Overview of OPNET Modeler

Xavier Pérez Costa

NEC Network Laboratories

Heidelberg, Germany

Empowered by Innovation NEC

Outline

Introduction

- **OPNET Technologies, Inc.**
  - Corporate Overview
    Founded in 1986
    Approximately 300 employees
    Worldwide coverage via direct offices and channel partners
- **Solutions for Enterprises** 
- **Solutions for Service Providers**

- **Solutions for Network R&D Organizations**

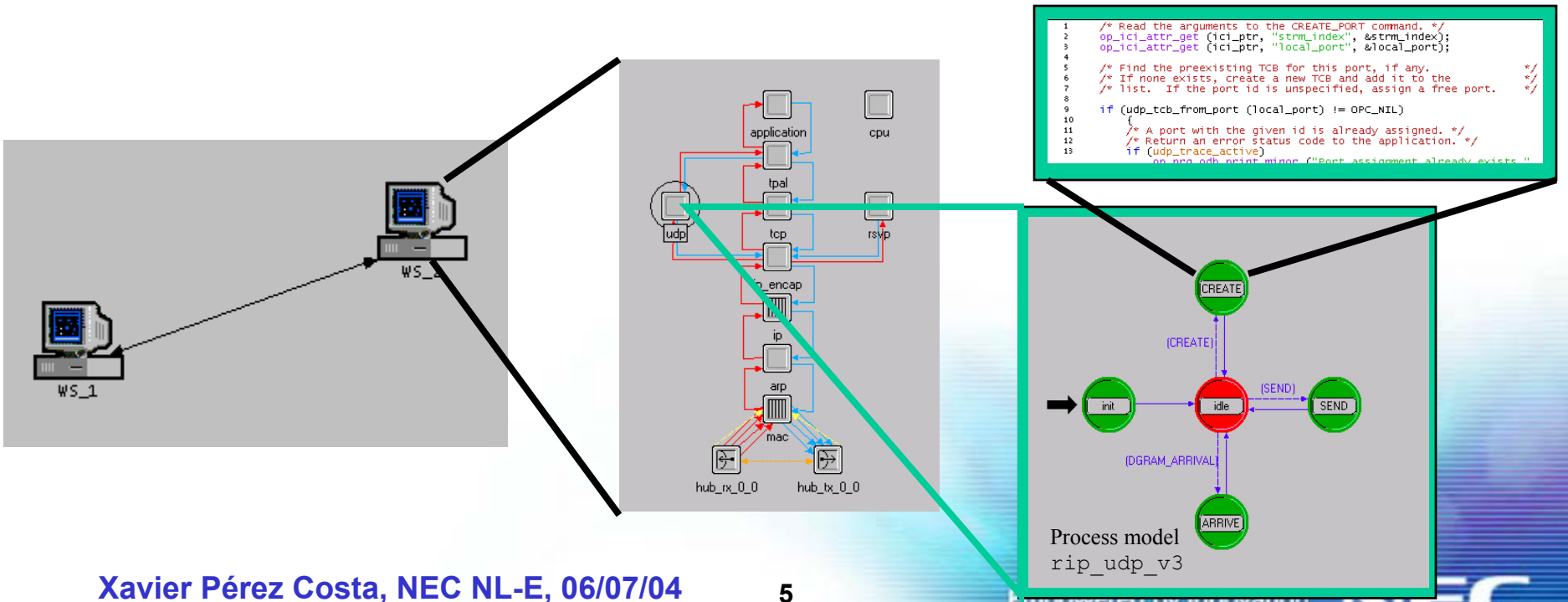
- **OPNET Modeler**
  - Modeling and simulation environment for designing and analyzing communication protocols, network equipment, and end-to-end systems.

Usage Prerequisites

- **Use of provided models**
  - Ability to understand C or C++
  - Basic understanding of networks
- **Modification or Development of new models**
  - C or C++ coding skills

Simulator Structure: **The Three-Tiered OPNET Hierarchy**

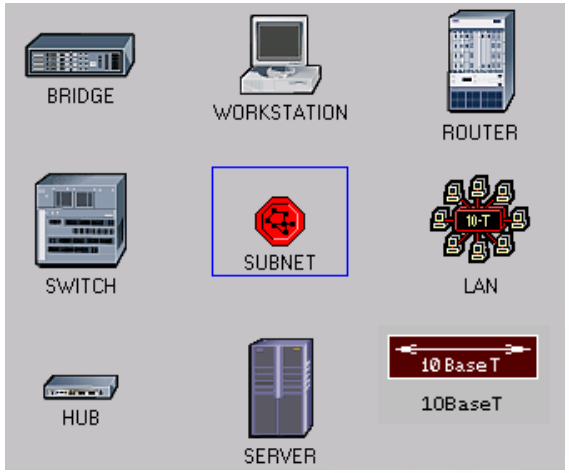
- **Three domains: network, node, and process**
- **Node model specifies object in network domain**
- **Process model specifies object in node domain**



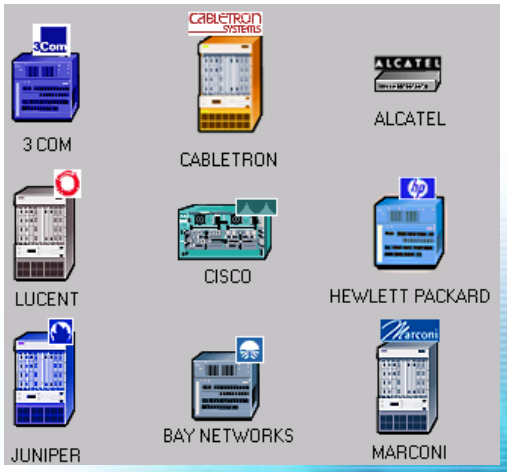
Simulator Structure: **Network Domain: Network Objects**

- **Network models consist of *nodes*, *links*, and *subnets***
- **Nodes represent network devices and groups of devices**
  - Servers, workstations, routers, etc.
  - LAN nodes, IP clouds, etc.
- **Links represent point-to-point and bus links**
- **Icons assist the user in quickly locating the correct nodes and links**
- **Vendor models are distinguished by a specific color and logo for each company**

Generic Devices

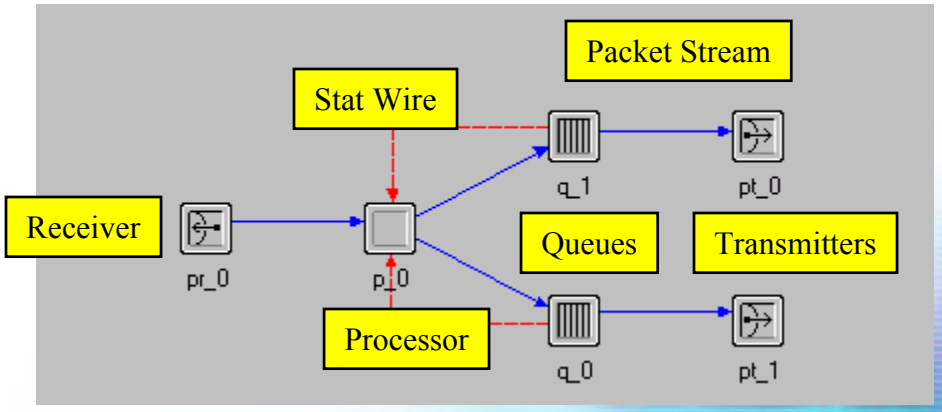


Vendor Devices



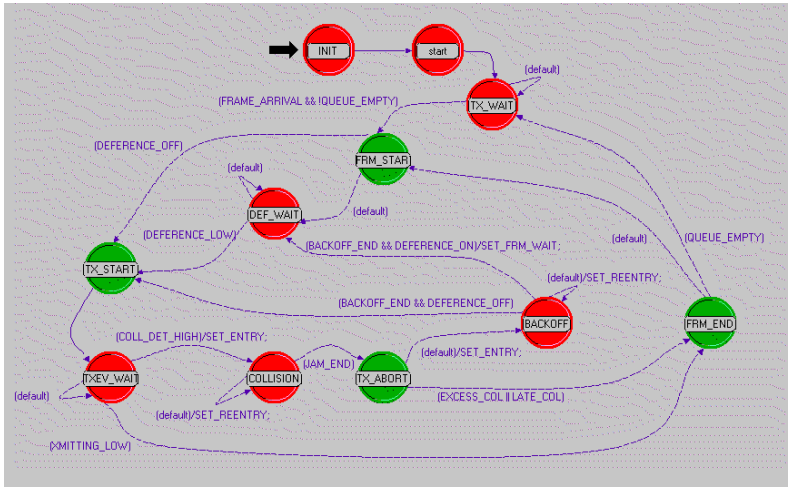
Simulator Structure: **Node Domain**

- **Basic building blocks (modules) include processors, queues, and transceivers**
  - Processors are fully programmable via their process model
  - Queues also buffer and manage data packets
  - Transceivers are node interfaces
- **Interfaces between modules**
  - Packet streams
  - Statistic wires



Simulator Structure: **Process Domain**

- **OPNET process models consist of:**
  - State transition diagrams
  - Blocks of C code
  - OPNET Kernel Procedures (KPs)
  - State variables
  - Temporary variables



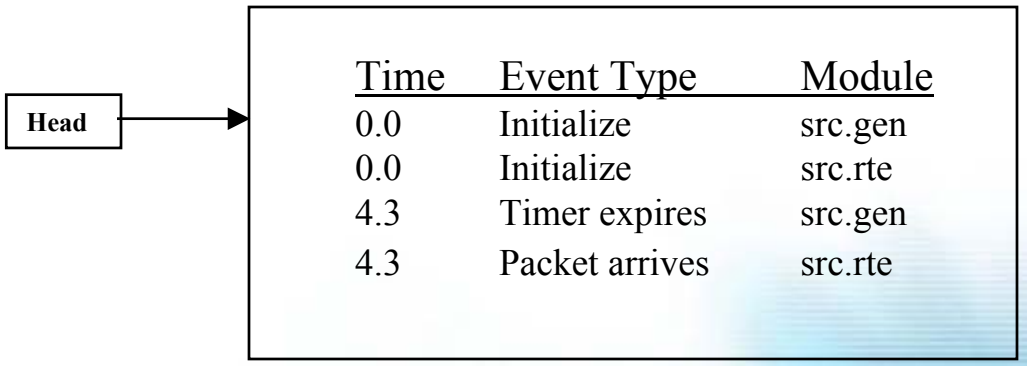
- **A process is an instance of a process model**
- **Processes can dynamically create child processes**
- **Processes can respond to interrupts**

Simulator Structure: **Simulation Output**

- **Three kinds of output**
  - Vectors
    - List of time-value pairs
  - Scalars
    - List of values dependent on parametric input
    - Not plotted vs. time
  - Animations
    - Packet flows
    - Node movements
- **Objects have pre-defined statistics**

Simulator Internals: **Event List Concepts**

- **Single global event list**
- **Shared simulation time clock**
- **Events scheduled in time order**
- **Event removed from event list when it completes**



