SYSC 4005/5001

<u>SIMULATION AND MODELING</u>

Introduction to Using OPNET Modeler

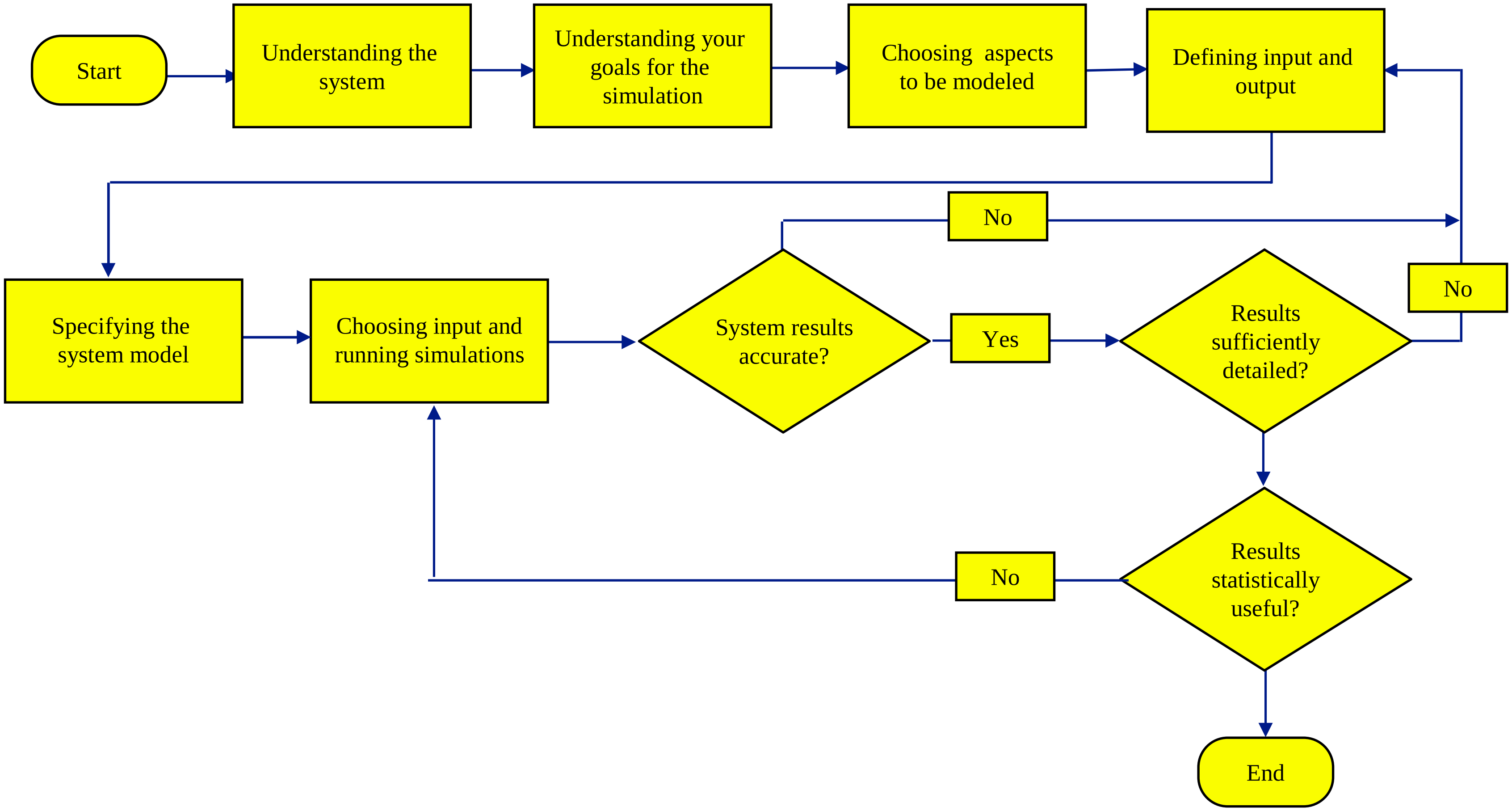
Reference: OPNETWORK2002

Outline

- Introduction
- Simulation Overview
- Events and Event List Concepts
- Network Modeling
- Node Modeling
- Link Modeling
- Process Modeling
- Running Simulations
- Collecting and Analyzing Results
- Conclusion

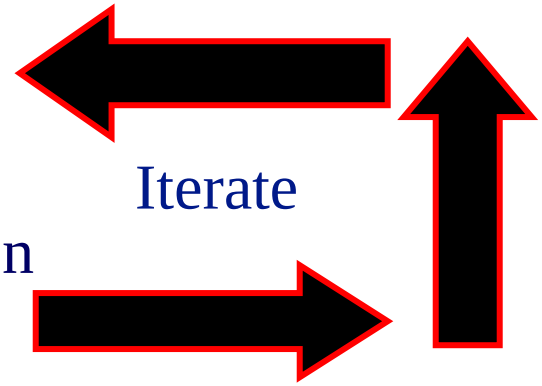
Note: You don’t have to use Opnet Modeler for your project!

Simulation Methodology


Start
Understanding the system
Understanding your goals for the simulation
Choosing aspects to be modeled
Defining input and output
Specifying the system model
Choosing input and running simulations
System results accurate?
No
Yes
Results sufficiently detailed?
No
Results statistically useful?
No
End

The Project/Scenario Workflow

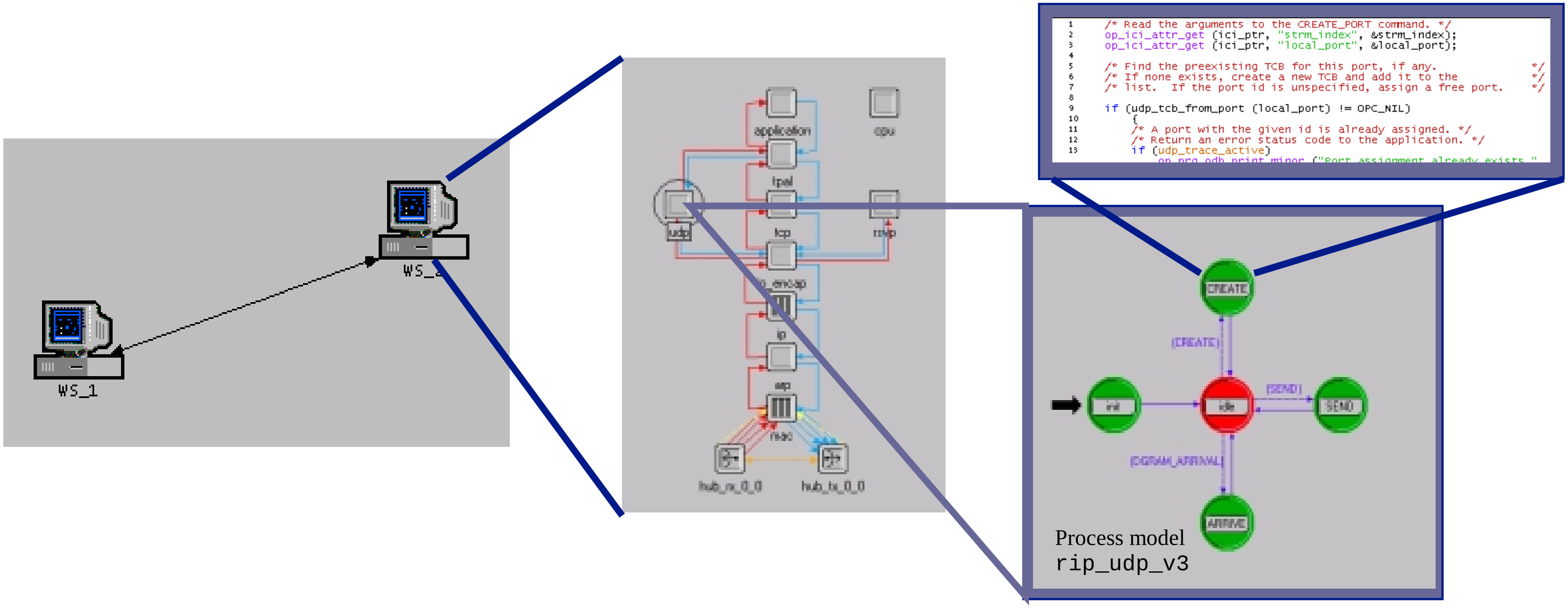
- Create project
- Create baseline scenario
  - Import or create topology
  - Import or create traffic
  - Choose results and reports to be collected
  - Run simulation
  - View results
- Duplicate scenario
  - Make changes
  - Re-run simulation
  - Compare results



