

OPNET/Riverbed Modeler: Building Network Topologies

Roman Dunaytsev

The Bonch-Bruевич Saint-Petersburg
State University of Telecommunications

roman.dunaytsev@spbgut.ru

Lecture № 3

Outline

- 1 Network topologies
- 2 Object palette
- 3 Building network topologies
- 4 Verifying connectivity
- 5 Failing and recovering objects
- 6 Subnets
- 7 Annotation palette

Outline

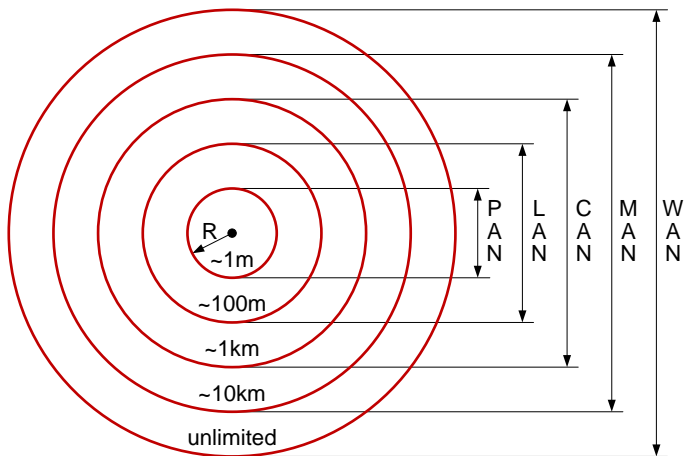
- 1 Network topologies
- 2 Object palette
- 3 Building network topologies
- 4 Verifying connectivity
- 5 Failing and recovering objects
- 6 Subnets
- 7 Annotation palette

Network Topologies

- **Personal Area Network (PAN)** – a network used for communications among devices close to a person
- **Local Area Network (LAN)** – a network covering a small geographic area, like a home, office, floor or a building
- **Campus Area Network (CAN)** – a network made up of an interconnection of LANs within a limited geographic area
- **Metropolitan Area Network (MAN)** – a network spanning a city
- **Wide Area Network (WAN)** – a network, which provides communications support to an area ranging in size from a region, country, or even a good portion of the entire world

Network Topologies (cont'd)

- Communication networks can be categorized by their scale
 - But beware of absolutes!

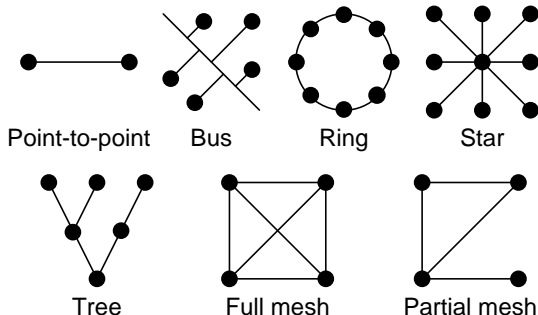


Network Topologies (cont'd)

| Characteristic | LANs | WANs |
|-----------------------|--|--|
| Number of users | Shared by a relatively small number of users | Shared by a large number of users |
| Topology | Usually limited to bus, ring, star, or tree | Virtually unlimited design capability |
| Data routing | Normally follow a fixed route | Use dynamic routing to reroute data in case of link failure or excessive traffic |
| Ownership | An organization that installs a LAN normally owns all of the components, including the cabling | The construction of a WAN requires the leasing of transmission facilities from one or more operators |
| Regulations | Primarily in the areas of building codes (level of electromagnetic emission, type of wiring, etc.) | Subject to a number of governmental regulations at the local and national levels |

Network Topologies (cont'd)

- Most LANs are designed to operate based on the interconnection of nodes that follow a specific topology
- **Physical topology** – describes how the nodes are physically connected
 - Some of physical topologies are variants of others



Network Topologies (cont'd)

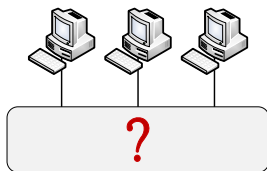
- **Logical topology** – describes how information is passed among nodes
- Basic logical topologies:
 - Bus
 - Ring
 - Star
- The physical and logical topologies are independent(!) of each other
 - A network may be laid out physically in one fashion but operate logically in an entirely different manner

Network Topologies (cont'd)

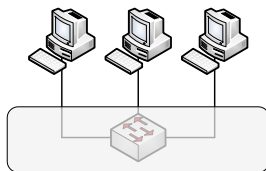
| Topology | Physical | Logical |
|-----------------|--|--|
| Bus | All nodes are connected to a single continuous cable | All data are broadcast to the entire network |
| Ring | All branches of the network are connected to a closed loop | Data flow from one node to the next in an ordered sequence; when the data reach the last node, they are returned to the originating node |
| Star | All branches of the network are connected through a central node | A central node directs all network transmissions so data are only delivered to the nodes they are intended for |

Network Topologies (cont'd)

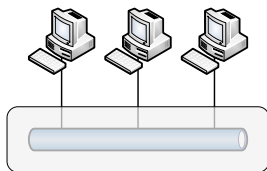
- Physical topology vs. logical topology



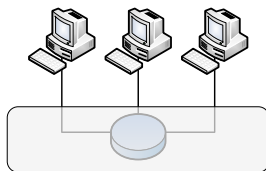
Physical topology: **star**



Logical topology: **star**



Logical topology: **bus**



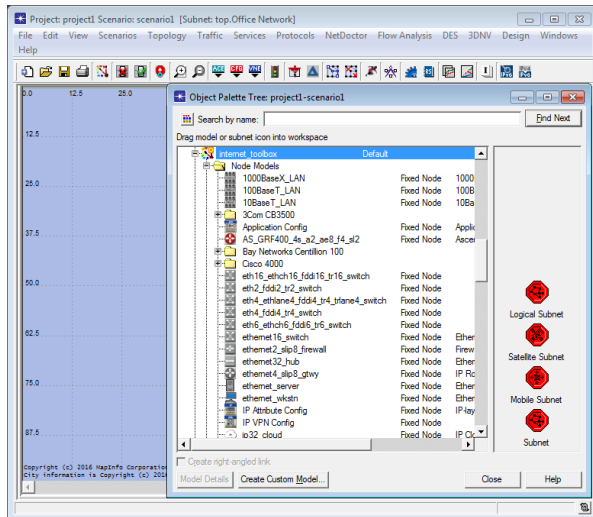
Logical topology: **ring**

Outline

- 1 Network topologies
- 2 Object palette**
- 3 Building network topologies
- 4 Verifying connectivity
- 5 Failing and recovering objects
- 6 Subnets
- 7 Annotation palette

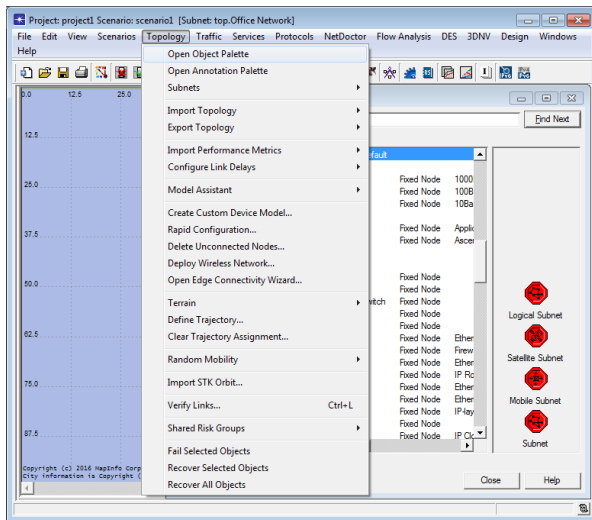
Object Palette

- **Object Palette** – provides access to all OPNET/Riverbed models



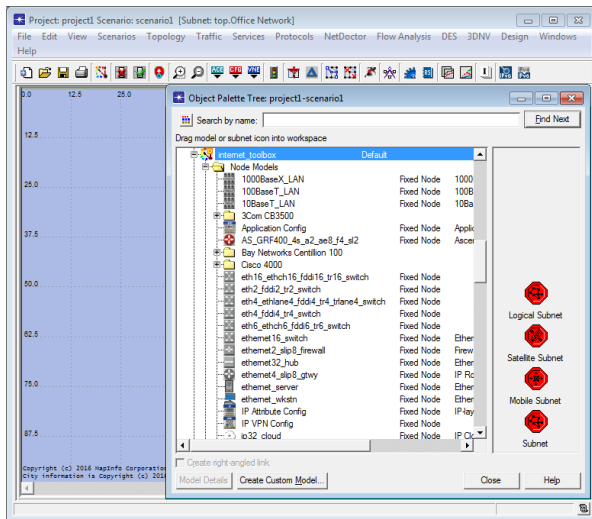
Object Palette (cont'd)