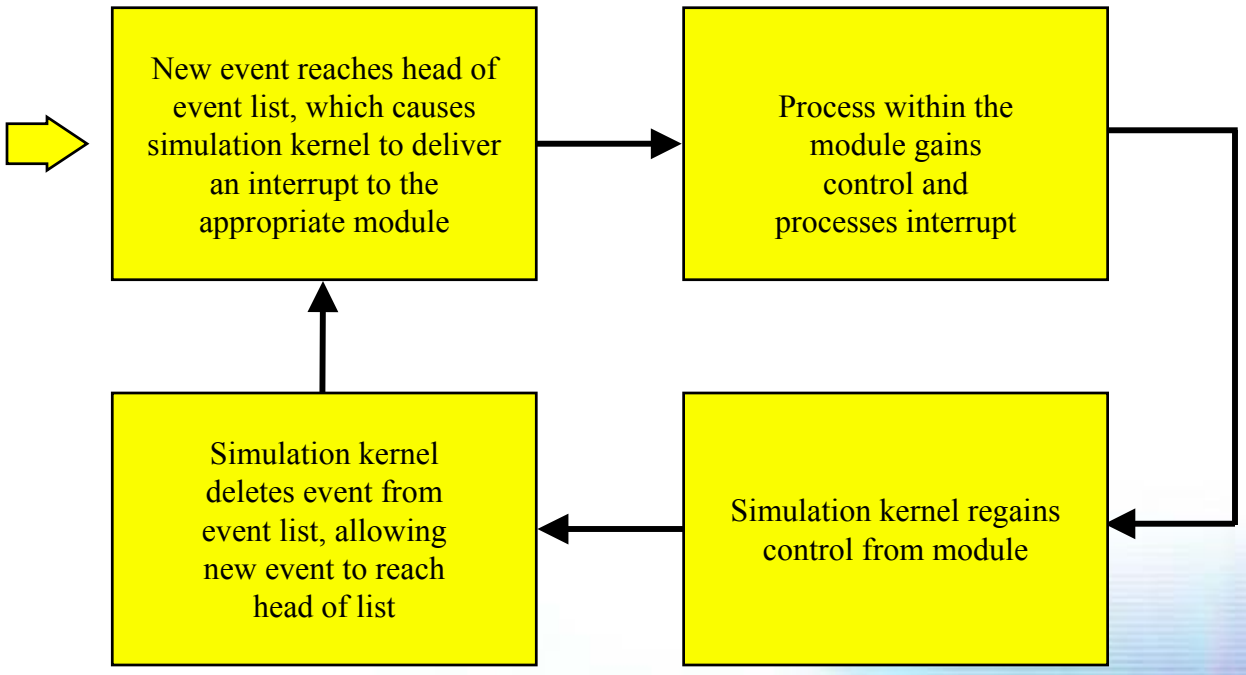
Simulator Internals: **The Simulation Kernel**

- **Simulation kernel (SK) manages the event list**
- **SK delivers each event, in sequence, to the appropriate module**
- **SK receives requests from processes and inserts new events in the event list**

Simulator Internals: **Interrupts**

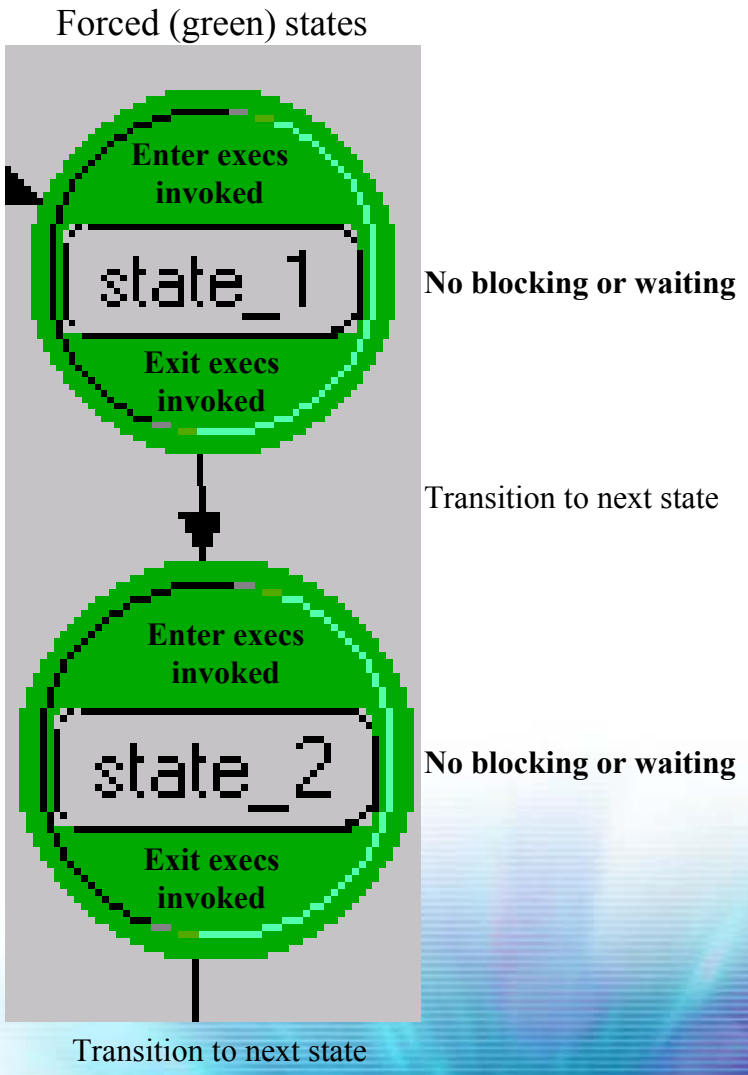
- **First event in the event list becomes an interrupt**
- **Delivered by the simulation kernel to the designated module**
- **Data associated with the event can be obtained by the module**
- **Processors and queues can have BEGSIM interrupts**

Simulator Internals: **How Does the Event List Work?**



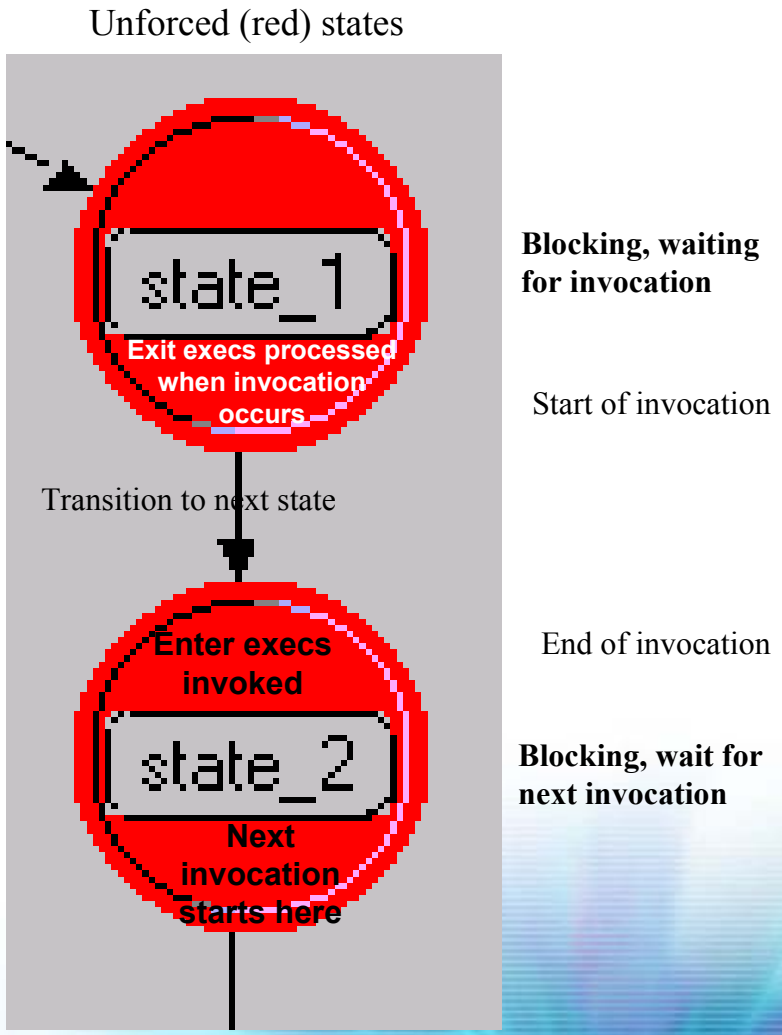
Simulator Internals: **Forced States**

- **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
- **OPNET convention: code in enter execs only**



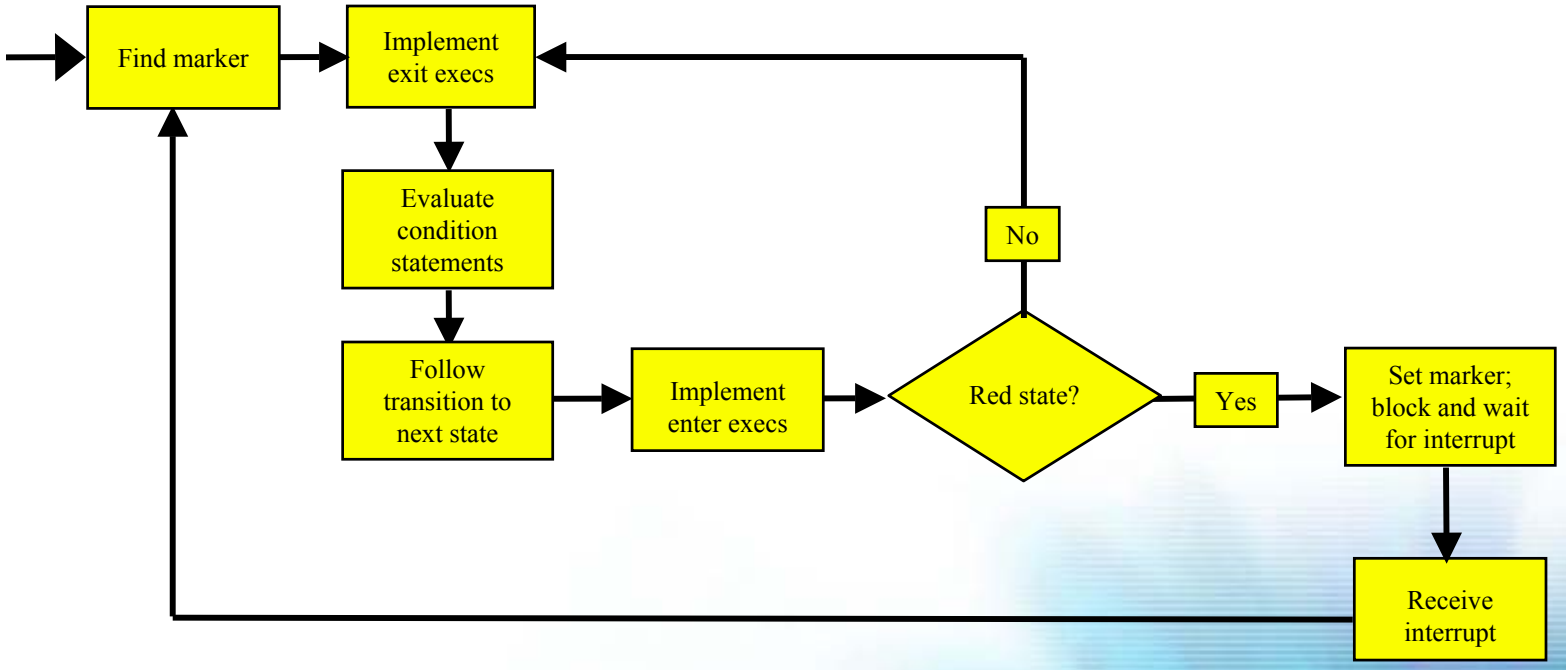
Simulator Internals: **Unforced States**