Iterate

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



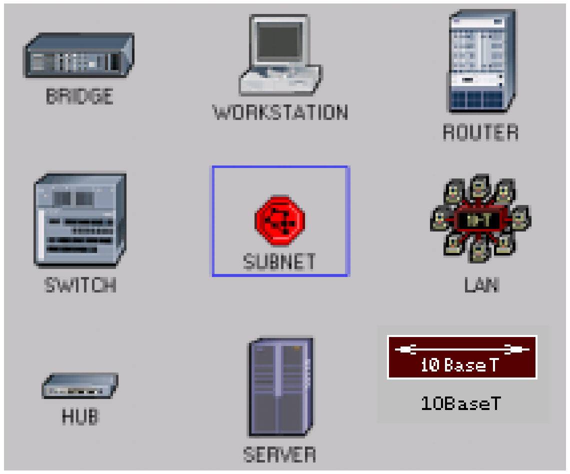
```
/* Read the arguments to the CREATE_PORT command. */
op_ici_attr_get (ici_ptr, "strm_index", &strm_index);
op_ici_attr_get (ici_ptr, "local_port", &local_port);

/* Find the preexisting TCB for this port, if any.            */
/* If none exists, create a new TCB and add it to the          */
/* list.  If the port id is unspecified, assign a free port.   */

if (udp_tcb_from_port (local_port) != OPC_NIL)
    {
    /* A port with the given id is already assigned. */
    /* Return an error status code to the application. */
    if (udp_trace_active)
        op_prg_odb_print_minor ("Port assignment already exists."
```

Process model
`rip_udp_v3`

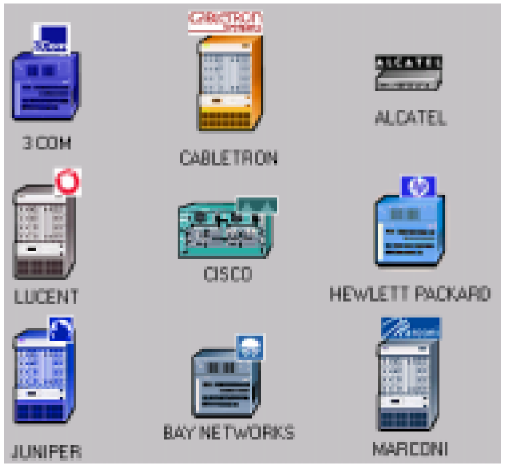
Network Domain: Network Objects

- 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

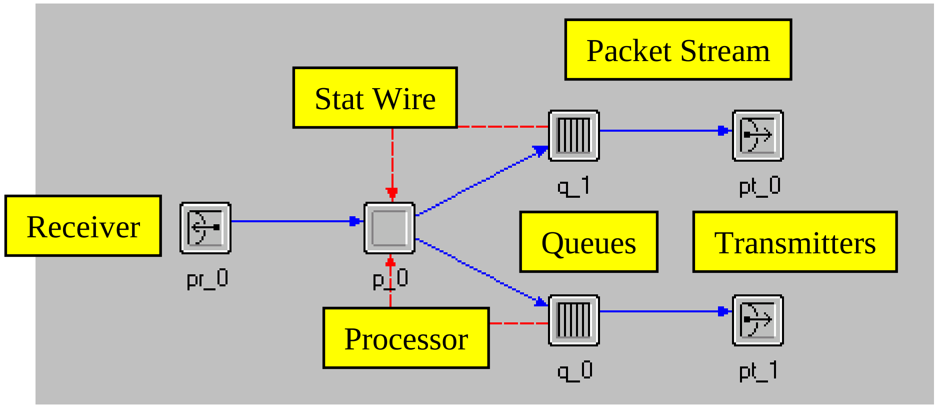


Vendor Devices



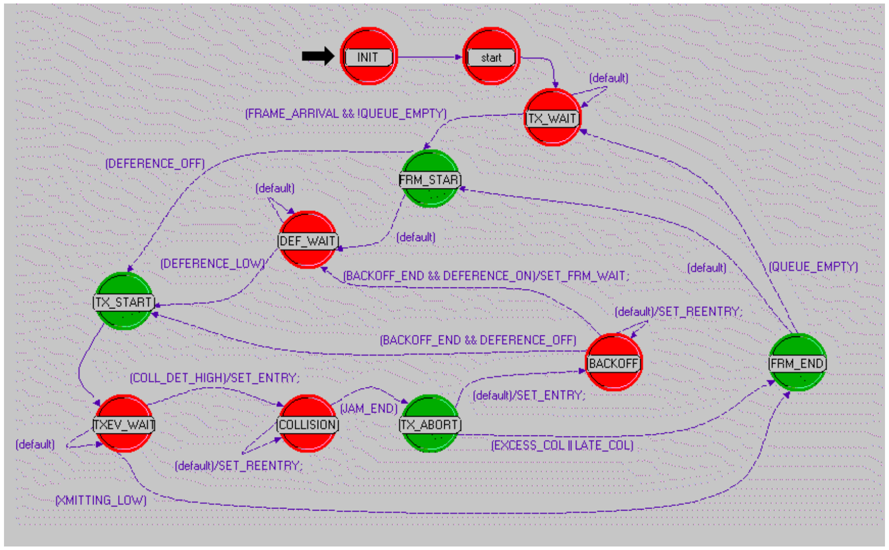
Node Domain

- 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



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

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
  - For example: throughput, bits received, bits forwarded, etc.

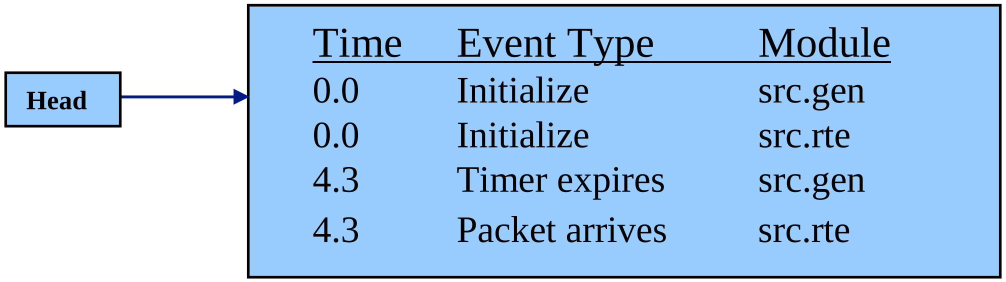
Events and Event List Concepts

Event-Driven Simulation

- Events are specific activities that occur at a certain time
- OPNET simulations are event-driven
- Simulation time advances when an event occurs
- A different method might be to sample at regular intervals
  - Disadvantages:
    - » Accuracy of results is limited by the sampling resolution
    - » Simulation is inefficient if nothing happens for long periods

Event List Concepts

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



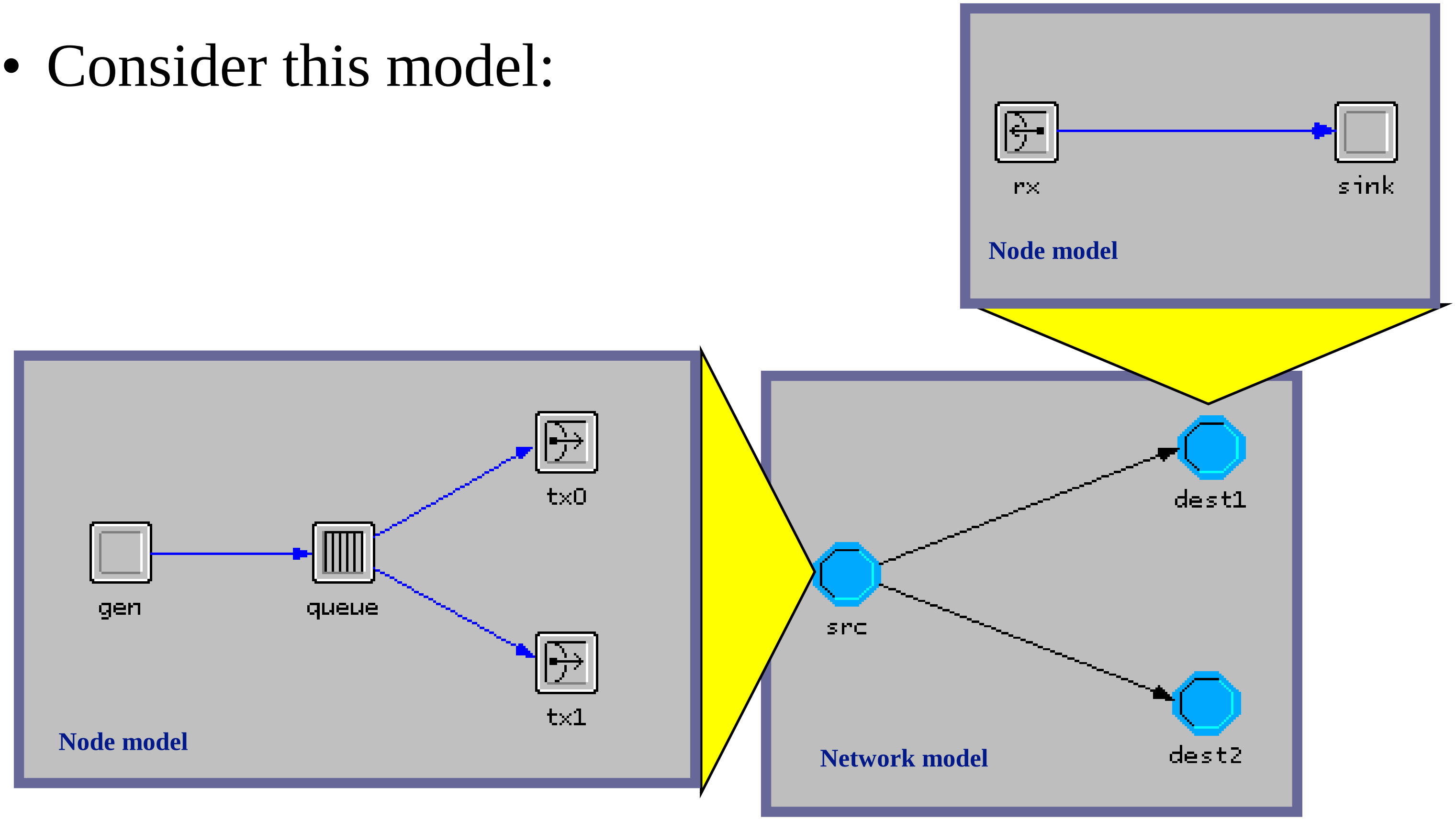
Head

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

Event List Example

- Consider this model:

Node model: dest1



Node model: src

Network model

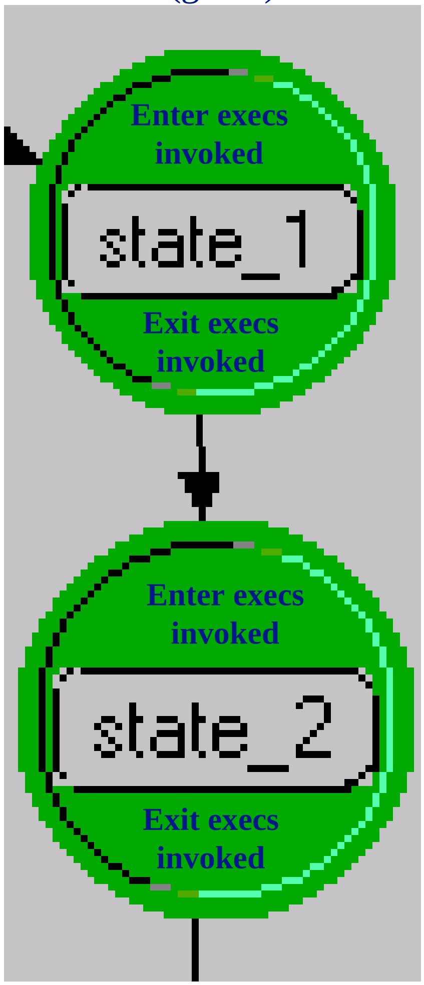
Event List Example (cont.)

- 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

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

Forced (green) states



No blocking or waiting

Transition to next state

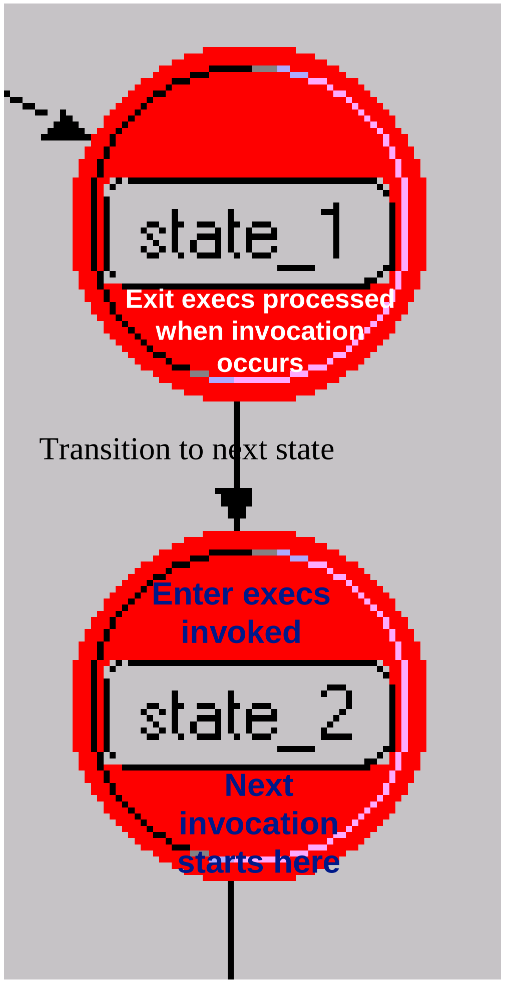
No blocking or waiting

Transition to next state

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

Unforced (red) states



Exit execs processed when invocation occurs

Transition to next state

Enter execs invoked

Next invocation starts here

Blocking, waiting for invocation

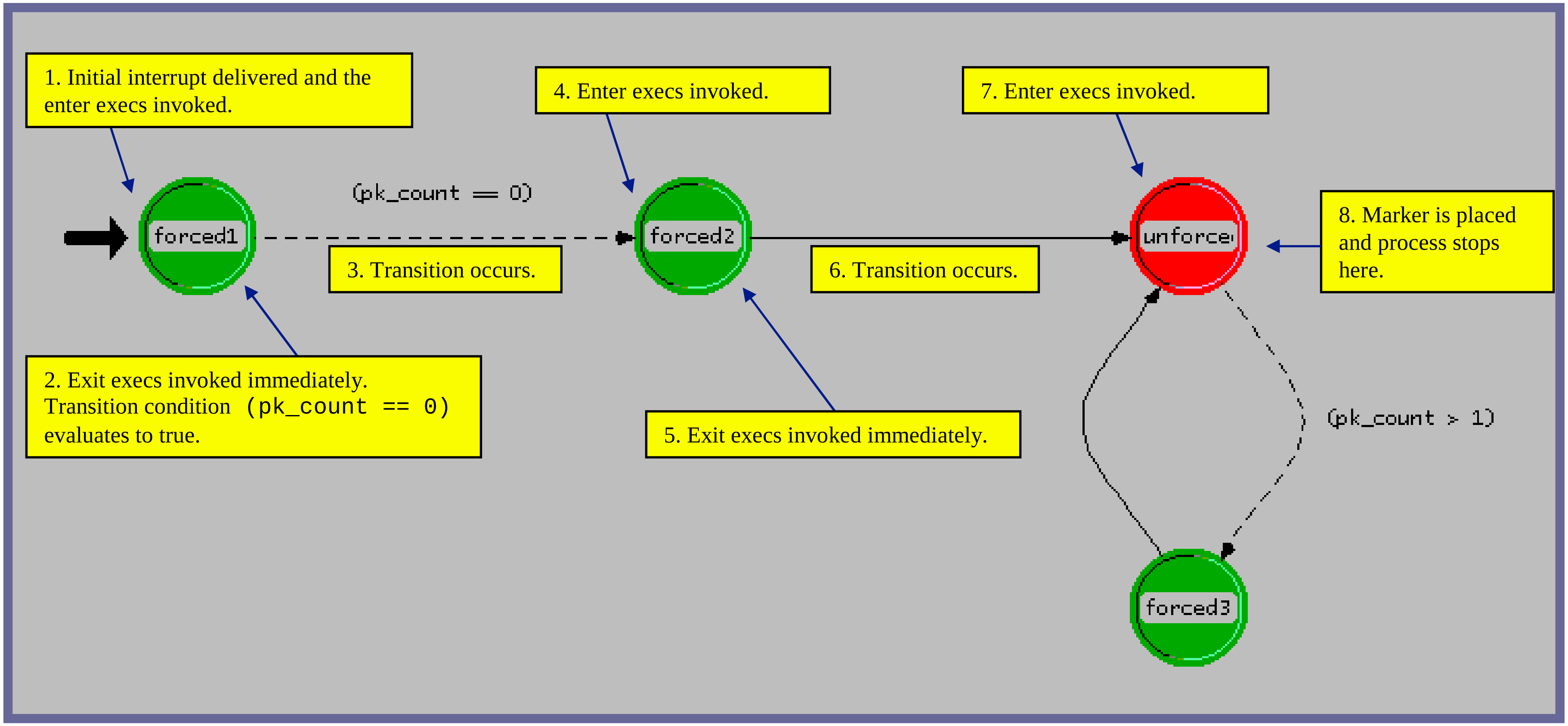
Start of invocation

End of invocation

Blocking, wait for next invocation

Process Model Example

- Model with three forced states and one unforced state:



1. Initial interrupt delivered and the enter execs invoked.

2. Exit execs invoked immediately. Transition condition `(pk_count == 0)` evaluates to true.

3. Transition occurs.

4. Enter execs invoked.

5. Exit execs invoked immediately.

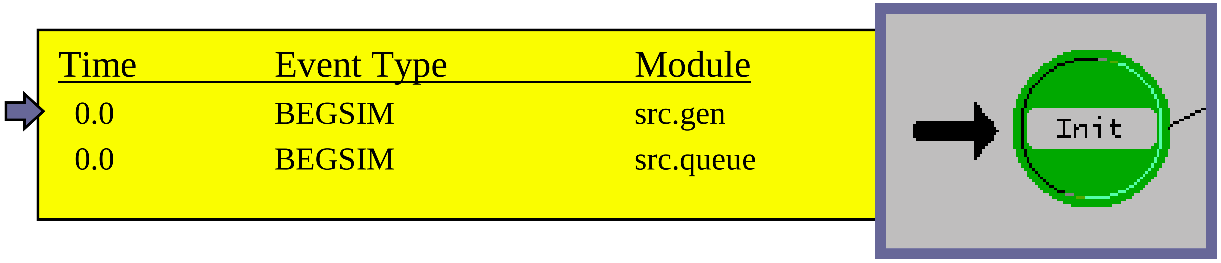
6. Transition occurs.