- Open Object Palette



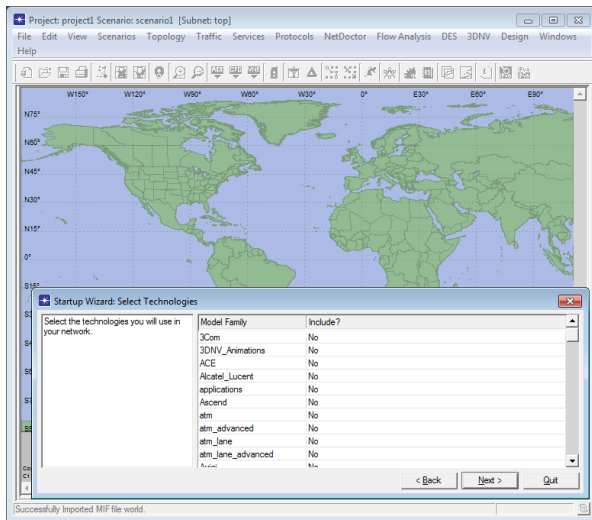
Object Palette (cont'd)

- **internet_toolbox** – the default model family



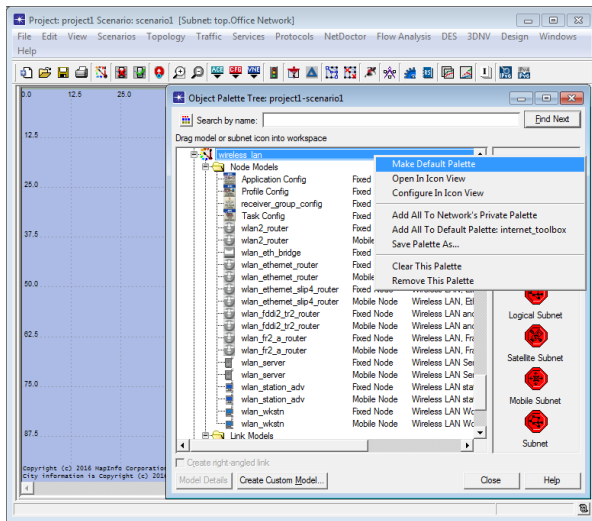
Object Palette (cont'd)

- Any other model families can be specified during project creation



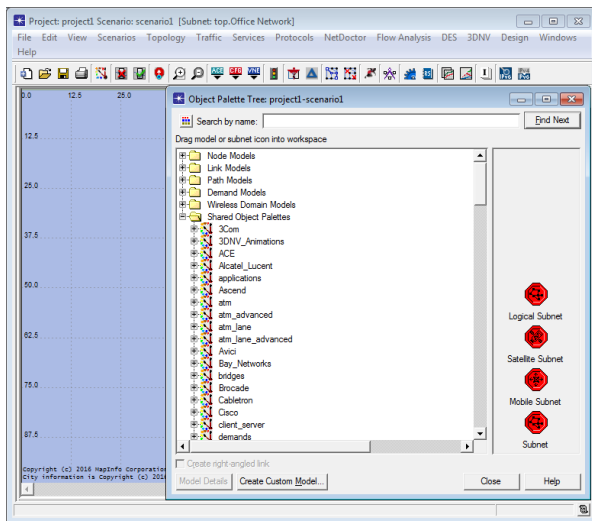
Object Palette (cont'd)

- You can change the scenario's default model family (default palette)



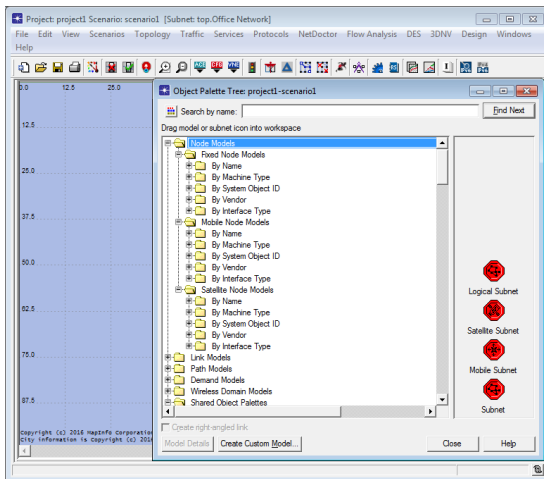
Object Palette (cont'd)

- Models: Node, Link, Path, Demand, Wireless Domain, Shared



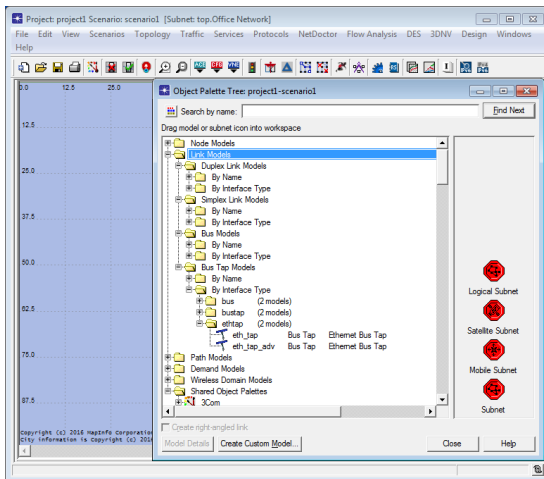
Object Palette (cont'd)

- **Node Models** – contains models of the communicating devices such as hubs, switches, routers, gateways, workstations, and servers



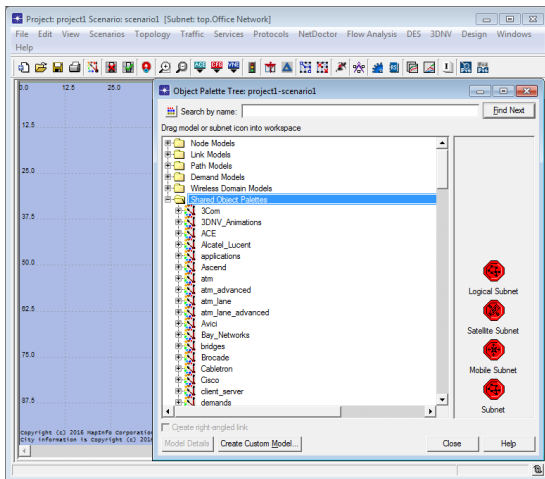
Object Palette (cont'd)

- **Link Models** – contains models of the links such as 1000Base-T Ethernet link, T1 duplex link, and 16 Mbps Token Ring