- **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



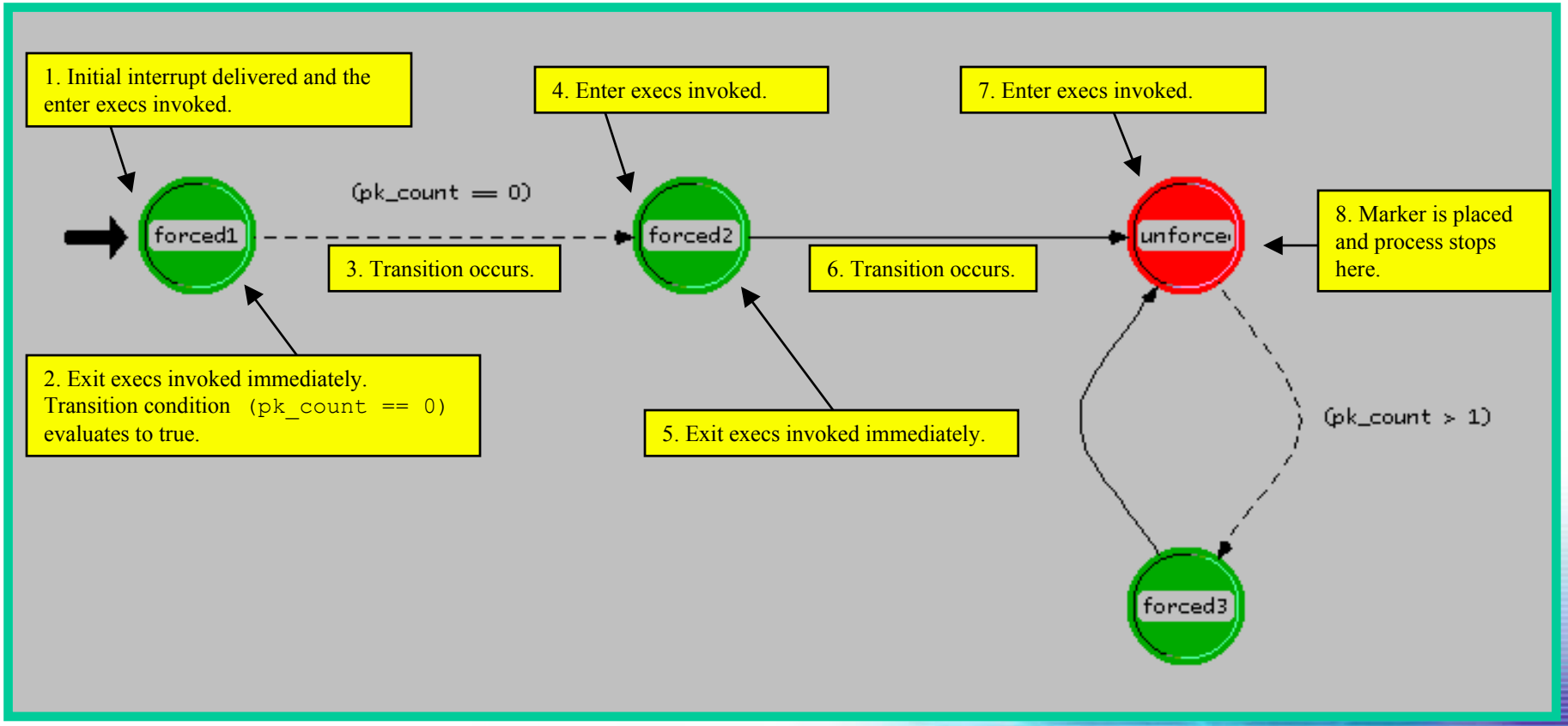
Simulator Internals: **How a Process Handles an Interrupt**

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



Simulator Internals: **Process Model Example**

- **Model with three forced states and one unforced state**



Modeling : **Creating Network Models**

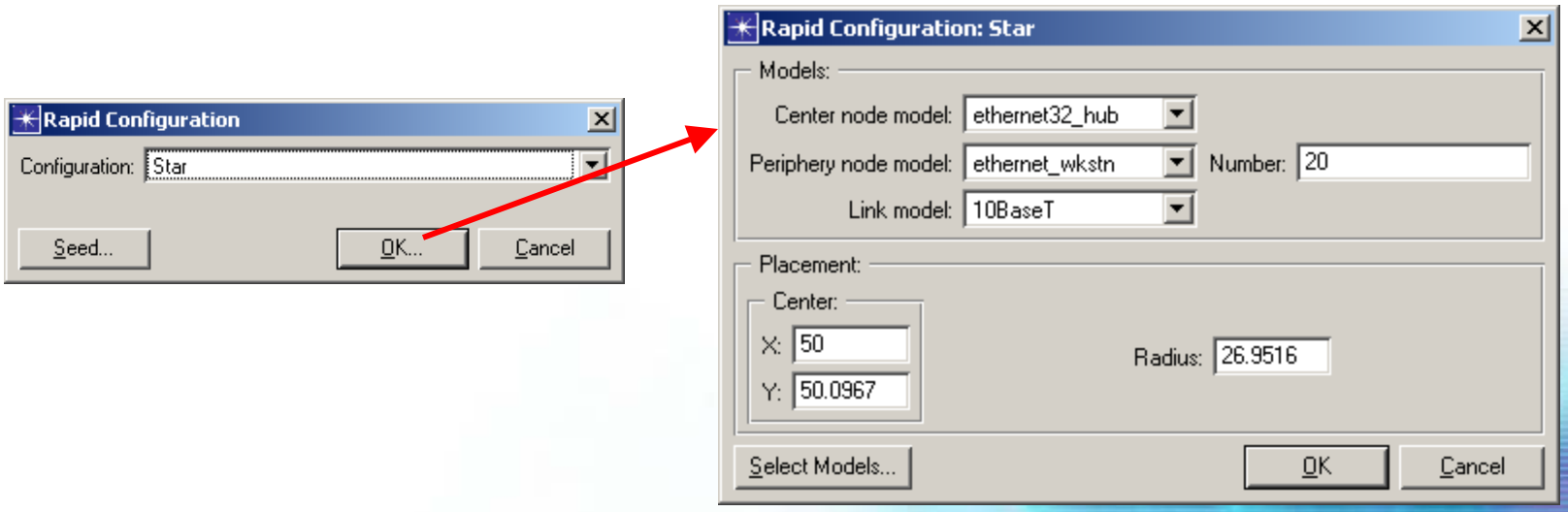
- **There are two ways to create new network models**
  - Manual creation
    - Drag and drop
    - Rapid configuration
  - Import from network management tool

Modeling : **Rapid configuration**

- **Rapid configuration allows you to quickly create networks of any size**
- **Available topology configurations:**

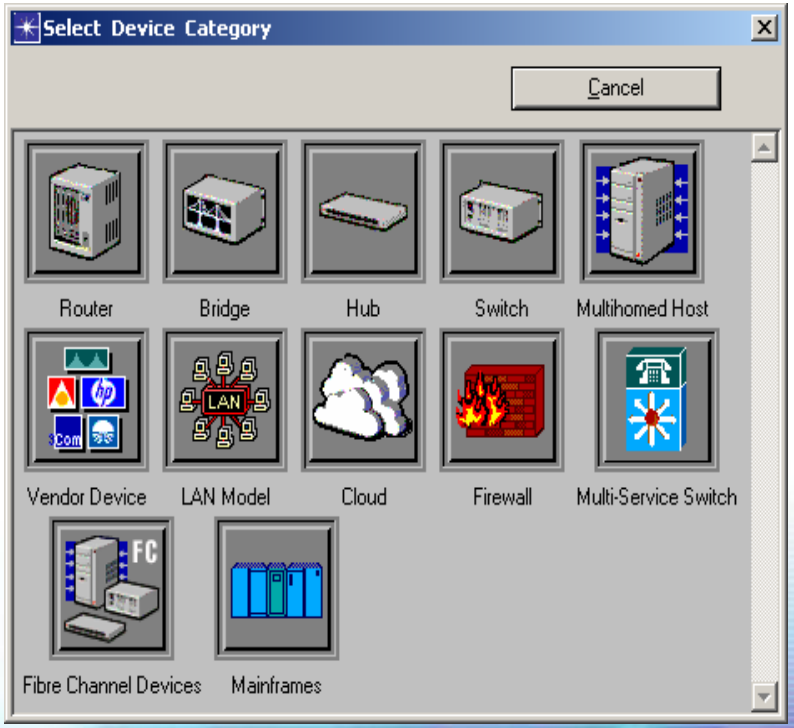
```
Bus; Ring; Star; Tree; Unconnected Net; Mesh (Full or Randomized)
```

- **You control the number of nodes, the node and link models used, how nodes will be arranged, and node locations within the workspace**



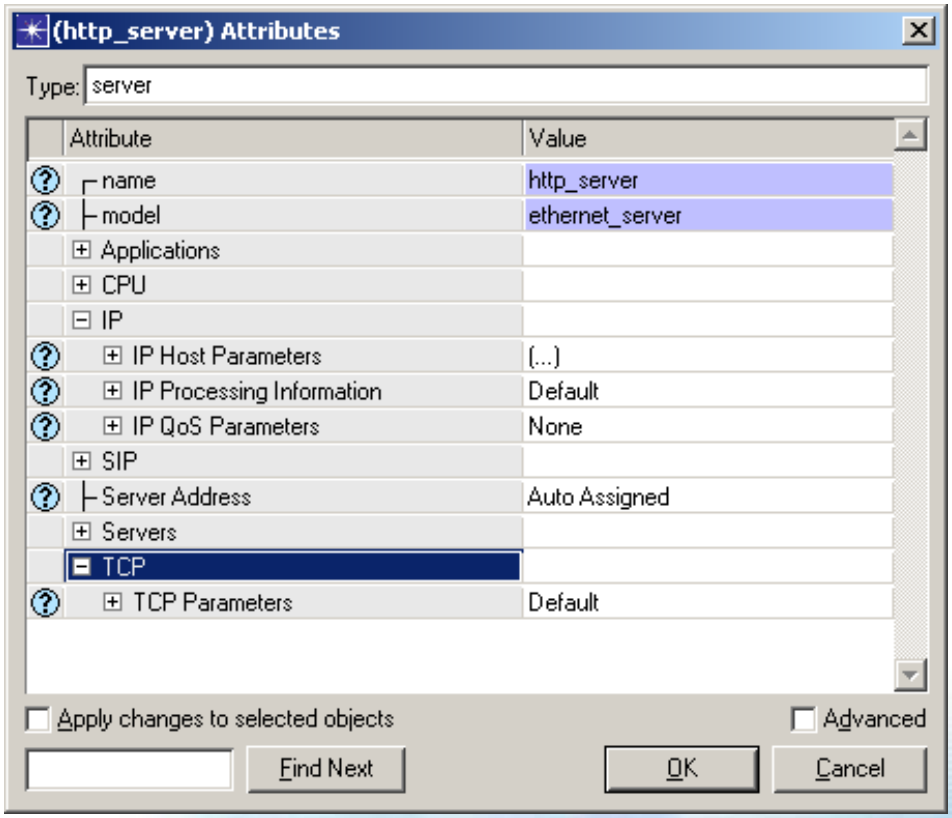
Modeling : **Device Creator**

- **Automatically create a particular device with a specific configuration**
- **Launched from** Topology **menu**