7. Enter execs invoked.

8. Marker is placed and process stops here.

Starting the Simulation

- Simulation Kernel reads the event at the head of the event list, and delivers control to the process in the `src.gen` module
- Process begins execution at the initial state, marked with the black arrow
- Process executes the Init state’s enter execs



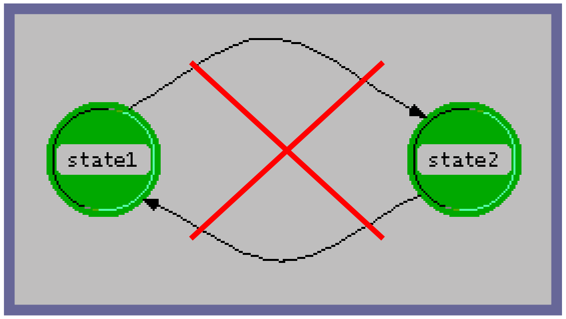
Time	Event Type	Module
0.0	BEGSIM	src.gen
0.0	BEGSIM	src.queue

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?

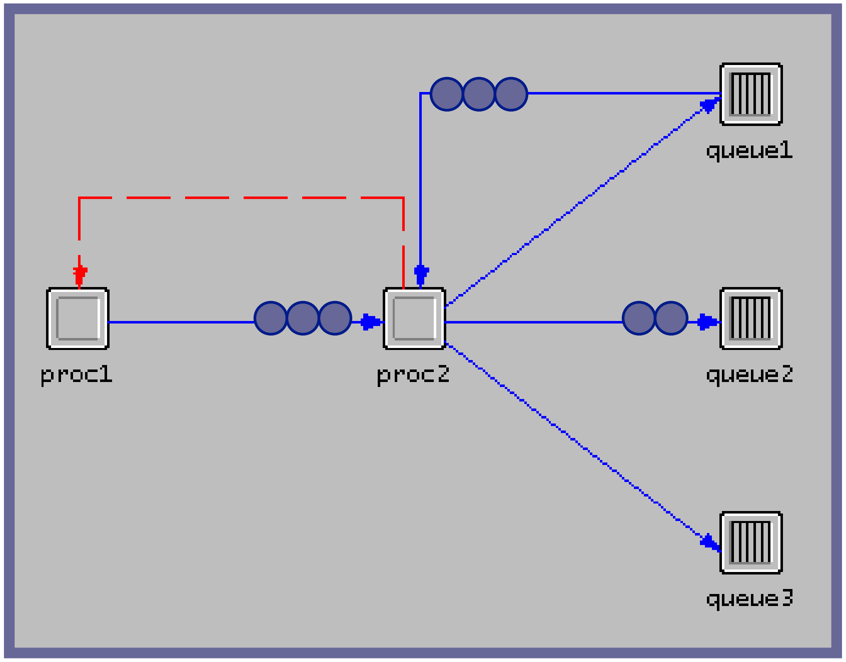
- Simulation time advances only when an event with a later time is processed from the event list
- No simulation time occurs during the execution of a process
- No time elapses during transitions between states
- A process model must always have an unforced (red) state so time can advance
  - Avoid endless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



Packets

Packets

- Information-carrying entities that circulate among system components
- General data structures, organized into fields of user-defined information
- Dynamically created and destroyed as the simulation progresses
- A single system may rely on multiple types of packets with different formats



Packet Formats

- Packets can either be unformatted or formatted
- Unformatted packets have no user-defined data fields
- Formatted packets have zero or more fields of type
  - Integer
  - Floating point
  - Structure
  - Packet
  - Information

Events for Packet Transmission

- All packet transmissions are modeled with 4 events
  - Start of Transmission
  - End of Transmission
  - Start of Reception
  - End of Reception
- Simulation Kernel automatically schedules the events

Network Modeling

Creating Network Models

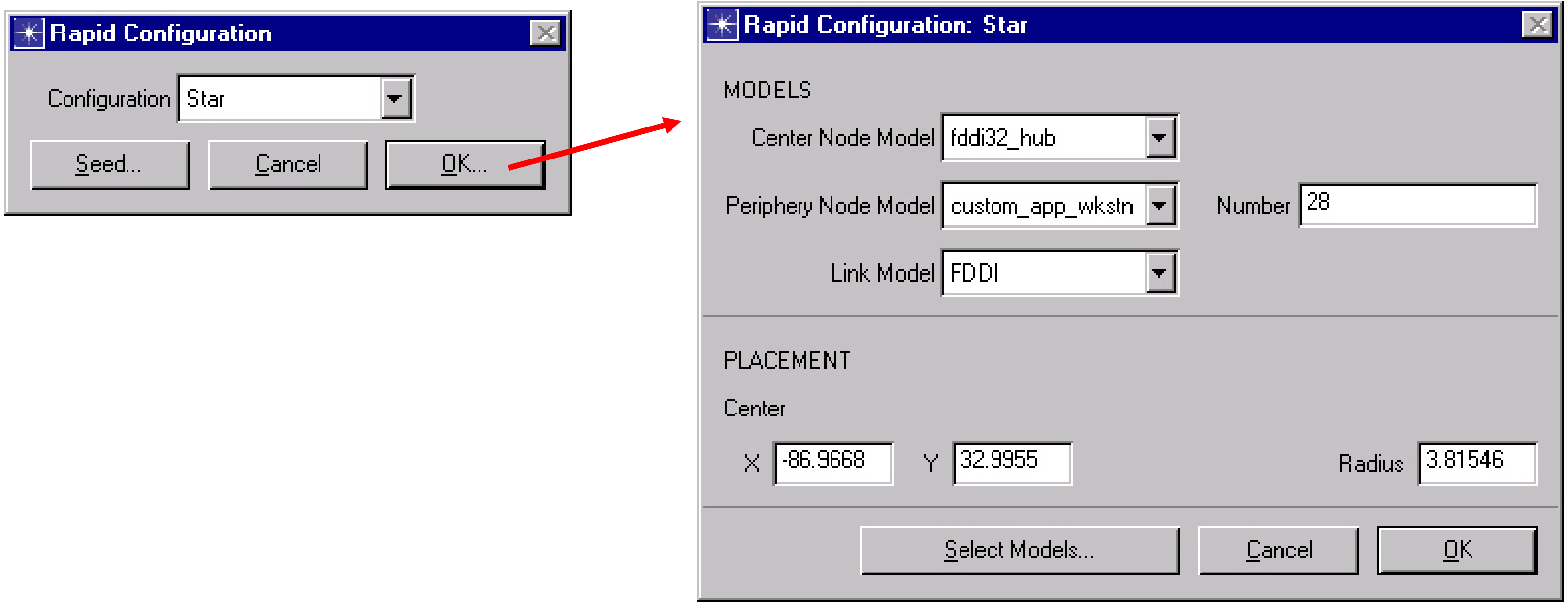
- There are two ways to create new network models
  - Manual Creation
    - Drag and drop
    - Rapid configuration
  - Import from network management tool
    - HP Network Node Manager
    - Tivoli Netview
    - Router configuration files
    - ATM text files
    - XML
    - ACE
    - VNE Server