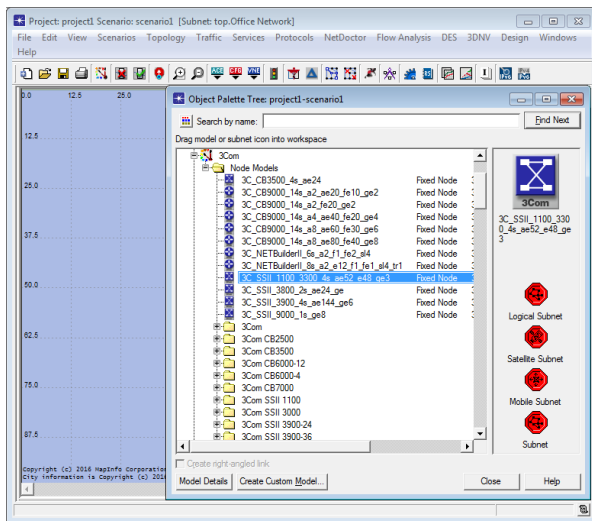
Object Palette (cont'd)

- **Shared Models** – contains a collection of different models grouped according to common properties such as vendors or technologies



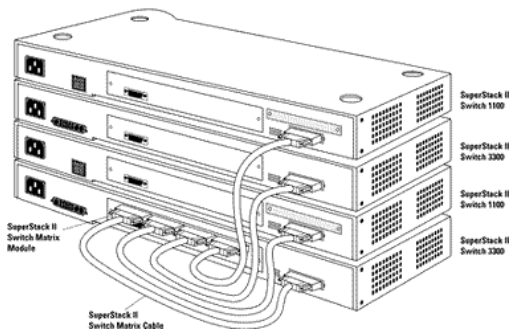
Object Palette (cont'd)

- Model naming conventions



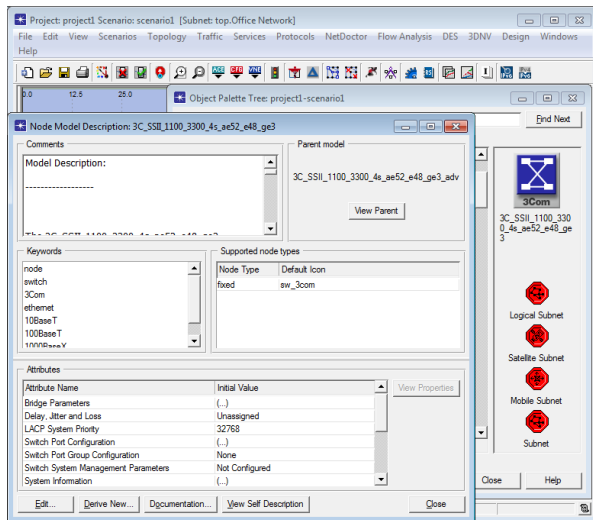
Object Palette (cont'd)

- 3C_SSII_1100_3300_4s_ae52_e48_ge3
 - 3C_..._4s = a stack of 4 switches manufactured by 3Com, Inc.
 - SSII_1100 = SuperStack II 1100
 - SSII_3300 = SuperStack II 3300
 - ae52 = 52 auto-sensing Ethernet ports
 - e48 = 48 Ethernet ports
 - ge3 = 3 Gigabit Ethernet ports



Object Palette (cont'd)

- See 'Model Description'



Object Palette (cont'd)

- Compare with an abstract node in ns-2 (network simulator v2):
 - `set node_0 [$ns node]`

The screenshot displays a network modeling application window titled "Project: project1 Scenario: scenario1 [Subnet: top.Office Network]". The interface includes a menu bar (File, Edit, View, Scenarios, Topology, Traffic, Services, Protocols, NetDoctor, Flow Analysis, DES, 3DNV, Design, Windows, Help), a toolbar, and a main workspace. A node icon labeled "3Com node_0" is selected in the workspace, and a context menu is open over it. The menu items include: Edit Attributes, Set Name, Unselect, Edit Node Model, Edit Documentation, Edit Attributes (Advanced), Add Ports..., Feature In Showcase, Attribute Reports, View Node Description (highlighted), Select Similar Nodes, Edit Similar Nodes, Fail This Node, Edit Aliases, Choose Individual DES Statistics, View Results, Open DES Log, Bring To Front, and Send To Back. The "Object Palette Tree" on the right shows a list of nodes, with "3C_SSII_1100_3300_4s_ae52_e48_ge3" selected. Below the tree, the "Node Description" for the selected node is displayed, detailing its hardware configuration and Layer-2 capabilities.

Object Palette Tree: project1-scenario1

Search by name: [] [Find Next]

Drag model or subnet icon into workspace

| Node Name | Type |
|--|------------|
| 3C_NETBuilderll_8s_a2_e12_f1_fe1_e14_tr1 | Fixed Node |
| 3C_SSII_1100_3300_4s_ae52_e48_ge3 | Fixed Node |
| 3C_SSII_3800_2s_ae24_ge | Fixed Node |
| 3C_SSII_3900_4s_ae144_ge6 | Fixed Node |
| 3C_SSII_9000_1s_ge8 | Fixed Node |
| 3Com | |
| 3Com CB2500 | |
| 3Com CB3500 | |

Node Description: 3C_SSII_1100_3300_4s_ae52_e48_ge3

Edit Options

Model Description:

The 3C_SSII_1100_3300_4s_ae52_e48_ge3 device model represents a stack of:

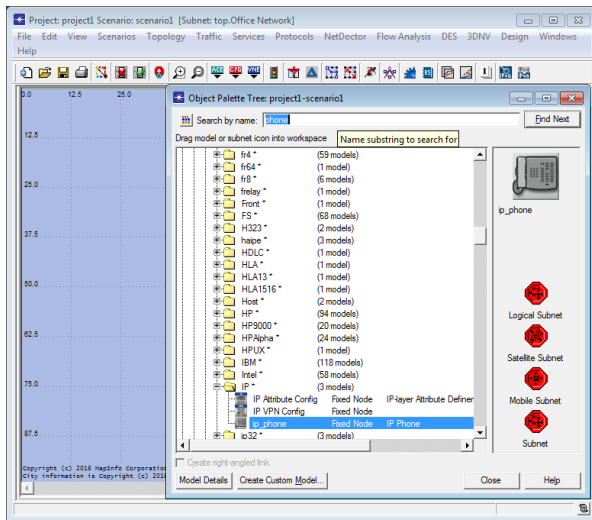
- two SuperStack II Switch 1100 chassis, and
- two SuperStack II Switch 3300 chassis.

Three of the above four chassis are equipped with an optional one-port 1000baseX Gigabit Ethernet module. The model represents only the Layer-2

Line: 1

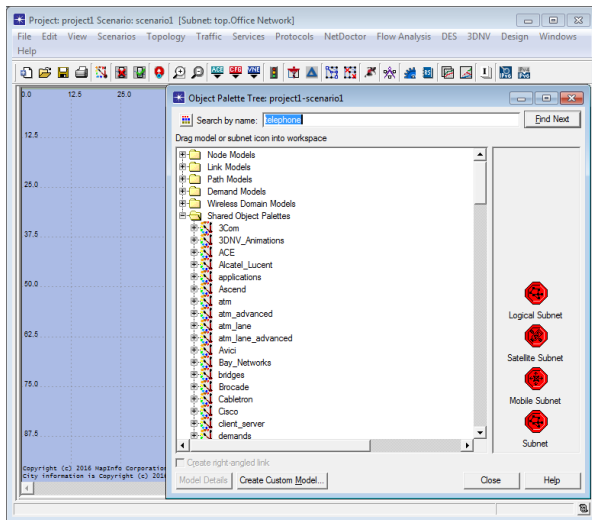
Object Palette (cont'd)

- Searching for models based on their names



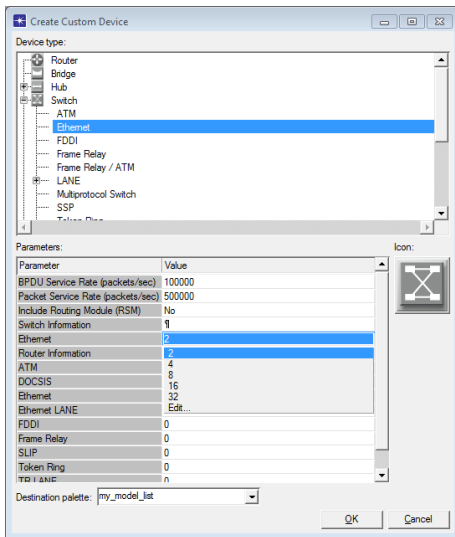
Object Palette (cont'd)

- Not all model names contain complete words!