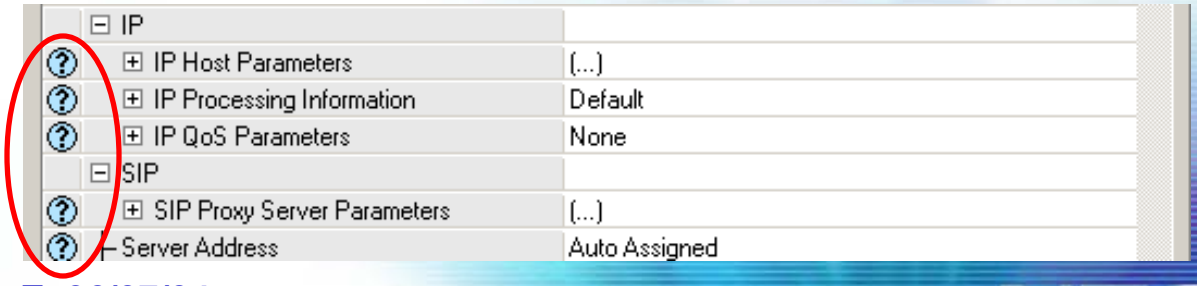
Modeling : **Object Attributes**

- **All objects have attributes that control aspects of their behavior**
- **Attributes may vary from one model to the next one**
- **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**
- **Can be promoted**
  - Allows you to set the value at a higher layer
  - Allows you to specify a range of values at runtime



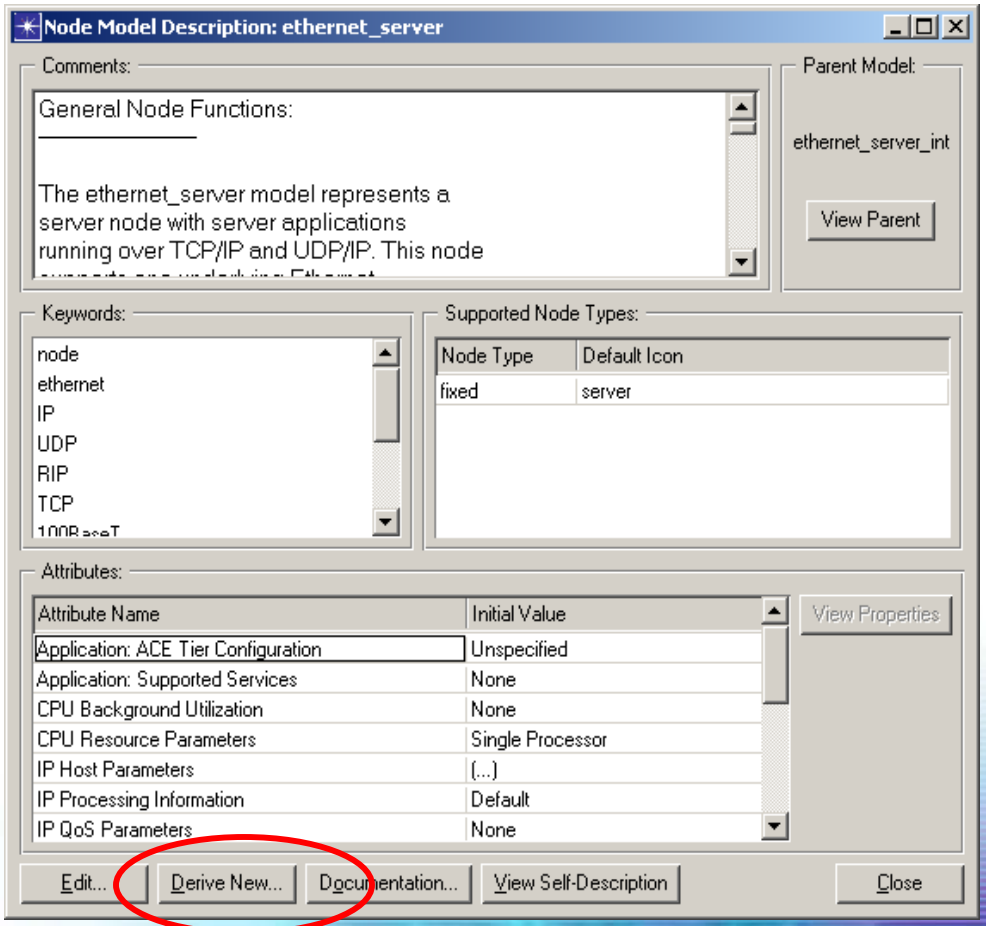
Modeling : **Where to Get Help**

- **Online Documentation from the** Help **menu**
- **Model help accessible by right-clicking object icons in the object palette or by right-clicking objects in the Project workspace and selecting** View Node Description
- **Tool Tips by holding the mouse cursor over any object to get a brief description of that object**
- **Attribute help accessible by clicking on the question mark next to the attribute**



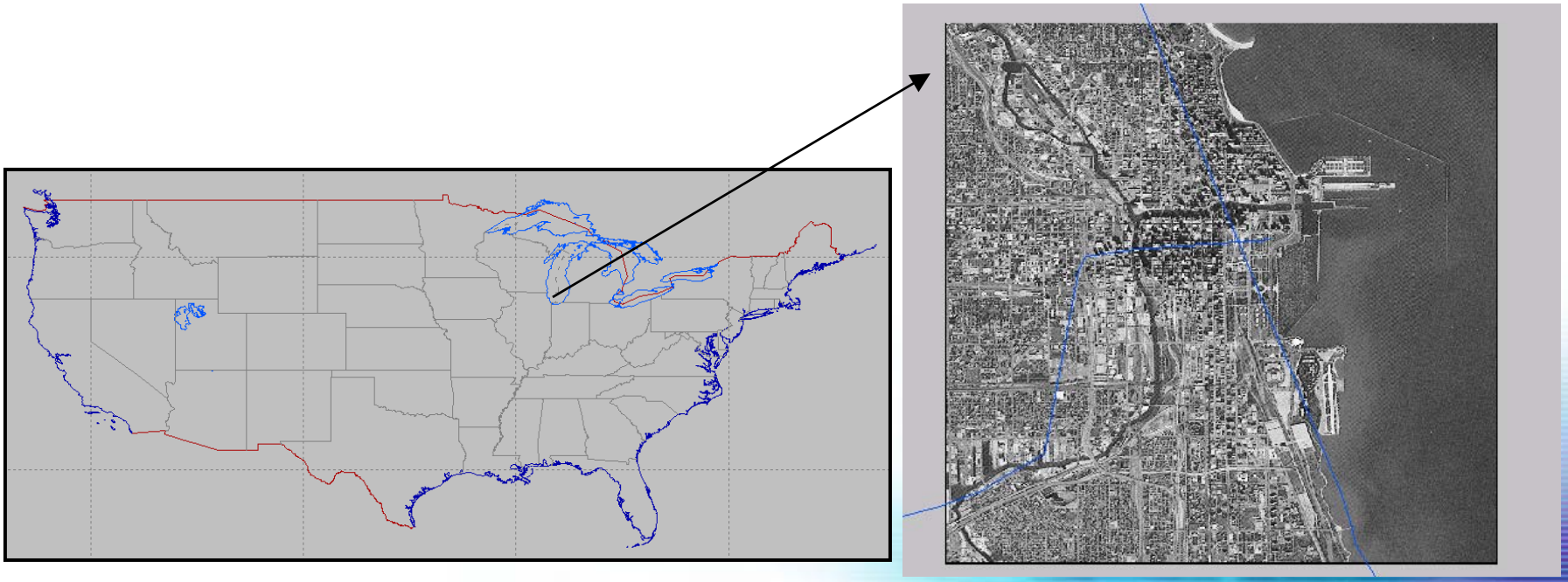
Modeling : **Deriving New Models**

- **Derive a new model based on any existing model**
- **Can alter the attributes of the newly-derived model**



Modeling : **Map Backgrounds**

- **Maps can be loaded as backgrounds for network models**
- **Map backgrounds provide a physical context for model specification**
- **Distance between nodes can be a factor affecting simulation results**

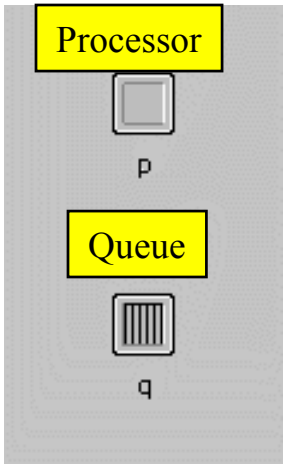


Modeling : **Node Objects**

- **Modules are the basic building blocks of node models**
  - Processors
  - Queues
  - Transceivers
    - Transmitters
    - Receivers
  - Antennas