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



Rapid Configuration

Configuration: Star

Rapid Configuration: Star

MODELS

Center Node Model: fddi32_hub

Periphery Node Model: custom_app_wkstn

Number: 28

Link Model: FDDI

PLACEMENT

Center

X: -86.9668

Y: 32.9955

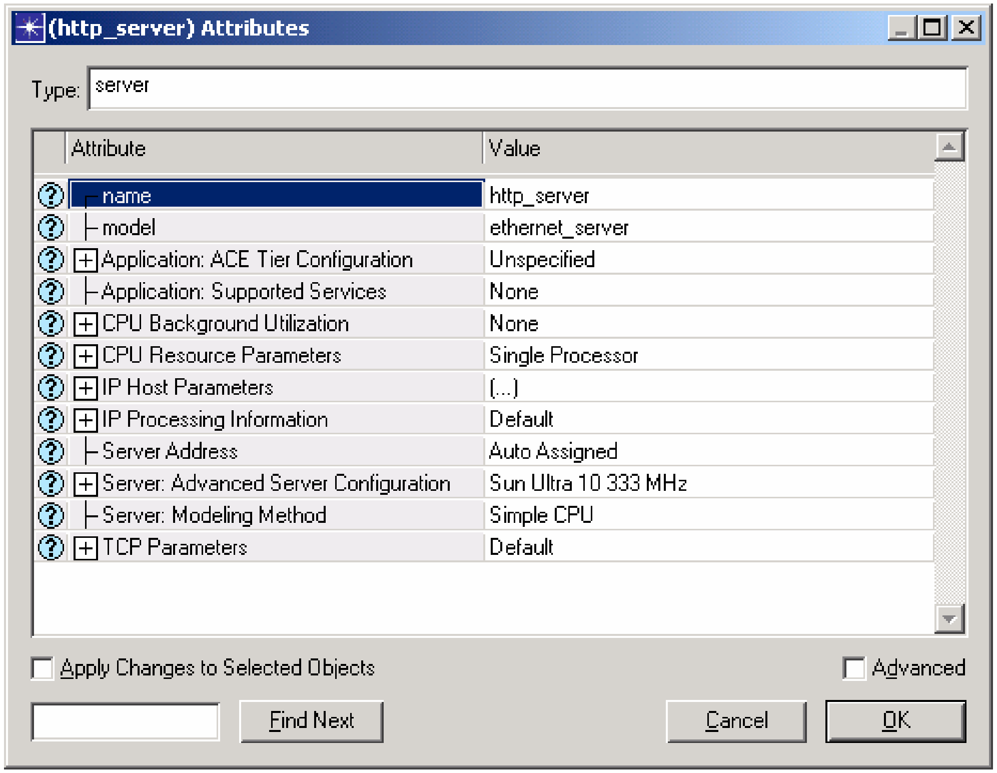
Radius: 3.81546

Startup Wizard

- The **Startup Wizard** can quickly configure a new scenario
- There are several settings for each scenario:
  - Name
  - Initial topology
  - Network scale
  - Network size
  - Technologies

Object Attributes

- 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



(http_server) Attributes

Type: server

Attribute	Value
name	http_server
model	ethernet_server
Application: ACE Tier Configuration	Unspecified
Application: Supported Services	None
CPU Background Utilization	None
CPU Resource Parameters	Single Processor
IP Host Parameters	(...)
IP Processing Information	Default
Server Address	Auto Assigned
Server: Advanced Server Configuration	Sun Ultra 10 333 MHz
Server: Modeling Method	Simple CPU
TCP Parameters	Default

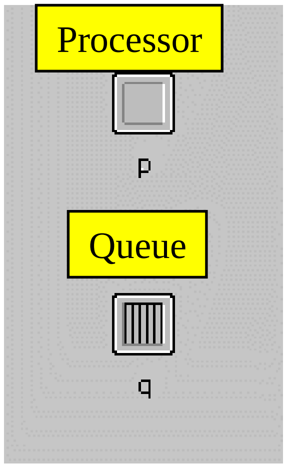
☐ Apply Changes to Selected Objects

☐ Advanced

Node Modeling

Node Objects: Processors and Queues

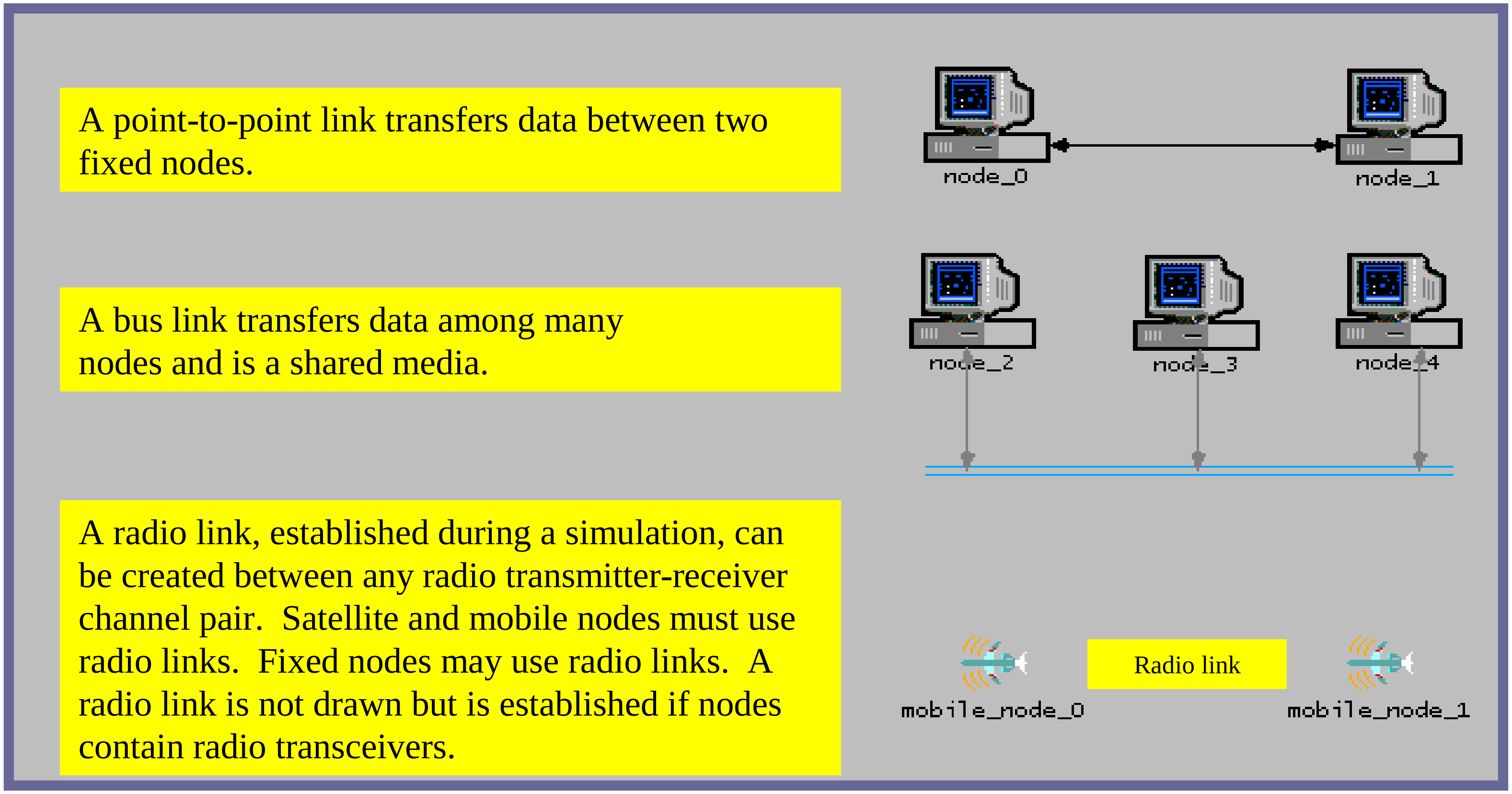
- Processors
  - General-purpose building blocks of node models
  - Fully programmable
- Queues
  - Offer all the functionality of processors
  - Can also buffer and manage a collection of data packets



Link 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.

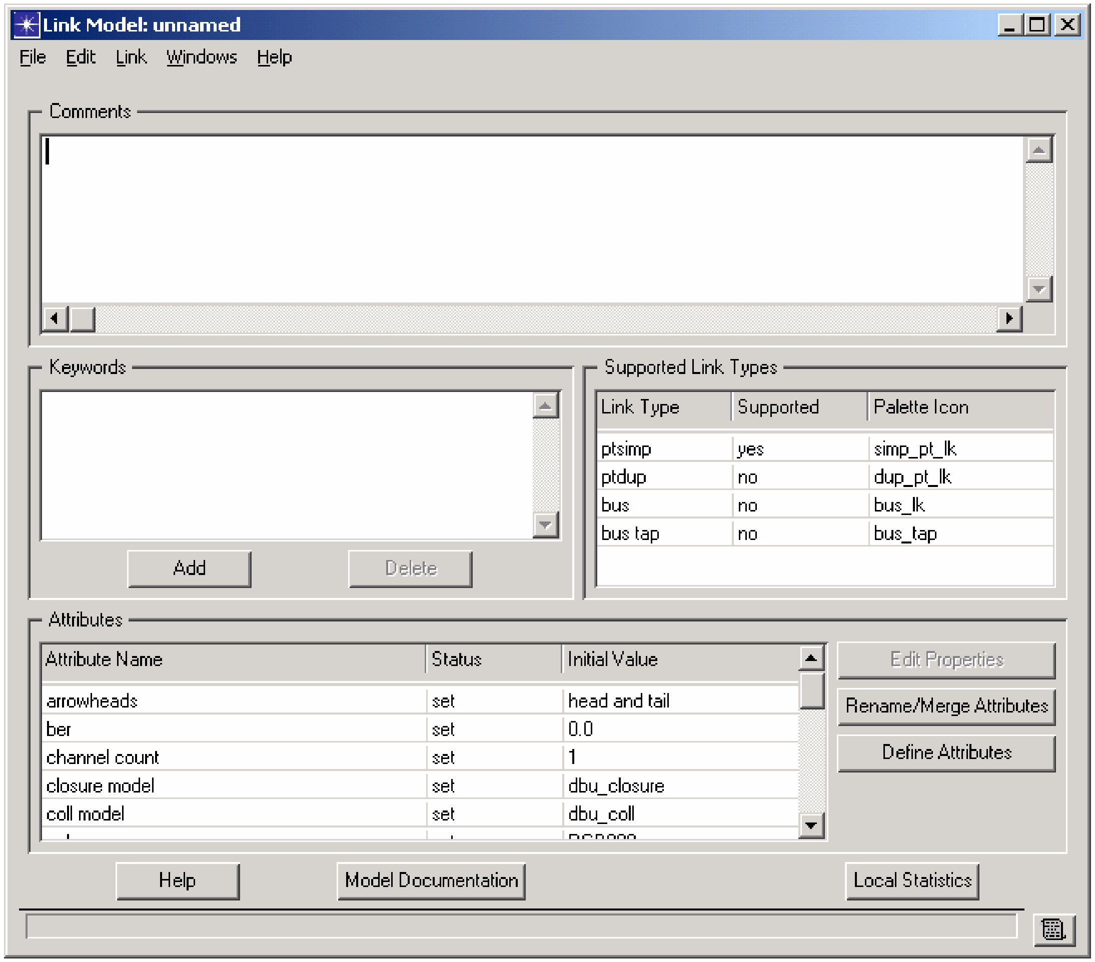
A bus link transfers data among many nodes and is a shared media.