Object Palette (cont'd)

- Use 'Create Custom Model' to create your own custom node models



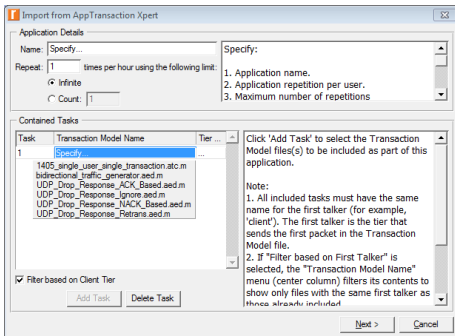
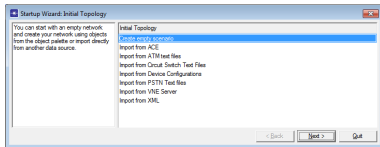
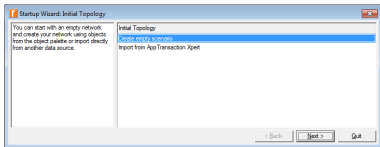
Outline

- 1 Network topologies
- 2 Object palette
- 3 Building network topologies**
- 4 Verifying connectivity
- 5 Failing and recovering objects
- 6 Subnets
- 7 Annotation palette

- Methods for building network topologies:
 - ① Manually, by dragging and dropping objects from the **Object Palette** window to the Project Editor workspace
 - ② Manually, using the Topology ⇒ **Rapid Configuration...** command from the toolbar to specify and build a complete network topology quickly
 - ③ Automatically, by **importing** the network model from an external data source – either a system that monitors your network or one or more data files that describe the network

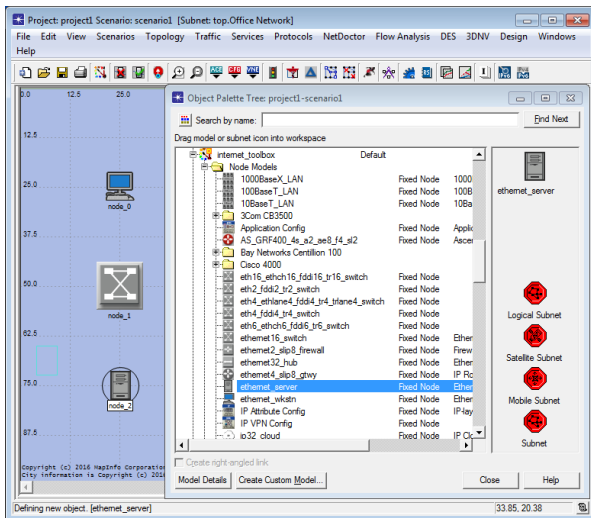
Building Network Topologies (cont'd)

- Riverbed Modeler Academic Edition 17.5 vs. **OPNET Modeler 14.5**



Building Network Topologies (cont'd)

- Adding nodes from Object Palette



Building Network Topologies (cont'd)

- Adding links from Object Palette

The screenshot displays a network modeling application window titled "Project: project1 Scenario: scenario1 [Subnet: top.Office Network]". The main workspace shows a network diagram with three nodes: "node_0" (a laptop), "node_1" (a switch), and "node_2" (a server). A red dashed line represents a link between "node_0" and "node_1", and another red dashed line connects "node_1" and "node_2".

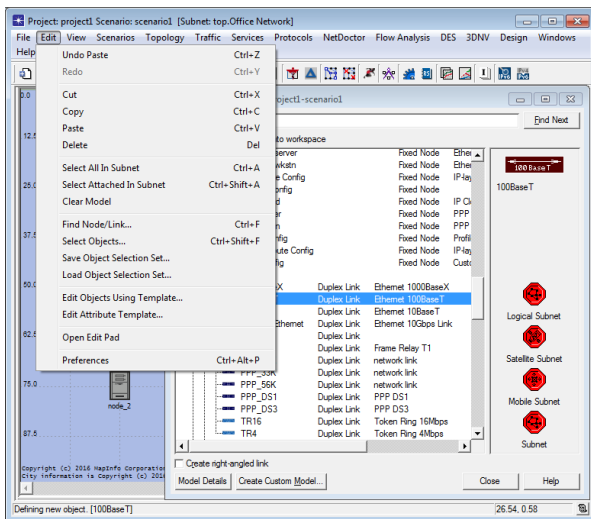
The "Object Palette Tree" is open, showing a search bar and a list of objects. The "Link Models" section is expanded, and "100BaseT" is selected. The palette lists various link models with their properties:

| Link Model | Type | Protocol |
|-----------------|-------------|----------------------|
| 100BaseX | Duplex Link | Ethernet 100BaseX |
| 100BaseT | Duplex Link | Ethernet 100BaseT |
| 10BaseT | Duplex Link | Ethernet 10BaseT |
| 10Gbps_Ethernet | Duplex Link | Ethernet 10Gbps Link |
| FDI | Duplex Link | |
| FR_T1 | Duplex Link | Frame Relay T1 |
| PPP_28K | Duplex Link | network link |
| PPP_33K | Duplex Link | network link |
| PPP_56K | Duplex Link | network link |
| PPP_DS1 | Duplex Link | PPP DS1 |
| PPP_DS3 | Duplex Link | PPP DS3 |
| TR16 | Duplex Link | Token Ring 16Mbps |
| TR4 | Duplex Link | Token Ring 4Mbps |

At the bottom of the palette, there are options to "Create right-angled link" and "Model Details", along with "Close" and "Help" buttons. The status bar at the bottom indicates "Defining new object. [100BaseT]" and the coordinates "36.15, 48.85".

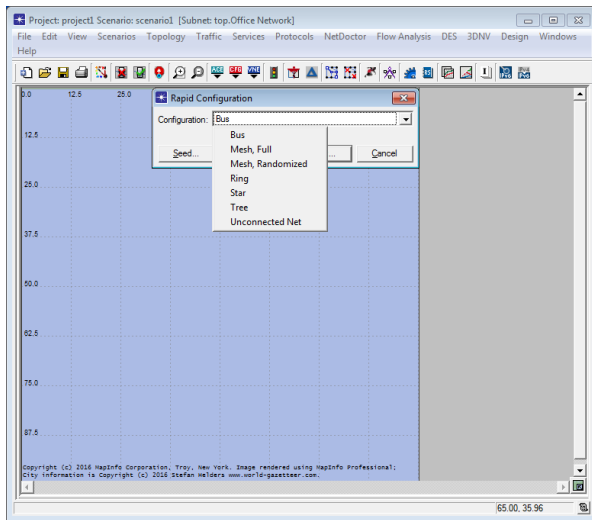
Building Network Topologies (cont'd)

- To delete nodes or links, select them and use DEL or Edit ⇒ Delete



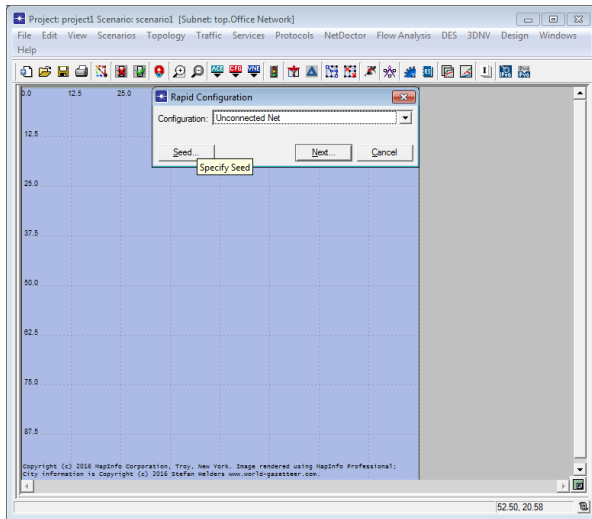
Building Network Topologies (cont'd)

- Topology \Rightarrow Rapid Configuration...



