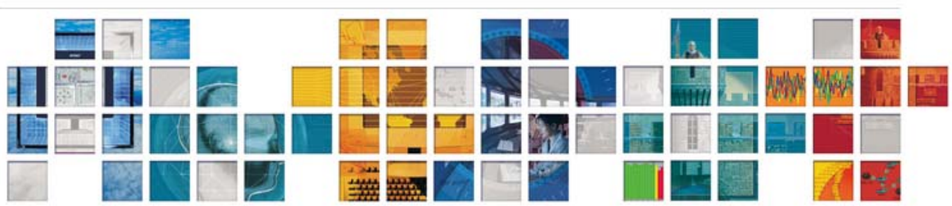
- ACE can characterize custom networked applications based on a packet trace file.
- ACE allows you to visualize, analyze, diagnose, and simulate an application.
- You can import an ACE characterized application into Modeler and play back the application task in a simulation environment.

OPNET Training

OPNET Technologies, Inc.

Making Networks and Applications Perform

OPNET Development Kit



Accelerate OPNET Modeler Projects

OPNET Confidential – Not for release to third parties
© 2007 OPNET Technologies, Inc. All rights reserved. OPNET and OPNET product names are trademarks of OPNET Technologies, Inc.

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Accelerate OPNET Modeler Projects with the OPNET Development Kit (ODK)

- Key application areas:
  - –Programmatically build network topologies
  - –Construct custom workflows to facilitate model configuration
  - –Customize network visualization
  - –Create custom workflows to selectively enable Modeler functionality

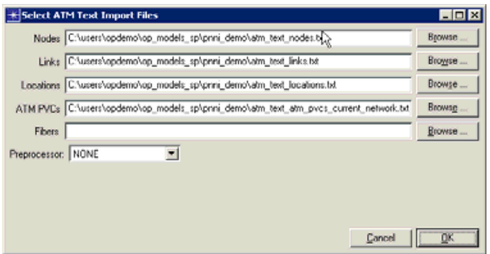
ODK Overview

- C/C++ APIs
- "Built-in" application support packages
  - –Network data import/export
  - –OPNET Advanced Graphical Interface
  - –ODK Information Modeling
  - –Intelligent hardware configuration
  - –Optimization engine
  - –Rules-based analysis
  - –Wizard-driven topology creation
  - –Report generation and export

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Programmatically Build Network Topologies



ODK enables a user to develop custom mechanisms for importing topology, traffic and configuration data from text files, databases, spreadsheets, etc.

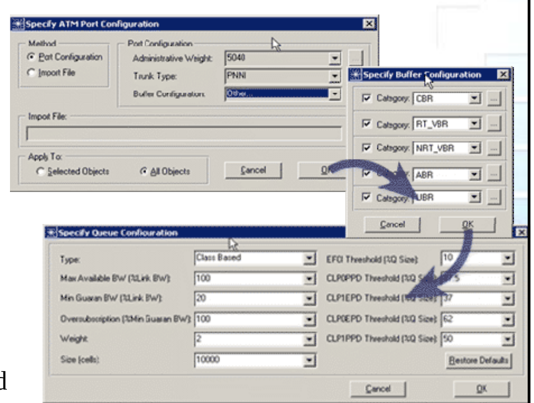
Example uses of ODK in IT Guru, SP Guru, and Modeler products

- –"Rapid Configuration"
- –"Group Nodes into Subnets"
- –Import Topology from ATM Text Files

OPNET Training

Construct Custom Workflow to Facilitate Model Configuration

Use ODK's dialog box editor and APIs to construct workflows that simplify the configuration process



Example uses of ODK in IT Guru, SP Guru, and Modeler products

- –"Create DES Application Demands"
- –"Configure Routing Protocols"
- –"Configure VLANS for Selected Nodes"
- –Active Attributes

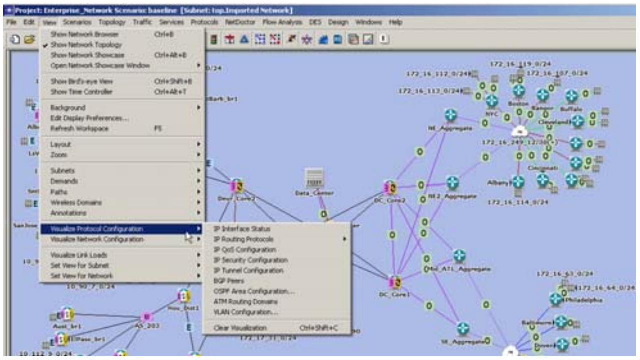
OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Customize Network Visualization

ODK can be used to define custom visualizations to provide a quick “top level” view of network configuration



Example uses of ODK in IT Guru, SP Guru, and Modeler products

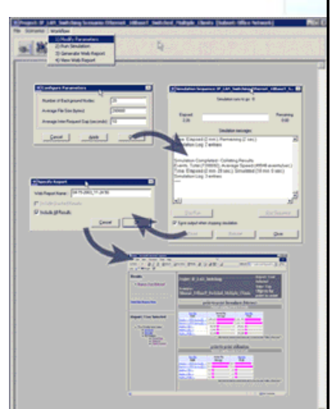
- “Color Links by Utilization”
- “Visualize Routing Domains”

361

OPNET Training

Create Custom Workflow to Selectively Enable Modeler Functionality

Tailor the UI to narrow access to required features, and customize workflows to support the target use cases



- Customized network management applications
  - –Slimmed down version of Modeler
  - –Spectral optical design solution

Example uses

- Spectral optical design solution
- Many consulting projects

362

OPNET Training

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

ODK Example: Highlighting T1 Links

- Changes color and thickness of T1 links to highlight them
- Developed using ODK, to be used in Modeler
- Lab 10 in Lab Manual

Wrap-Up: Questions

- We open up the class for questions and discussion on any items we discussed or issues that we did not discuss.
- If you think of any questions after the end of this class, check the documentation, the FAQs on our web site, or contact us via Technical Support (telephone (240-497-1200), fax (240-497-1064)), email (support@opnet.com)

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE, FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.

Wrap-Up

- OPNET extends our thanks to all the students for their time and attention.
- This is a dynamic course in that we are constantly seeking to improve it to best present the material and to better instruct our students.
- You may download the course material (Powerpoint slides, lab manual, and model files) from:
  - ftp://opftp@ftp.opnet.com
  - password: CaNeCo+
  - Go to pub/training/current folder
- Please fill out the student evaluations in the beginning of this manual. Your constructive criticism and suggestions assist us in increasing the quality of this training and OPNET Modeler.

Wrap-up

- OPNET appreciates your business and wishes you luck in your modeling endeavors!

CONFIDENTIAL INFORMATION: DO NOT DISCLOSE,
FORWARD, DISTRIBUTE, SHARE, OR MAKE COPIES OF THIS
DOCUMENT IN WHOLE OR IN PART.

© Copyright 2007 OPNET Technologies, Inc.