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.

Radio link

Link Editor

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



Link Type	Supported	Palette Icon
ptsimp	yes	simp_pt_lk
ptdup	no	dup_pt_lk
bus	no	bus_lk
bus tap	no	bus_tap

Attribute Name	Status	Initial Value
arrowheads	set	head and tail
ber	set	0.0
channel count	set	1
closure model	set	dbu_closure
coll model	set	dbu_coll

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

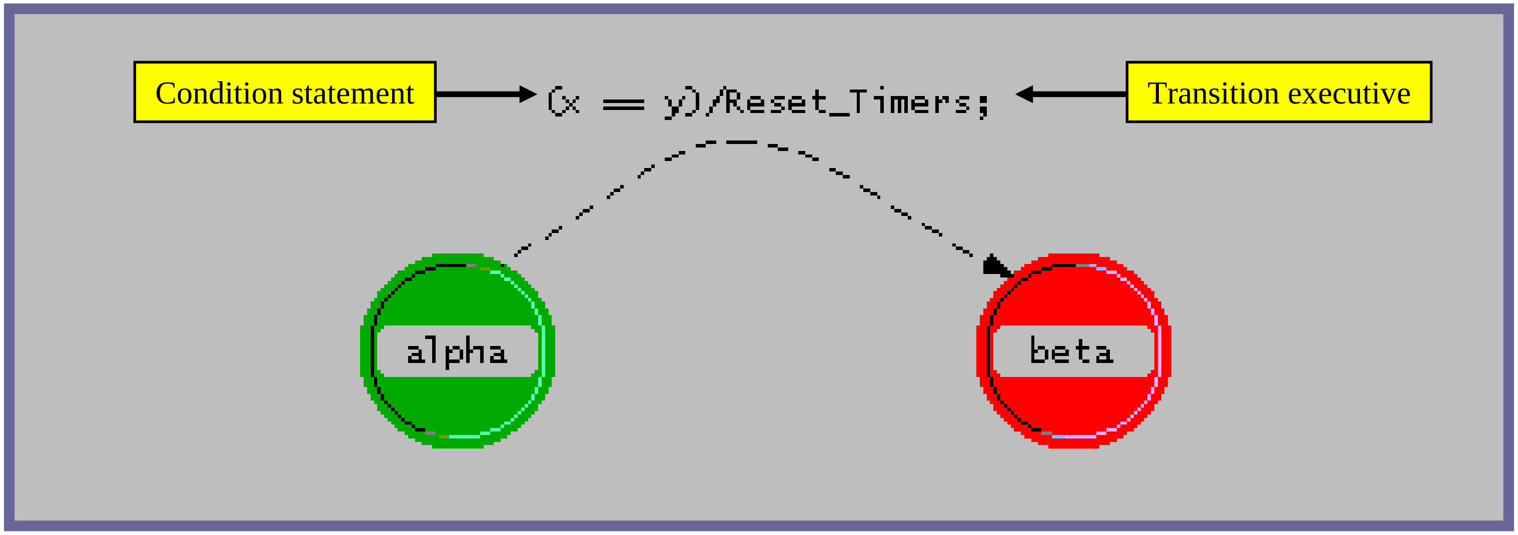
Process Modeling

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

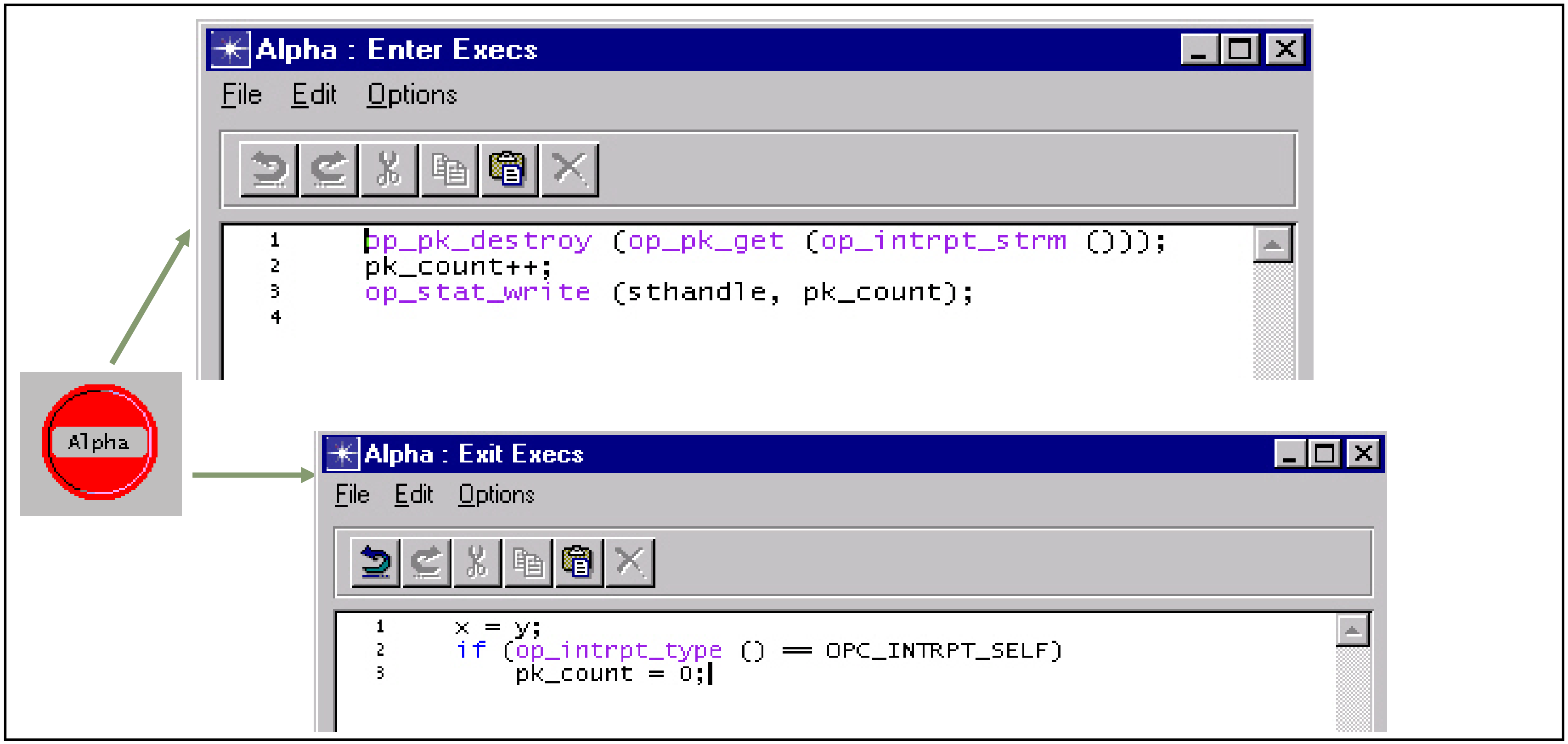
State Transitions

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



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

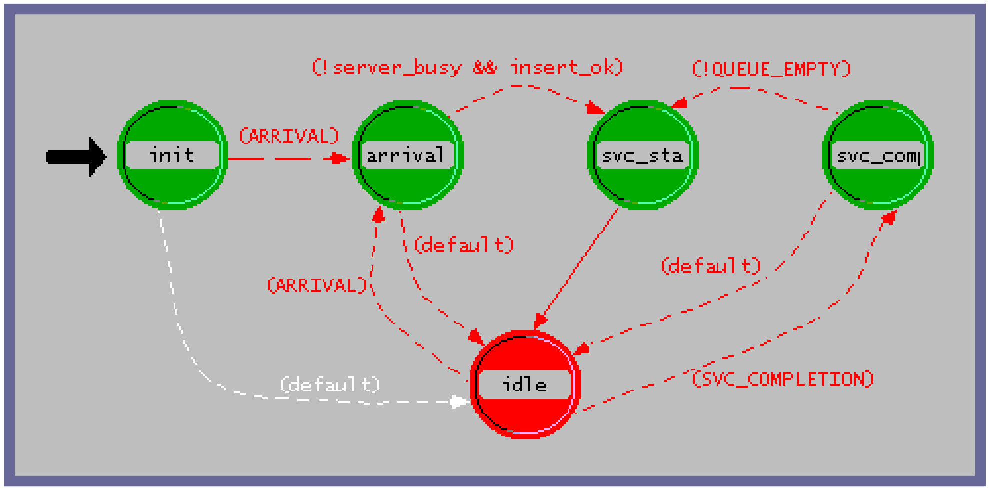


```
op_pk_destroy (op_pk_get (op_intrpt_strm ()));
pk_count++;
op_stat_write (sthandle, pk_count);
```

```
x = y;
if (op_intrpt_type () == OPC_INTRPT_SELF)
    pk_count = 0;
```