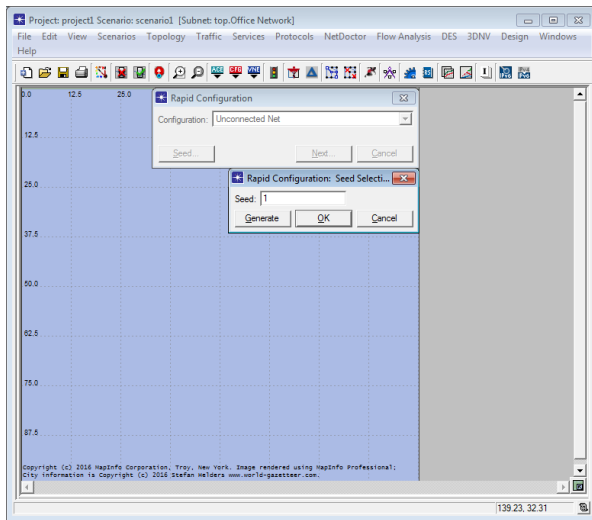
Building Network Topologies (cont'd)

- Specifying a **seed** value for the random number generation



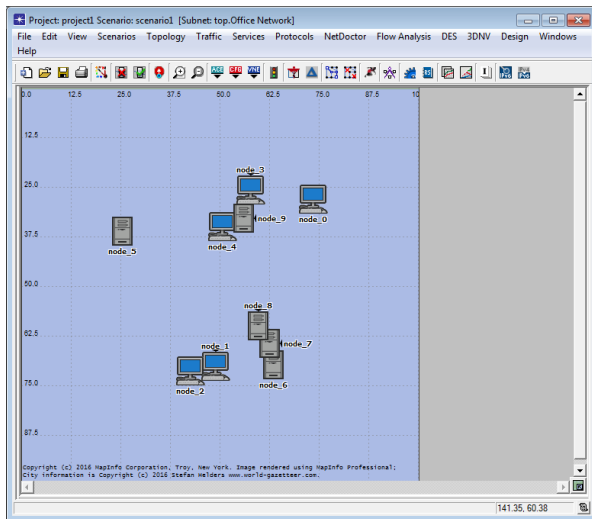
Building Network Topologies (cont'd)

- Some topologies may require randomness for node placement



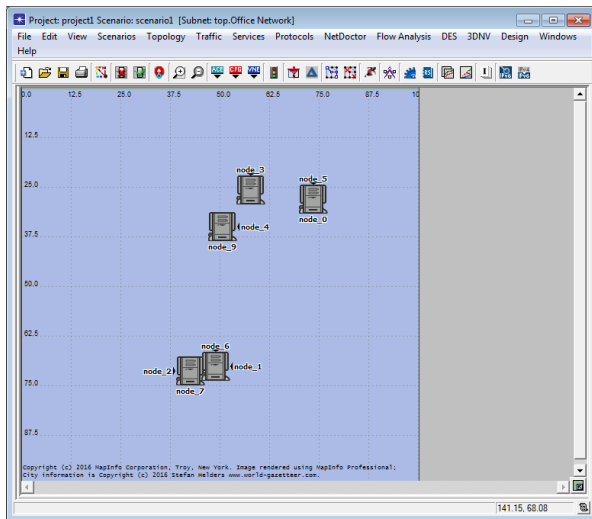
Building Network Topologies (cont'd)

- **Example 1:** 5 workstations (seed = 1) and 5 servers (seed = 2)



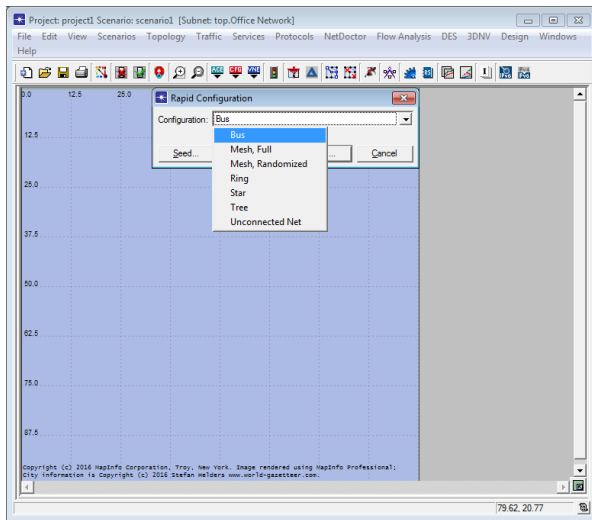
Building Network Topologies (cont'd)

- **Example 2: 5 workstations (seed = 1) and 5 servers (seed = 1)**



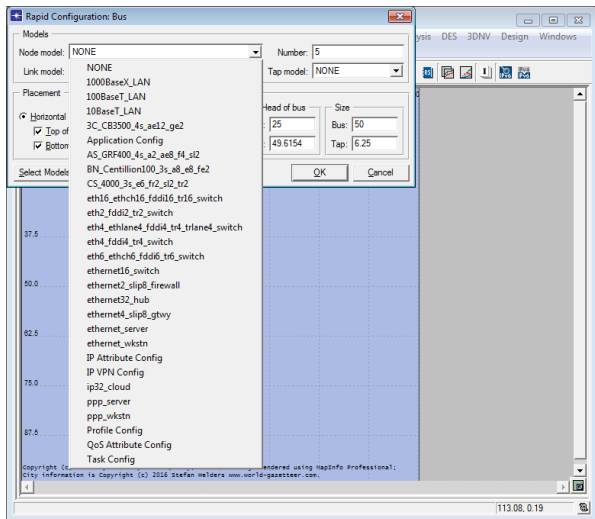
Building Network Topologies (cont'd)

- Select the desired topology configuration from the pull-down list



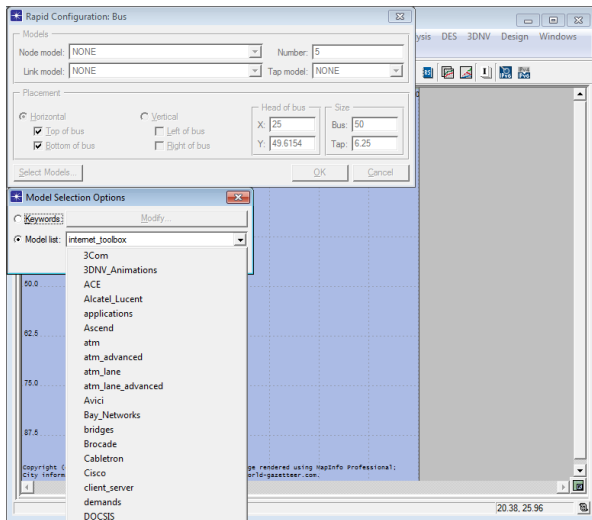
Building Network Topologies (cont'd)

- If the Node or Link model lists do not contain the models needed



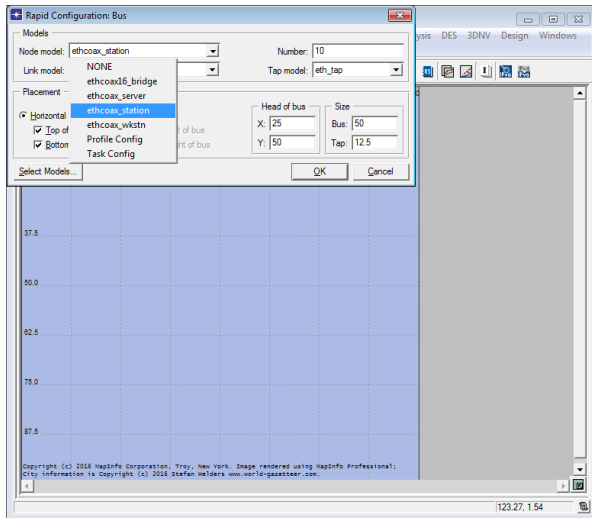
Building Network Topologies (cont'd)