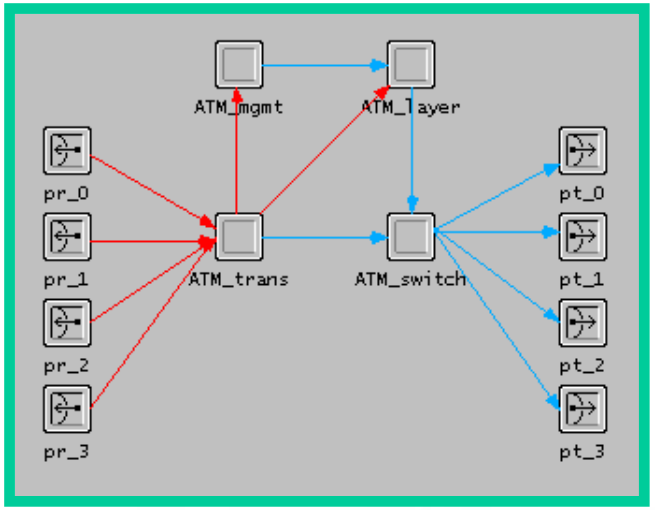
Modeling : **Processors and Queues**

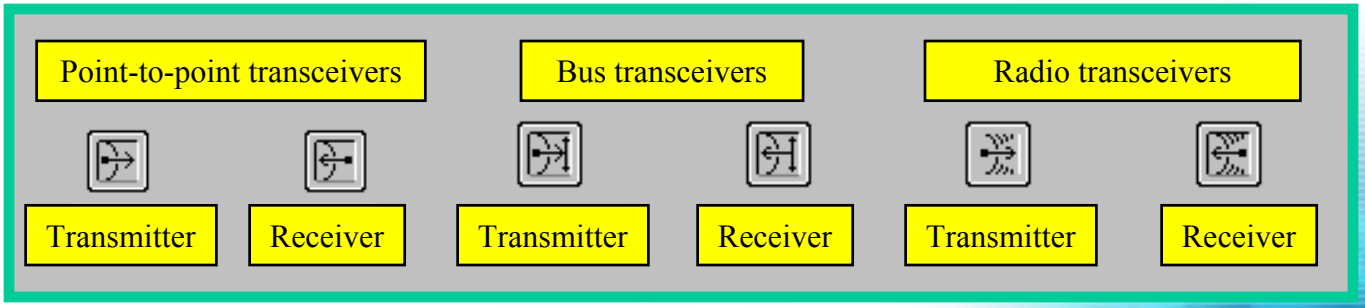
- **Processors**
  - General-purpose building blocks of node models
  - Fully programmable
- **Queues**
  - Offer all the functionality of processors
  - Can also buffer and manage a collection of data packets



Modeling : **Transceivers**

- **Transceivers are the interfaces between objects inside a node and communication links outside of it**
  - Transmitters are the outbound interfaces
  - Receivers are the inbound interface
- **Some transceiver attributes:**
  - Number of channels
  - Data rate
  - Supported packet formats

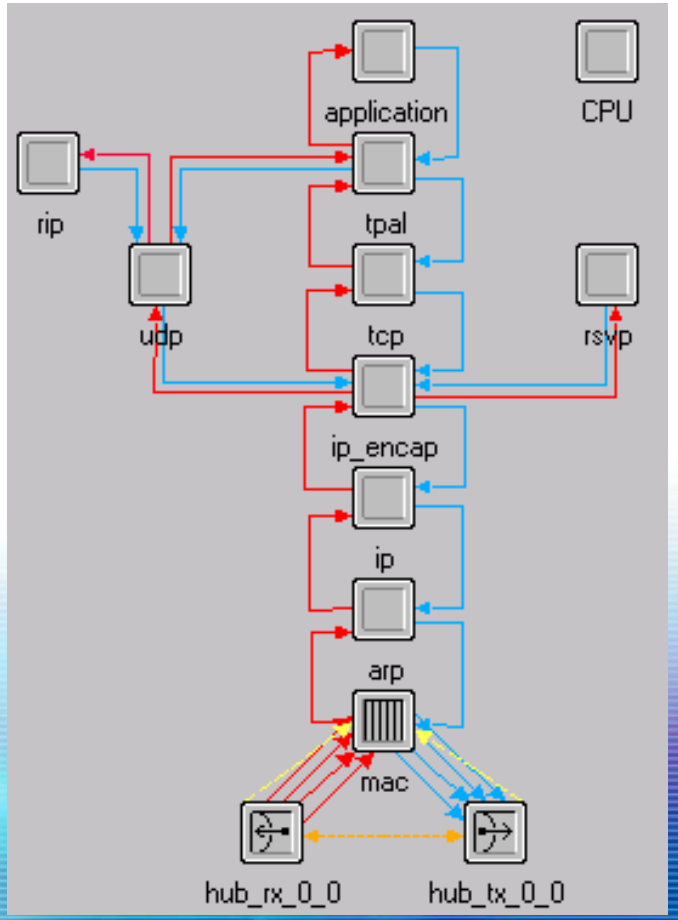




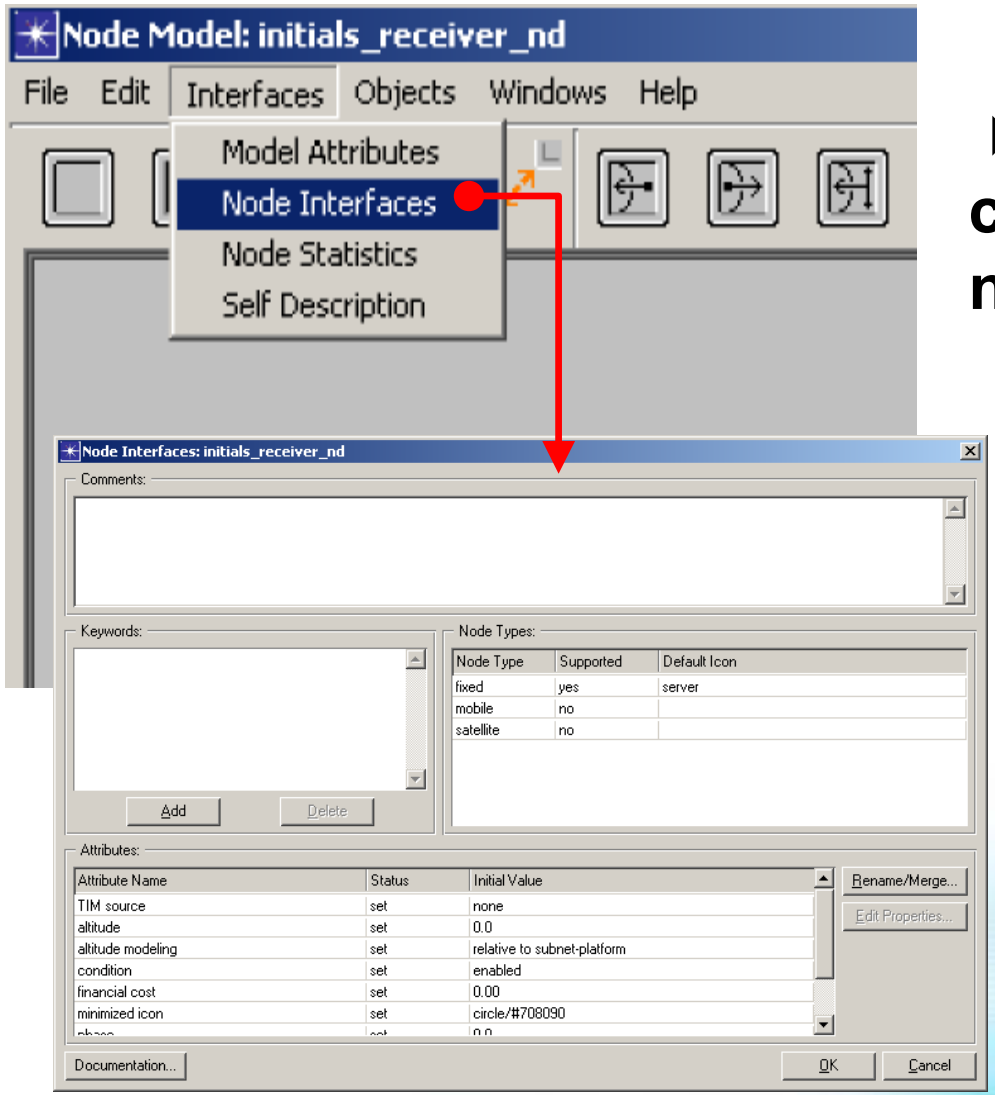
Modeling : **Example of a Node Model**

- **Node models can support:**
  - Layering of protocol functions
  - Dynamic inter-module monitoring
  - Arbitrary node architectures

Node Model : ethernet_wkstn_adv



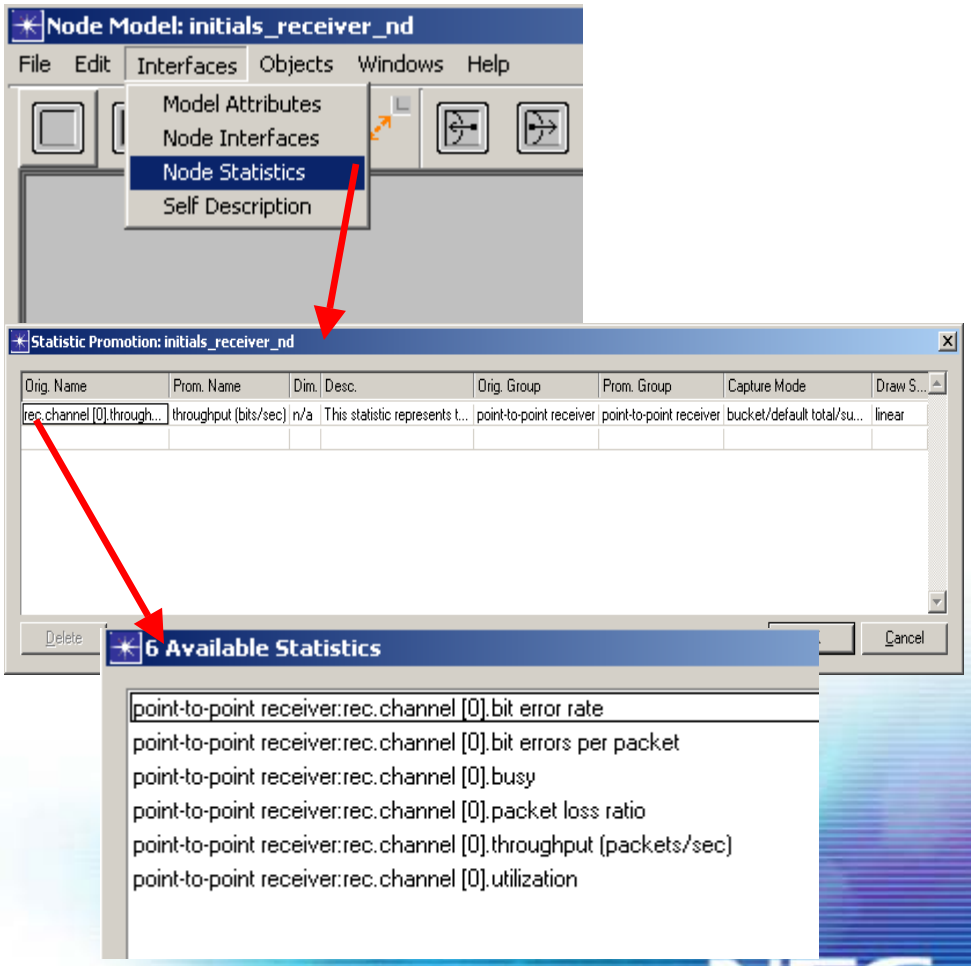
Modeling : **Specifying Node Interfaces**



- **Specify various characteristics of the node**
  - Rename attributes
  - Set attribute values
  - Hide attributes
  - Specify node type
  - Add comments

Modeling : **Specifying Available Node Statistics**

- Specify statistics available from project editor
- Selecting a statistic from the “Available Statistics” table adds the statistic to the “Statistic Promotion” table



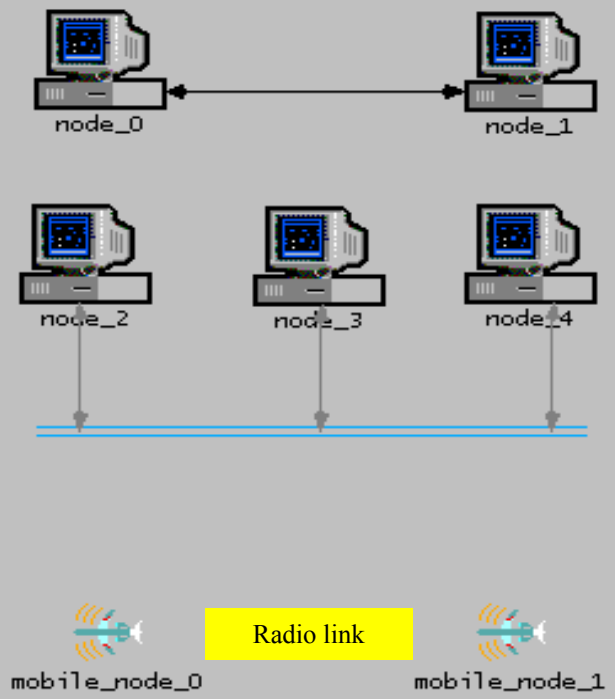
Modeling : **Link Types**

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

A point-to-point link transfers data between two fixed nodes.