Proto-C™

- State transition diagrams
- C programming language
- Library of OPNET Kernel Procedures (KPs)
- State variables (private to each process)
- Temporary variables



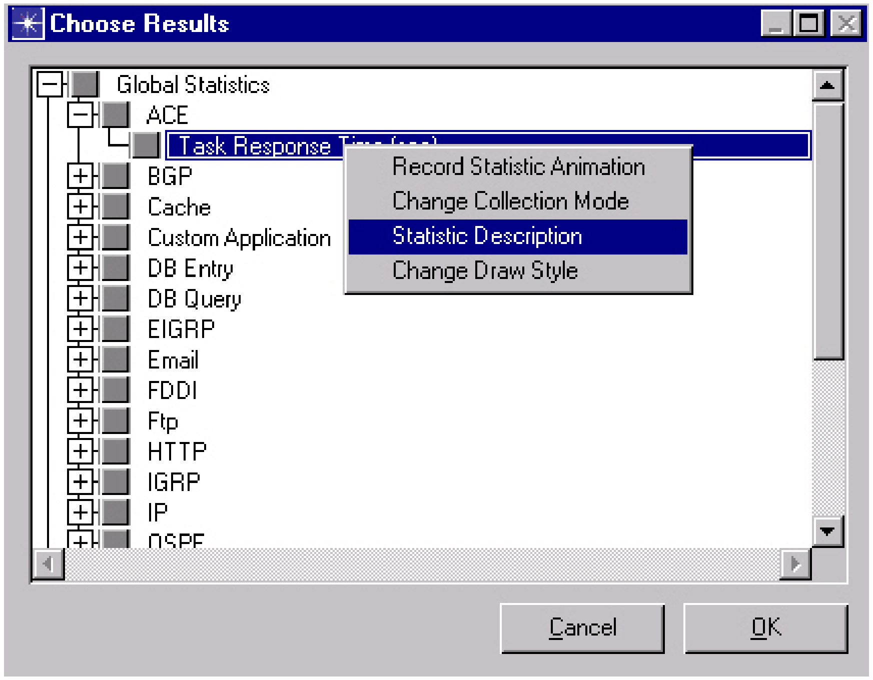
Running a Simulation

Statistic Collection

- Statistic Attributes
- Descriptions of Statistics
- Statistic collection modes

Statistic Attributes

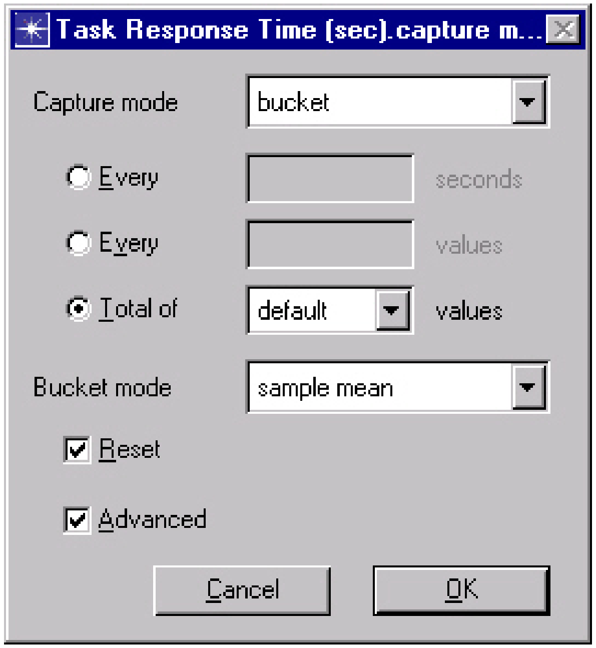
- Right-clicking on a statistic while in the Choose Results dialog box presents a menu of statistic attributes
- Right Click on the work-space and select Choose Individual Statistics to select the statistics to be measured during the simulation



- Statistic attributes include:
  - Record Statistic Animation
  - Change Collection Modes
  - Statistic Description
  - Change Draw Style

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



Task Response Time (sec).capture m...

Capture mode: bucket

○ Every ____ seconds

○ Every ____ values

● Total of default values

Bucket mode: sample mean

☑ Reset

☑ Advanced

Understanding Statistics

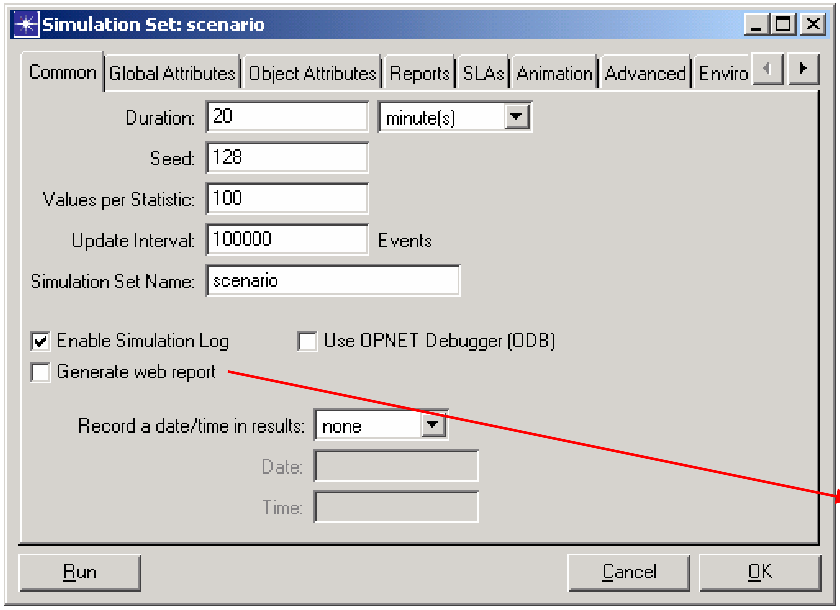
- It is essential to define the goals of the study and to understand the statistics needed to get useful results

- Browse available statistics and view their descriptions

- Understand the default collection mode to help interpret results

Configuring Simulations

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



Duration: 20 minute(s)

Seed: 128

Values per Statistic: 100

Update Interval: 100000 Events

Simulation Set Name: scenario

☑ Enable Simulation Log

□ Use OPNET Debugger (ODB)

□ Generate web report

Record a date/time in results: none

Date:

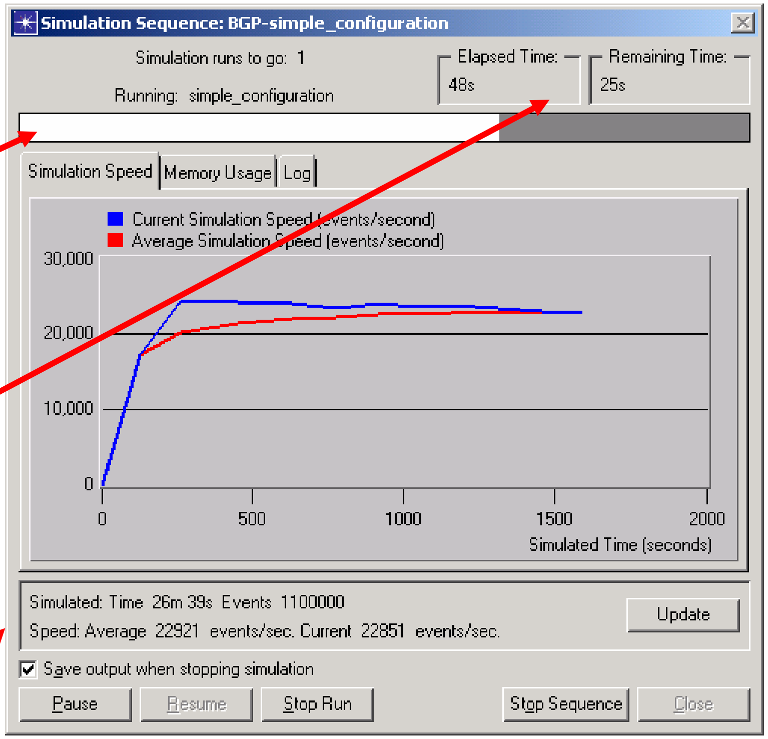
Time:

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
Generate Web Report	If checked, the simulation will produce a web report for the results.