- Then use 'Select Models...' to choose the necessary model family



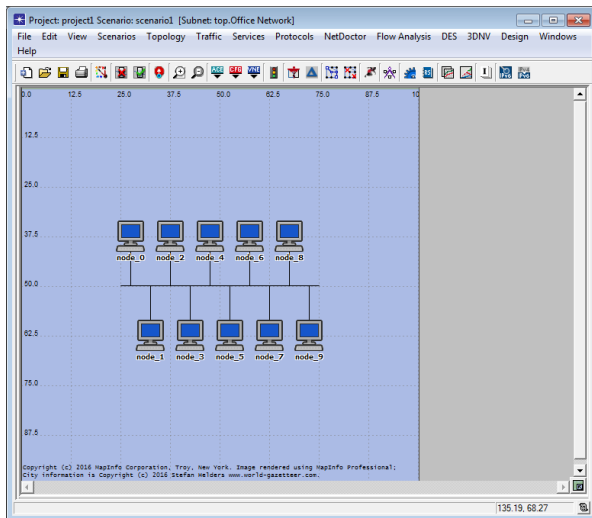
Building Network Topologies (cont'd)

- internet_toolbox ⇒ ethcoax



Building Network Topologies (cont'd)

- Creating a bus topology with Rapid Configuration



Outline

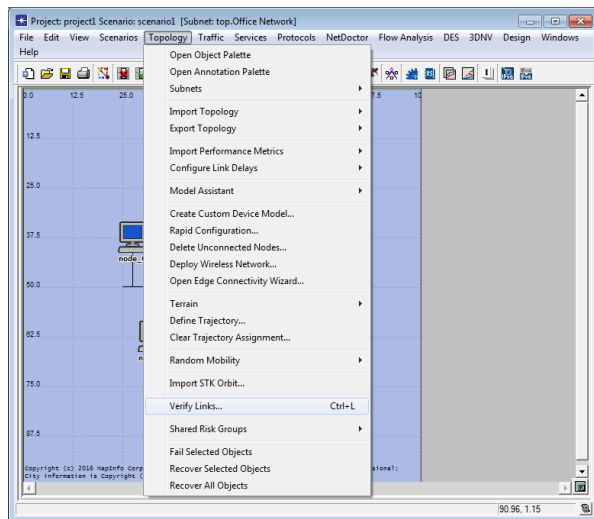
- 1 Network topologies
- 2 Object palette
- 3 Building network topologies
- 4 Verifying connectivity**
- 5 Failing and recovering objects
- 6 Subnets
- 7 Annotation palette

Verifying Connectivity

- Simply placing nodes into a project workspace and connecting them with links is not sufficient to create a working model of a network topology
- There are a variety of constraints on how nodes can be connected with links
- The simulation will not proceed if these constraints have not been followed
- OPNET/Riverbed Modeler provides sophisticated debugging tools

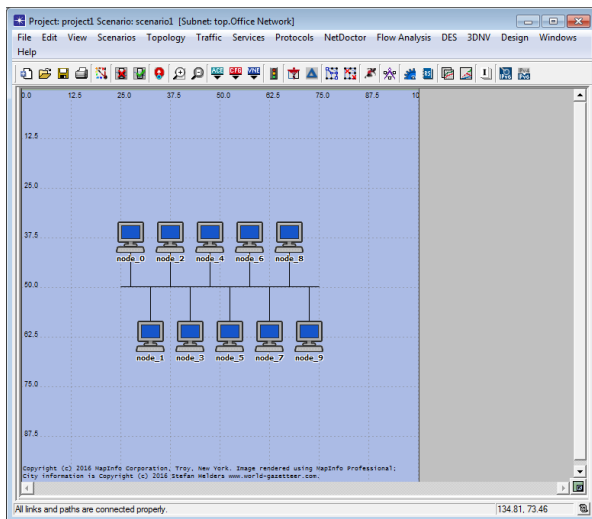
Verifying Connectivity (cont'd)

- Topology \Rightarrow Verify Links... \Rightarrow Verify links



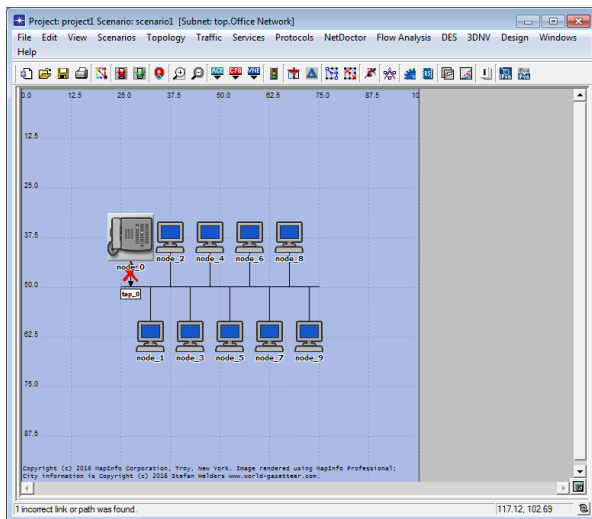
Verifying Connectivity (cont'd)

- All links and paths are connected properly 😊



Verifying Connectivity (cont'd)

- 1 incorrect link or path was found ☹️



Verifying Connectivity (cont'd)

- **Common mistakes in connecting nodes:**
- Connecting a link to a node that doesn't support the link's protocol
 - E.g., FDDI vs. ethernet_server
- Connecting more nodes to a device than the number of available ports
 - E.g., 17 ethernet_wkstn vs. ethernet16_switch
- Connecting a link to a wrong port
 - E.g., 5 100BaseT and 7 PPP_DS1 vs. ethernet4_slip8_gtwy

Outline

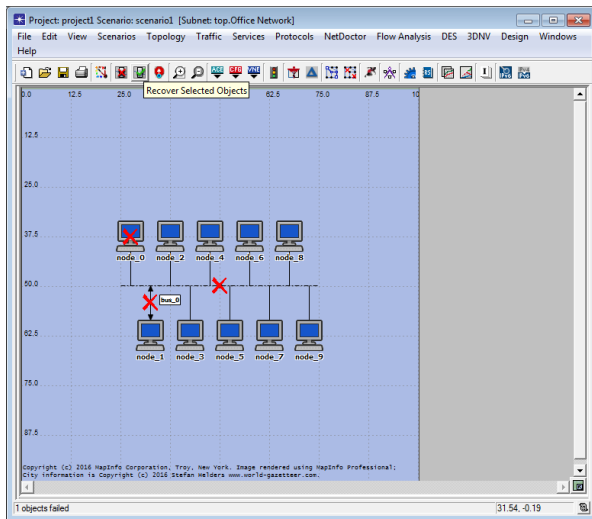
- 1 Network topologies
- 2 Object palette
- 3 Building network topologies
- 4 Verifying connectivity
- 5 Failing and recovering objects**
- 6 Subnets
- 7 Annotation palette

Failing and Recovering Objects

- Often it is important to determine how the system behaves when one or more nodes/links go down
- Methods for failing and recovering objects
 - 1 Manually, by removing objects from the Project Editor workspace and adding them back
 - 2 Manually, using the Topology \Rightarrow **Fail Selected Objects** and **Recover Selected Objects** commands from the toolbar
 - 3 Automatically, using the **Failure Recovery** node to fail and recover objects at specific times during the simulation