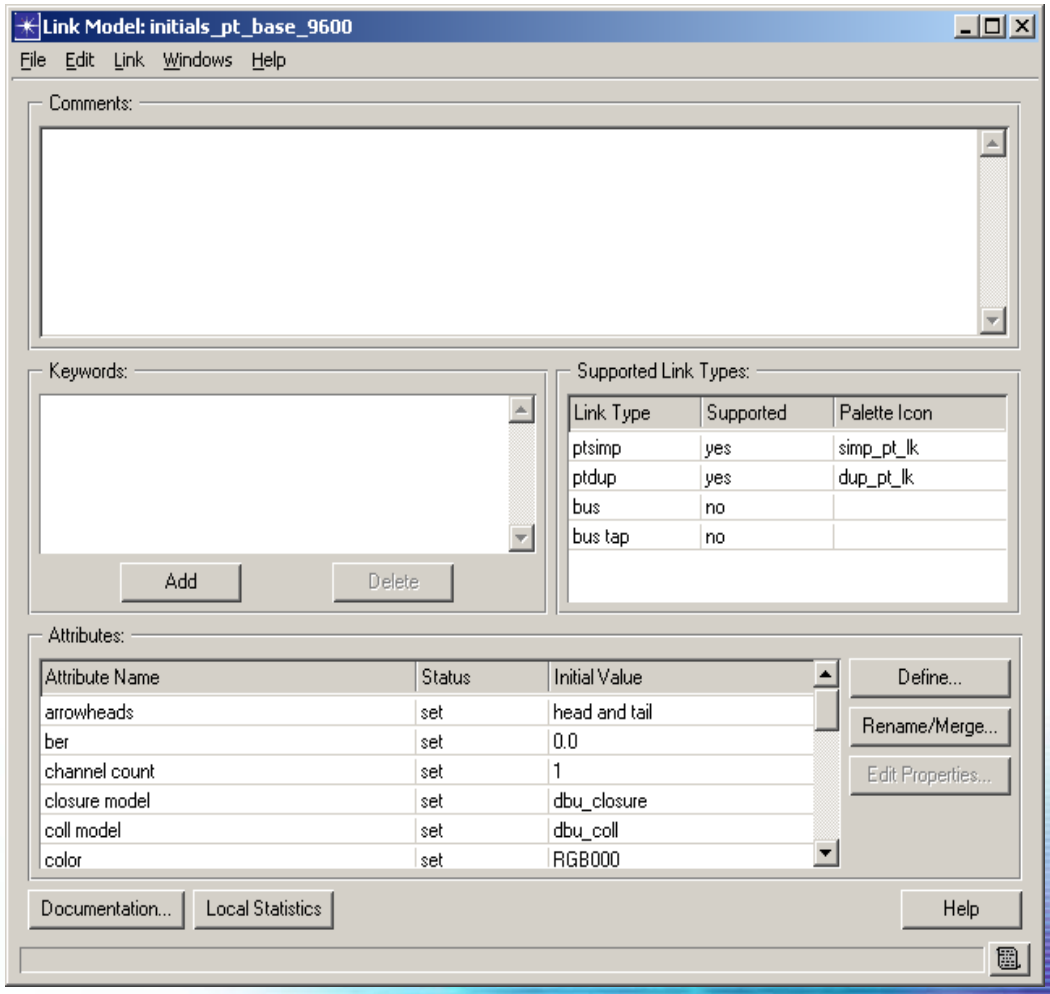
A bus link transfers data among many nodes and is a shared media.

A radio link, established during a simulation, can be created between any radio transmitter-receiver channel pair. Satellite and mobile nodes must use radio links. Fixed nodes may use radio links. A radio link is not drawn but is established if nodes contain radio transceivers.



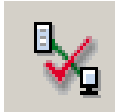
Modeling : **Link Editor**

- Create or modify links
- Choose link types
- Modify attributes



Modeling : **Verify Links**

- **Verify links before running a simulation**

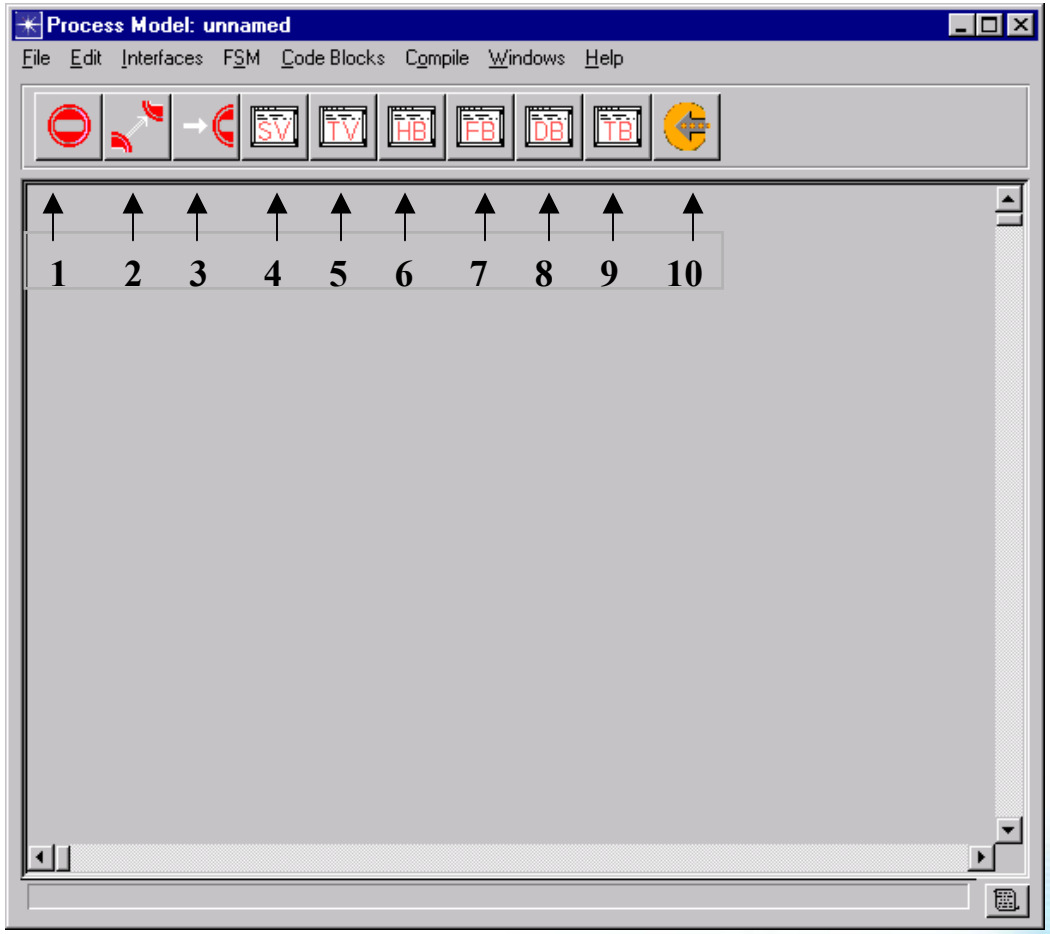


- **Ensures that point-to-point and bus link connections are valid**
  - Enough transmitters and receivers to support all of the incoming and outgoing links
  - Data rates of the connected transmitter and receiver match the data rate of the link
  - Transceivers support the attached link technology

Modeling: **Process Models**

- **Process models represent algorithms**
  - Communications protocols and algorithms
  - Shared-resource managers
  - Queuing disciplines
  - Specialized traffic generators
  - Statistic-collection mechanisms
  - Control processes
- **Process editor provides the features for creating process models**