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



Simulation Sequence: BGP-simple_configuration

Simulation runs to go: 1

Running: simple_configuration

Elapsed Time: 48s

Remaining Time: 25s

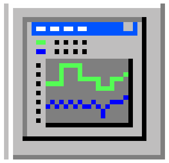
Simulated: Time 26m 39s Events 1100000

Speed: Average 22921 events/sec. Current 22851 events/sec.

☑ Save output when stopping simulation

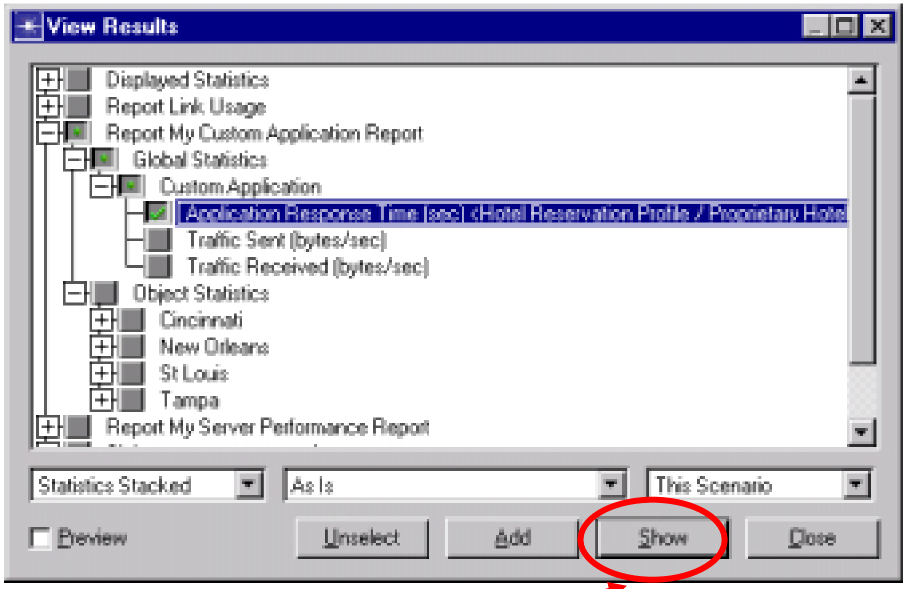
Viewing Results

- Results can be displayed by:
  - Selecting the "View Results" button on the tool bar
  - Selecting View Results from the **Results** 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



View Results

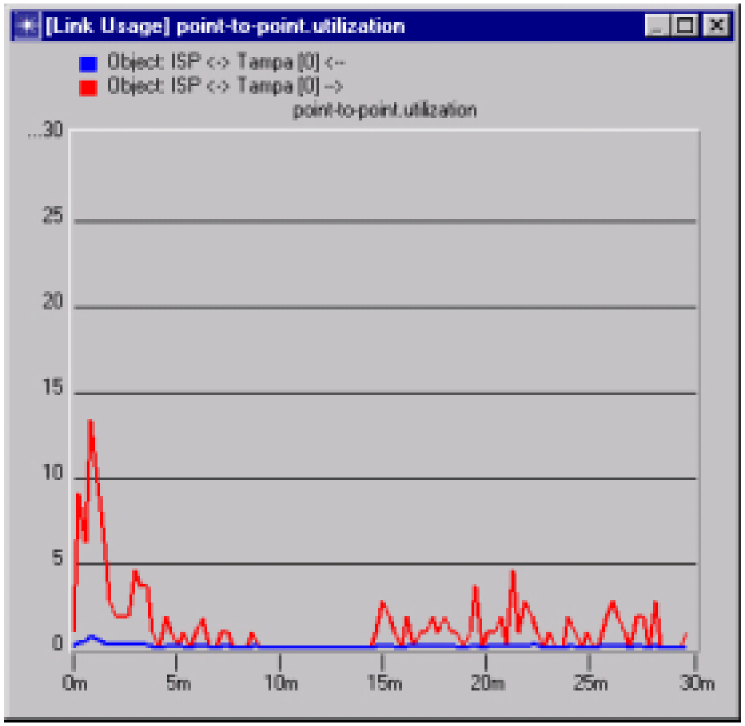
- Displayed Statistics
- Report Link Usage
- Report My Custom Application Report
  - Global Statistics
    - Custom Application
      - ☑ Application Response Time (sec) <Hotel Reservation Profile / Proprietary Hote
      - Traffic Sent (bytes/sec)
      - Traffic Received (bytes/sec)
  - Object Statistics
    - Cincinnati
    - New Orleans
    - St Louis
    - Tampa
- Report My Server Performance Report

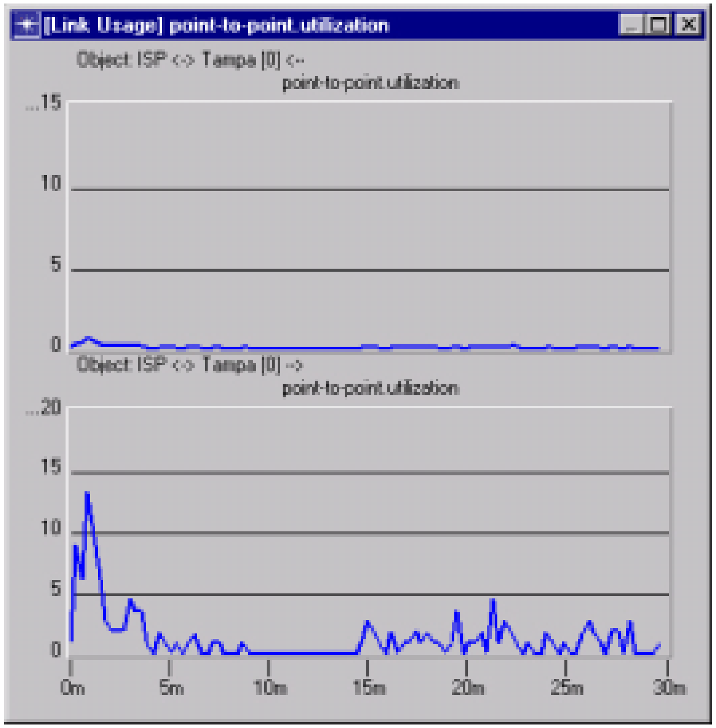
Statistics Stacked | As Is | This Scenario

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

Viewing Results (cont.)

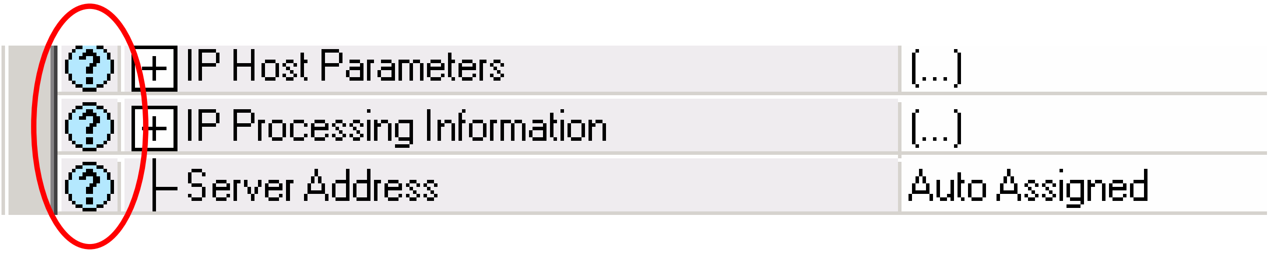
- Multiple graph panels can be displayed at the same time
- Each panel can contain one or more traces in an **Overlaid** or **Stacked** layout


[Link Usage] point-to-point.utilization
🟦 Object: ISP <-> Tampa [0] <--
🟥 Object: ISP <-> Tampa [0] -->
point-to-point.utilization
...30
25
20
15
10
5
0
0m
5m
10m
15m
20m
25m
30m


[Link Usage] point-to-point.utilization
Object: ISP <-> Tampa [0] <--
point-to-point.utilization
...15
10
5
0
Object: ISP <-> Tampa [0] -->
point-to-point.utilization
...20
15
10
5
0
0m
5m
10m
15m
20m
25m
30m

Where to Get Help

- Online Documentation from the **Help** menu
  - Online Tutorials
    - M/M/1 Queue
    - Basic Processes
- 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 over any object to get a brief description of that object
- Attribute help accessible by clicking on the question mark next to the attribute



? ⊞ IP Host Parameters	(...)
? ⊞ IP Processing Information	(...)
? ├ Server Address	Auto Assigned