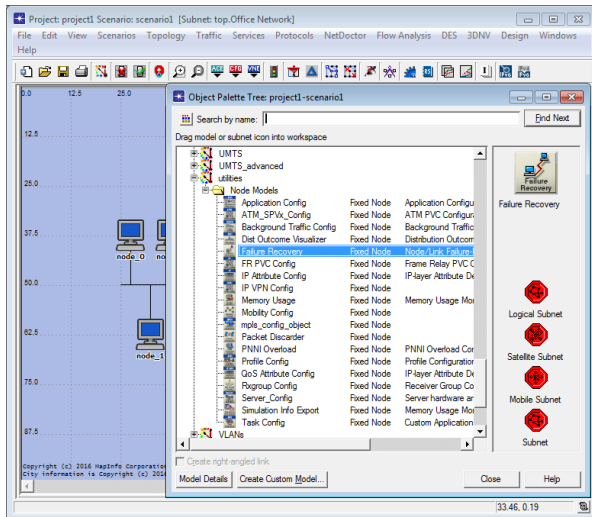
Failing and Recovering Objects (cont'd)

- Fail/Recover Selected Objects



Failing and Recovering Objects (cont'd)

- Failure Recovery



Failing and Recovering Objects (cont'd)

- node_0 will fail at time 100 and then recover at time 200 seconds

The screenshot shows the NetDoctor software interface. The main window displays a network topology with five nodes (node_0, node_1, node_2, node_3, and node_10) connected in a star topology. Node_10 is highlighted with a red circle and a 'Failure Recovery' icon. The 'Attributes' window for node_10 is open, showing the following configuration:

| Attribute | Value |
|--|-----------------------|
| Name | node_10 |
| Failure/Recovery Modeling | Enabled |
| Link Failure/Recovery Specification | No Failure/Recovery |
| Link Failure/Recovery Specification File | NOT_USED |
| Node Failure Mode | Node Only |
| Node Failure/Recovery Specification | (...) |
| Number of Rows | 2 |
| Office Network.node_0 | |
| Name | Office Network.node_0 |
| Time (seconds) | 100 |
| Status | Fail |
| Office Network.node_0 | |
| Name | Office Network.node_0 |
| Time (seconds) | 200 |
| Status | Recover |

The 'Advanced' checkbox is checked, and the 'Apply to selected objects' checkbox is unchecked. The 'Exact match' checkbox is also unchecked. The 'OK' and 'Cancel' buttons are visible at the bottom right of the window.

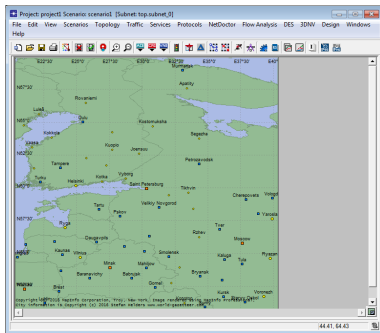
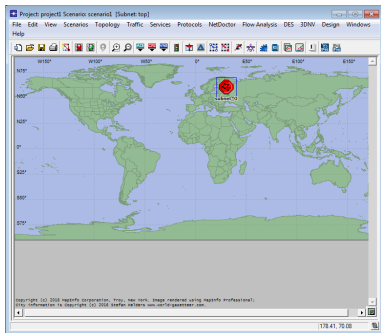
Outline

- 1 Network topologies
- 2 Object palette
- 3 Building network topologies
- 4 Verifying connectivity
- 5 Failing and recovering objects
- 6 Subnets**
- 7 Annotation palette

- **Subnet (aka subnetwork)** – a container that encompasses a set of nodes and links to represent a physical or logical grouping of objects
- Subnets can also contain other subnets
- Subnets within other subnets form the hierarchy of the network model
- Besides the objects it contains, the primary attributes of a subnet are:
 - Geographical position
 - Physical span
 - Mobility
- However, subnets may be used just to arrange network objects in a 'virtual topology' without considering physical properties such as geographic positions

Subnets (cont'd)

- **Fixed Subnet (Subnet)** – has a physical span specified in degrees



Subnets (cont'd)

- **Riverbed Modeler Academic Edition**: no x/y span ☹
 - Longitude = x position
 - Latitude = y position

Project: project1 Scenario: scenario1 [Subnet: top]

File Edit View Scenarios Topology Traffic Services Protocols NetDoctor Flow Analysis DES 3DNV Design Windows Help

W150° W120° W90° 0° E60° E100° E150°

N75°

(subnet_0) Attributes

| Attribute | Value |
|---------------------|----------------------|
| name | to-front_0 |
| priority | 0 |
| user id | 0 |
| x position | 30.32 |
| y position | 59.93 |
| x span | 34.8298000059 |
| y span | 24.8096085796 |
| threshold | 0.0 |
| map | world |
| icon name | subnet |
| outline color | blue |
| attribute promotion | disabled |
| doc file | rt_fixed_subnet |
| minimized icon | circle/#FF0000 |
| creation source | Object Palette |
| creation timestamp | 23:34:42 Feb 20 2018 |
| creation data | |
| label color | black |

Extended Attrs Object Documentation

Filter

Match: Exact Names Values Substring RegEx Possible values Tags

Advanced Apply to selected objects

OK Cancel

117.72, 21.11

Project: project1 Scenario: scenario1 [Subnet: top]

File Edit View Scenarios Topology Traffic Protocols DES Windows Help

W120° W90° 0° E60° E120° E150°

(subnet_0) Attributes

| Attribute | Value |
|---------------------|----------------------|
| name | to-front_0 |
| priority | 0 |
| user id | 0 |
| x position | 30.32 |
| y position | 59.93 |
| threshold | 0.0 |
| map | world |
| icon name | subnet |
| outline color | blue |
| attribute promotion | disabled |
| doc file | rt_fixed_subnet |
| minimized icon | circle/#FF0000 |
| creation source | Object Palette |
| creation timestamp | 23:43:43 Feb 20 2018 |
| creation data | |
| label color | black |
| rate | |

Extended Attrs Object Documentation

Filter

Match: Exact Names Values Substring RegEx Possible values Tags

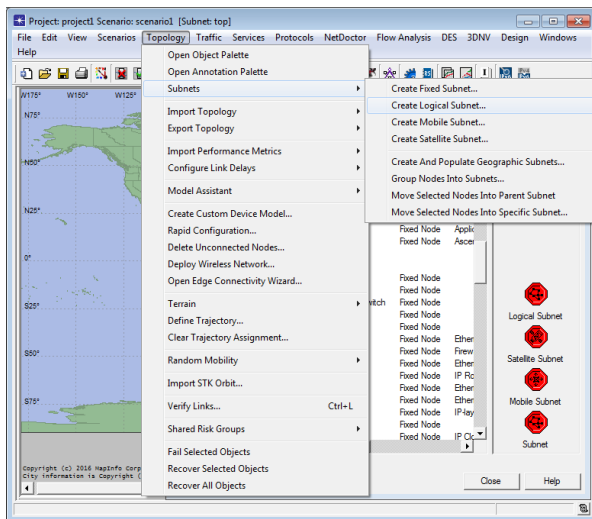
Advanced Apply to selected objects

OK Cancel

142.45, 52.12

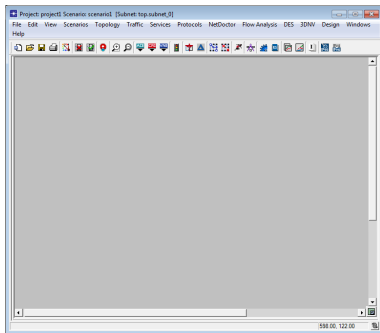
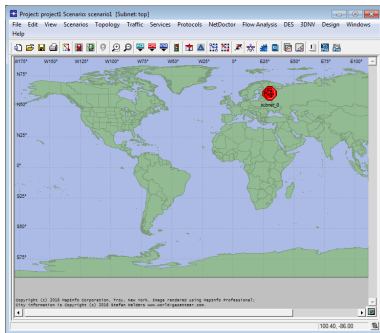
Subnets (cont'd)