Modeling : **Process Editor**

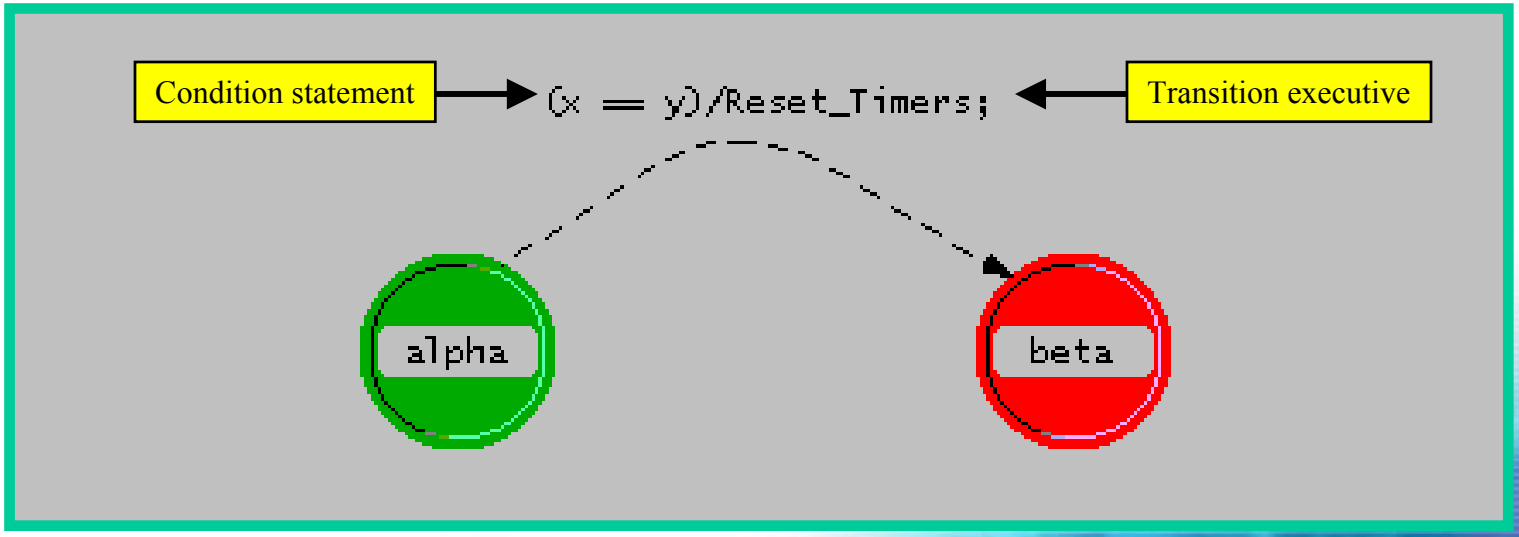


- **Toolbar buttons:**

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

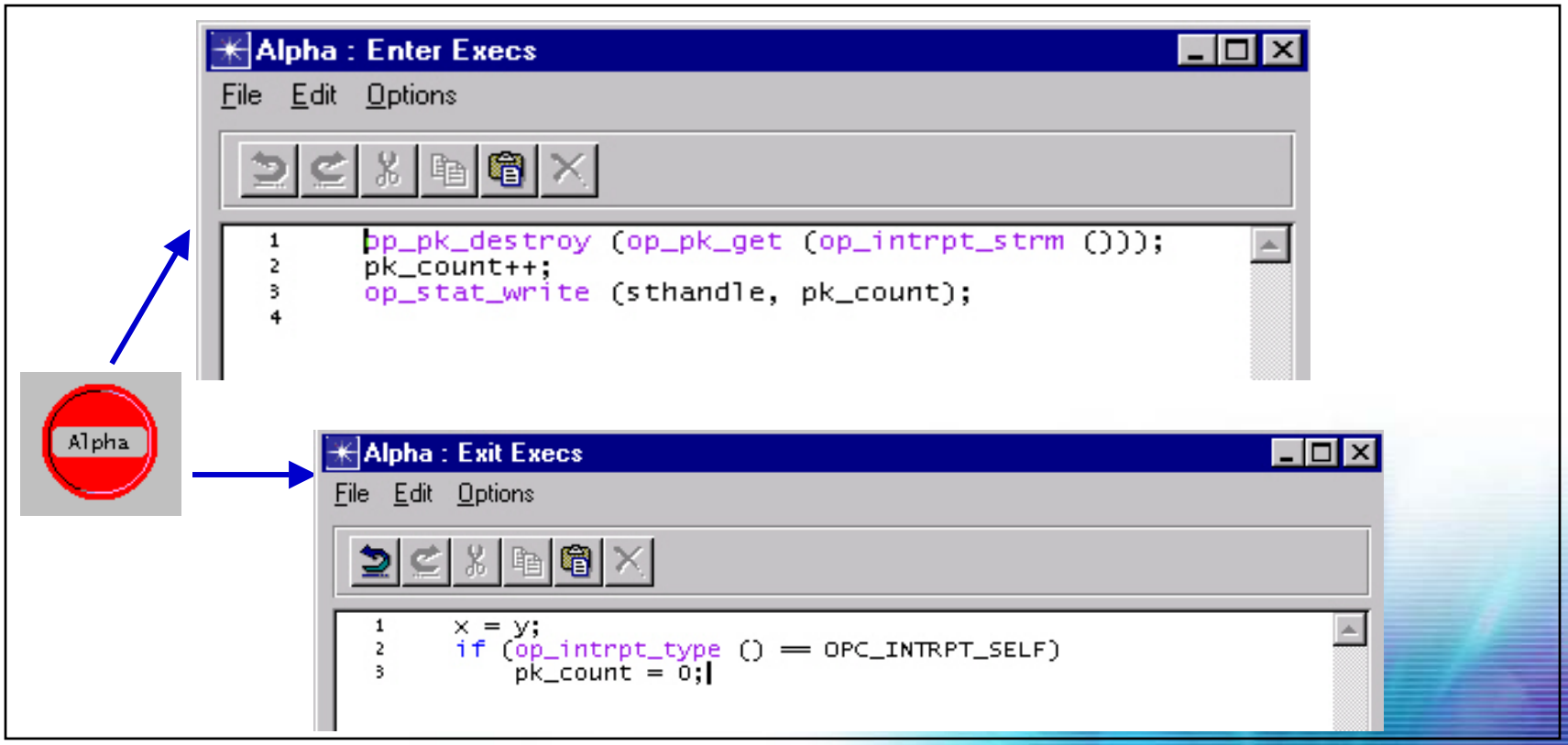
Modeling: **State Transitions**

- **Transitions connect states**
  - Conditional
  - Unconditional
  - Transition executive
- **If the condition statement *(x == y)* is true, the transition executive *(Reset_Timers)* is invoked**



Modeling: **State Executive Blocks**

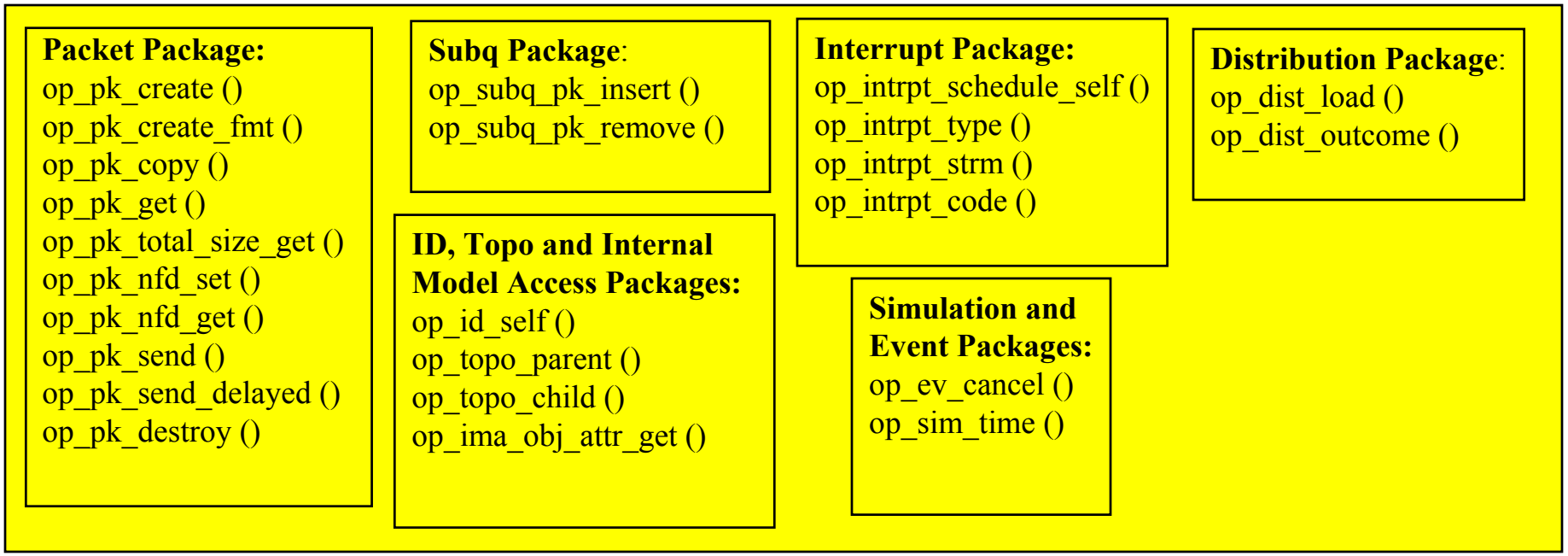
- **Each state has two executive blocks**
  - *Enter executives* are invoked upon entering a state
  - *Exit executives* are invoked before exiting a state



Modeling: **Kernel Procedures (KP)**

- **Pre-written functions for difficult, tedious, or common operations**
- **KPs free users from addressing memory management, data structure, handling event processing, etc.**
- **KPs focus on communication modeling**
- **All KPs begin with prefix** op_*

Modeling: **Common Kernel Procedures**

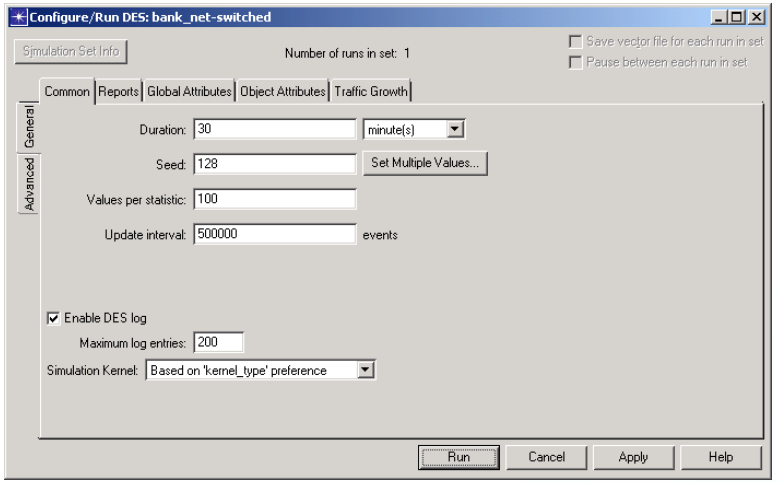


Naming convention for kernel procedures -

op_<family name describing object >_<action>

Performing a Simulation: **Configuring Simulations**

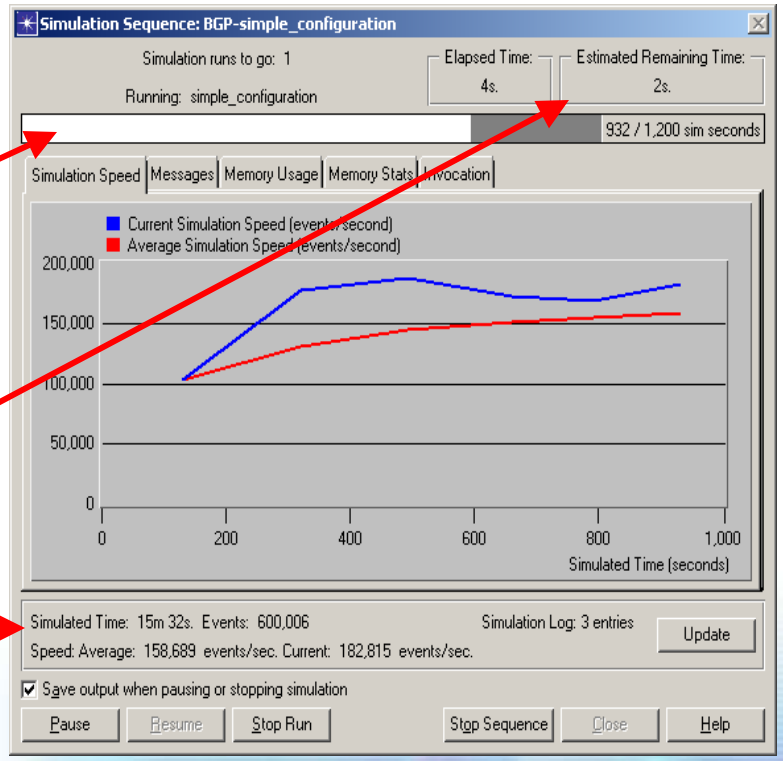
- **Scenarios automatically provide a default duration and random number seed for simulations**
- **Users can set simulation attributes by choosing** Configure/Run Discrete Event Simulation **from the** DES **menu, or by clicking on the "running man" icon:**