- Use 'Create Logical Subnet...' or drag and drop 'Logical Subnet'



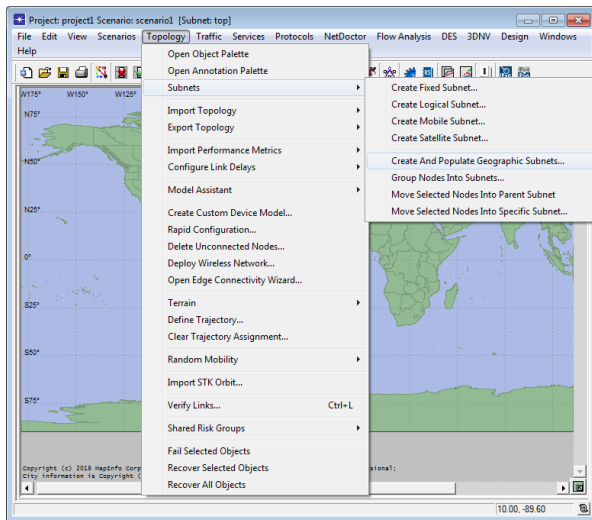
Subnets (cont'd)

- **Logical Subnet** – has no physical span
 - All objects inside are assumed to be in the same geographic location (i.e., the location of the logical subnet)



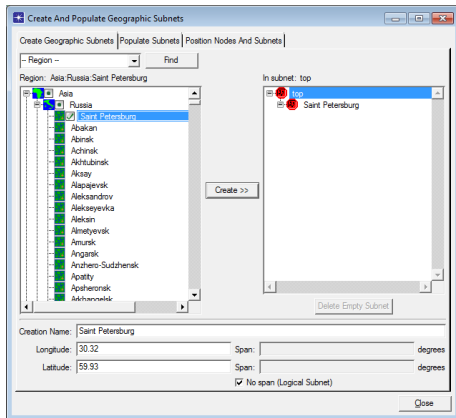
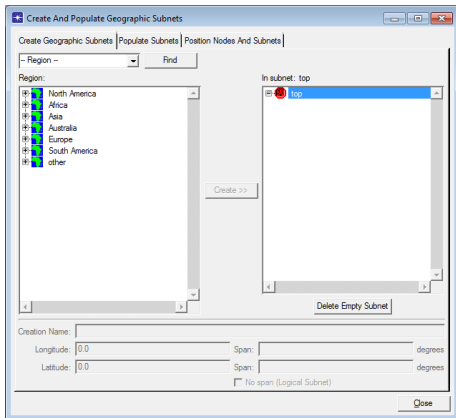
Subnets (cont'd)

- Create And Populate Geographic Subnets...



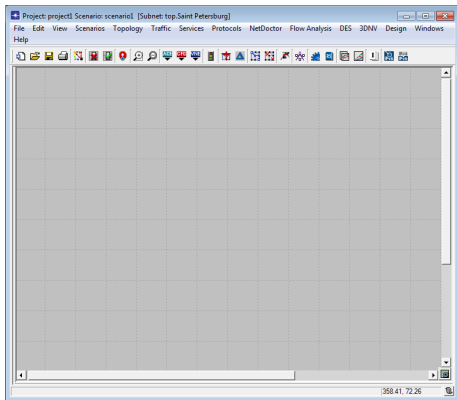
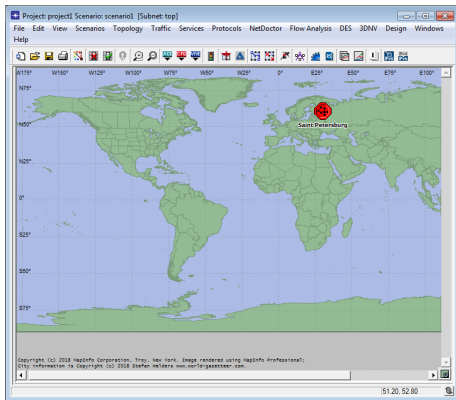
Subnets (cont'd)

- Asia ⇒ Russia ⇒ Saint Petersburg ⇒ Create »



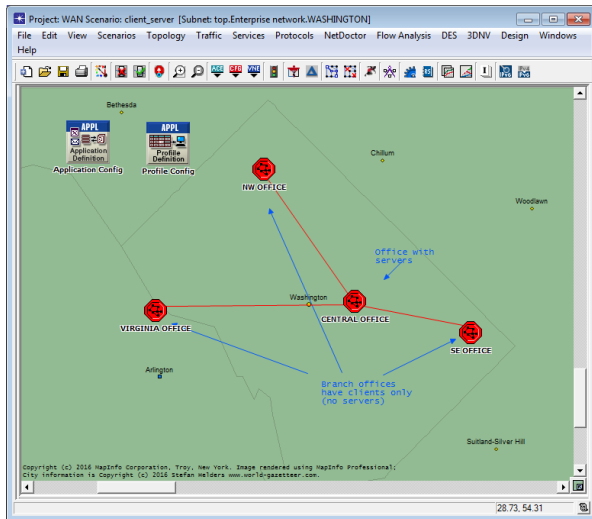
Subnets (cont'd)

- Saint Petersburg (logical subnet)



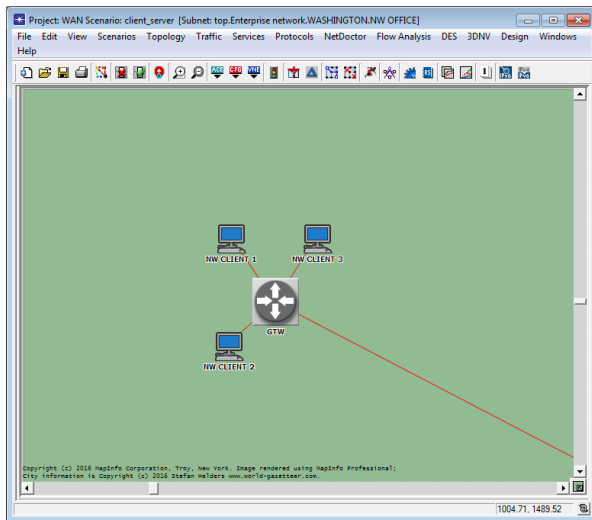
Subnets (cont'd)

- Add one or more subnets



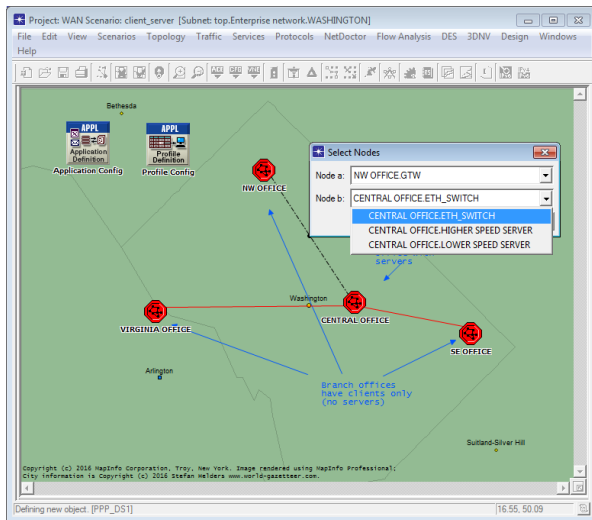
Subnets (cont'd)

- Populate each of the created subnets with network objects



Subnets (cont'd)

- Utility objects should be added only once, usually in the top subnet



Outline

- 1 Network topologies
- 2 Object palette
- 3 Building network topologies
- 4 Verifying connectivity
- 5 Failing and recovering objects
- 6 Subnets
- 7 Annotation palette**

Annotation Palette

- Topology ⇒ Open Annotation Palette