Configure Simulation Dialog Box	
Name	Description
Duration	Duration of simulation, in simulated time
Values Per Statistic	Number of values to be collected for each statistic
Seed	Random number generation seed

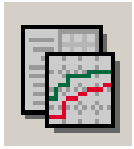
Performing a Simulation: **Running a Simulation**

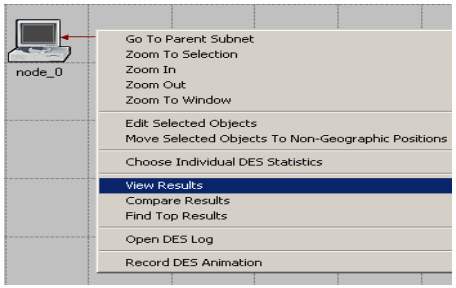
- **The Simulation Sequence window shows the progress of simulation**
- **Elapsed time bar displays the progress of the simulation**
  - Appears after 1,000,000 events by default
- Elapsed/Remaining Time**: Real time elapsed and remaining time**
- Simulation Time**: Simulation time elapsed and number of events processed**



Performing a Simulation: **Viewing Results**

- **Results can be displayed by:**
  - Selecting the **View Results** button on the tool bar
  - Selecting **View Statistics** from the **DES** menu
  - Right-clicking the project workspace and selecting from the pop-up menu

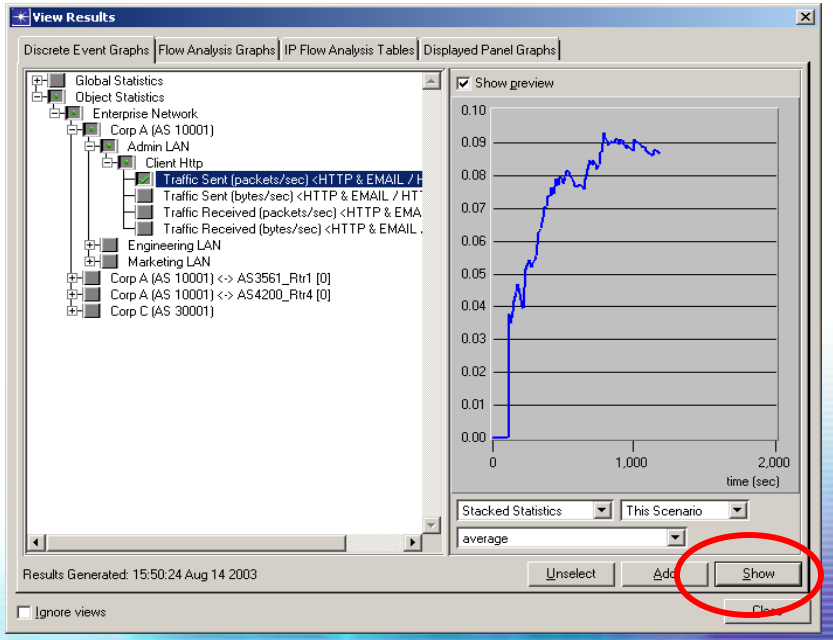




- **View Results dialog box allows the user to select the results to display.**

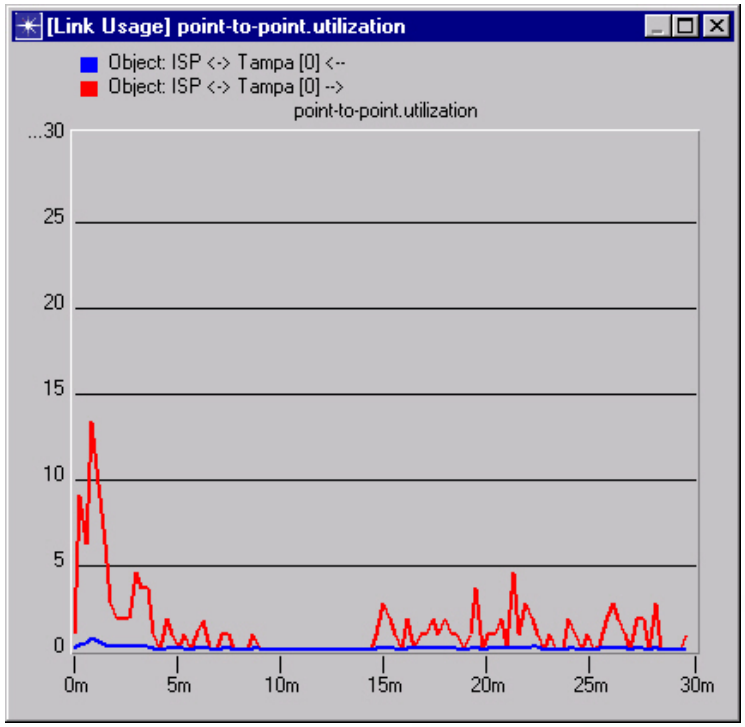
  - ***Note*: Only the statistics you chose for collection will be available**

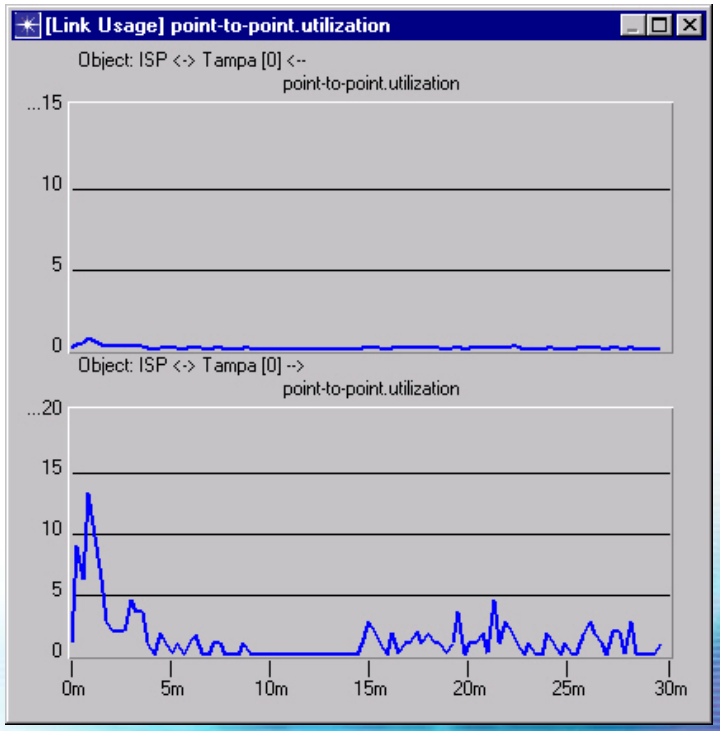
- **The** Show **button in the View Results dialog box displays a graph of the selected statistics**



Performing a Simulation: **Viewing Results (cont.)**

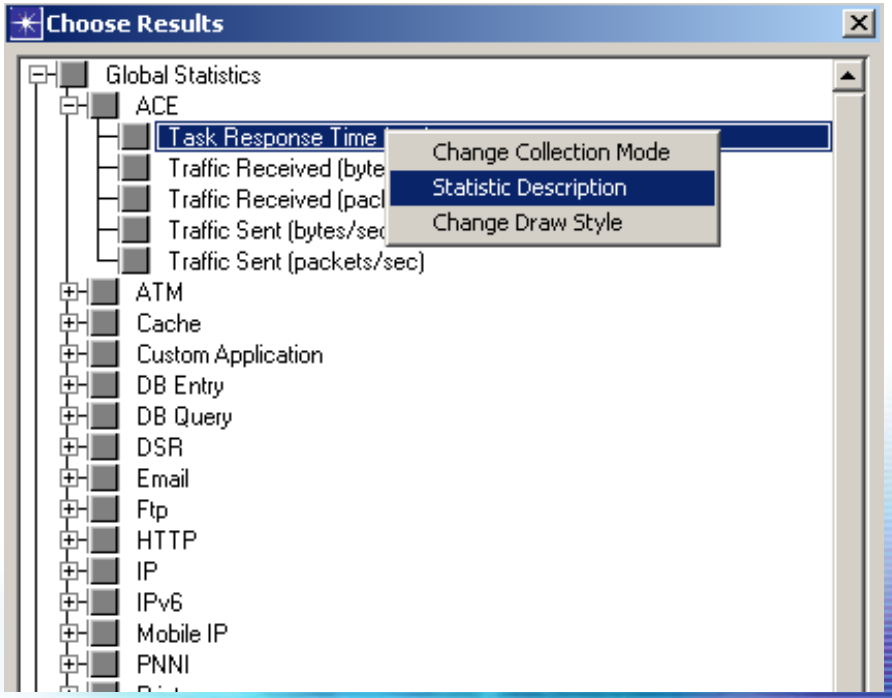
- **Multiple graph panels can be displayed at the same time**
- **Each panel can contain one or more traces in an** Overlaid **or** Stacked **layout**





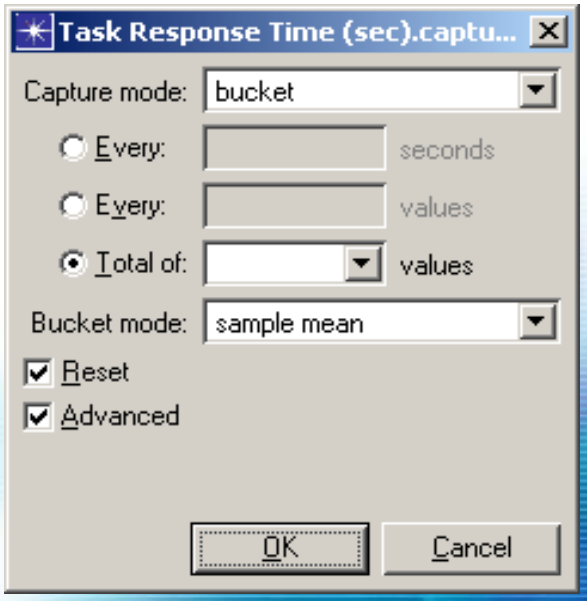
Performing a Simulation: **Statistic Attributes**

- **Right-clicking on a statistic while in the Choose Results dialog box presents a menu of statistic attributes**
- **Statistic attributes include:**
  - Change Collection Modes
  - Statistic Description
  - Change Draw Style



Performing a Simulation: **Statistic Collection Modes**

- **Normal 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**
- **Bucket mode: All the data points in a bucket are collected and processed according to a user-defined parameter**
  - Max
  - Min
  - Sum
  - Count
  - Sample mean
  - Time average



References

The material to prepare this presentation has been extracted from:

[1] Overview of OPNET's Product Portfolio, OPNETWORK2003, Session 1000.

[2] Overview of Modeler and ODK, OPNETWORK2003, Session 1006.

[3] Introduction to using Modeler, OPNETWORK2003, Session 1201.

[4] Habits of highly-effective OPNET users, OPNETWORK2003, Session 1213.

[5] www.opnet.com