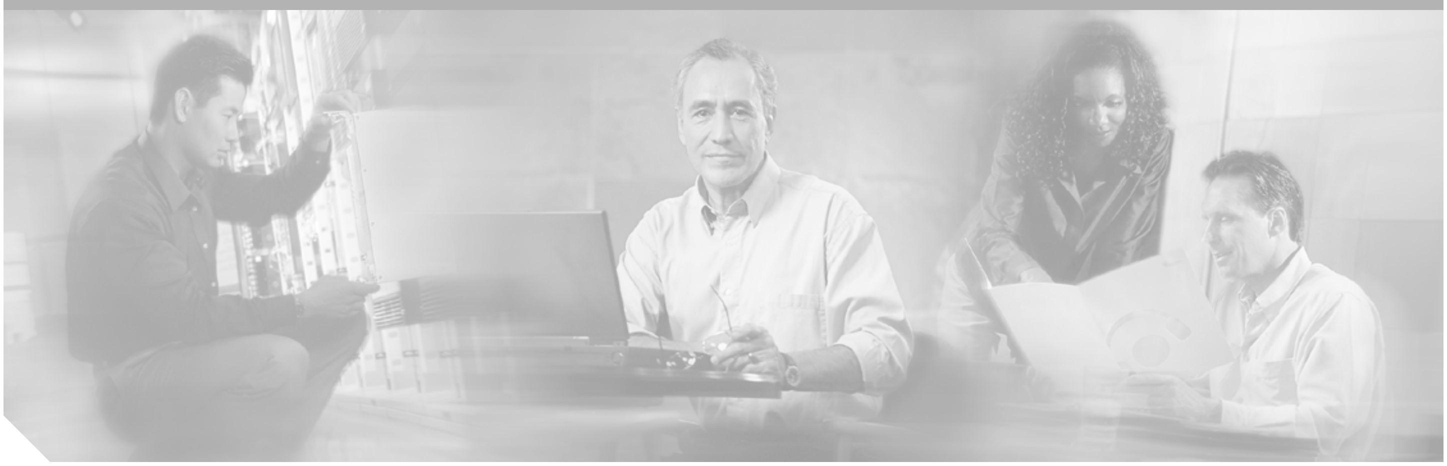


Cisco Network Planning Solution Design and Analysis Design Module User Guide

Software Release 11.0

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-7540-01

Cisco Network Planning Solution
Design and Analysis
Design Module User Guide

Copyright

Document Copyright

Document Title: Design Module User Guide
Document Part Number: D00267
Version: 5

Software Copyright

Product Name: IT Guru
Product Release: 11.0

Documentation Conventions

OPNET documentation uses specific formatting and typographic conventions to present the following types of information:

- Objects, examples, and system I/O
- Object hierarchies, notes, and warnings
- Computer commands
- Lists and procedures

Objects, Examples, and System I/O

- Directory paths and file names are in plain Courier typeface:

  ```
  opnet\release\models\std\ip
  ```

- Function names in body text are in italics:

  op_dist_outcome()

- The names of functions of interest in example code are in bolded Courier typeface:

  ```
  /* determine the object ID of packet’s creation module */
  src_mod_objid = op_pk_creation_mod_get (pkptr);
  ```

- Variables are enclosed in angle brackets (< >):

  ```
  <opnet_user_home>/op_admin/err_log
  ```

Object Hierarchies, Notes, and Warnings

Menu hierarchies are indicated by right angle brackets (>); for example:

Open File > Print Setup > Properties...

Attribute hierarchies are represented by angled arrows (➘) that indicate that you must drill down to a lower level of the hierarchy:

Attribute level 1 ➘ Attribute level 2 ➘ Attribute level 3

Note—Notes are indicated by text with the word Note at the beginning of the paragraph. Notes advise you of important supplementary information.

WARNING—Warnings are indicated by text with the word WARNING at the beginning of the paragraph. Warnings advise you of vital information about an operation or system behavior.

Computer Commands

These conventions apply to Windows systems and navigation methods that use the standard graphical-user-interface (GUI) terminology such as click, drag, and dialog box.

- Key combinations appear in the form "press **`<button>+x`**"; this means press the **`<button>`** and **`x`** keys *at the same time* to do the operation.
- The mouse operations *left-click* (or *click*) and *right-click* indicate that you should press the left mouse button or right mouse button, respectively.

Lists and Procedures

Information is often itemized in bulleted (unordered) or numbered (ordered) lists:

- In bulleted lists, the sequence of items is not important.
- In numbered lists, the sequence of items is important.

Procedures are contained within procedure headings and footings that indicate the start and end of the procedure. Each step of a procedure is numbered to indicate the sequence in which you should do the steps. A step may be followed by a description of the results of that step; such descriptions are preceded by an arrow.

Procedure FM-1 Sample Procedure Format

1 Procedure step.

➥ Result of the procedure step.

2 Procedure step.

End of Procedure FM-1

For more information about using and maintaining OPNET documentation, see the OPNET IT Guru Documentation Guide.

Document Revision History

Release Date	Product Version	Chapter	Description of Change
January 2004	10.5	Link Dimensioning	New chapter
		Link Dimensioning Resilient	New chapter
		Ring Backbone	New chapter
		Spanning Tree	New chapter
		Dual Tree	New chapter
		MPLS TE	Moved from Protocols menu (SP Guru version only)
		Design Actions	Moved chapter from Guru *User Guide*

Contents

List of Figures

List of Tables

List of Procedures

1 OPNET Design Actions

A *design action* is a high-level operation that automates the process of changing a network model, usually to achieve a design goal. With a design action, you can modify your network models with one operation. For example, you can run the MPLS TE design action to find primary and protection explicit routes (ERs) for each LSP.

The OPNET Model Library includes a set of standard design action models. Each design action combines internal design logic with a set of configurable attributes. You can configure a design action in the same way that you configure other types of objects. If you own a license to the OPNET Development Kit, you can also extend the standard design action models to embed your custom logic or create completely new custom design actions of your own.

Note—To use this feature, you must have a Design Module license.

Workflows for Design Actions

Design actions provide a very flexible way to make automated changes to your network. The workflow for a design action varies in complexity, depending on the extent of customization required.

Table 1-1 Workflows for Design Actions

Extent of Customization	Workflow / References
No customization	• Run action (Design > Run Design Action) • View results (Viewing Results on page DA-1-10)
Configure action	• Select action (Design > Configure/Run Design Action) • Configure attributes (Configuring a Design Action on page DA-1-3) • Run action (Running a Design Action on page DA-1-8) • View results (Viewing Results on page DA-1-10)
Select subactions / program blocks	• Select action (Design > Configure/Run Design Action) • Select subactions to fill specific functional roles (Compound Actions, Roles, and Subactions on page DA-1-4) • Configure attributes (Configuring a Design Action on page DA-1-3) • Run action (Running a Design Action on page DA-1-8) • View results (Viewing Results on page DA-1-10)
Modify / customize existing action	• Open existing action in Design Action Editor (Custom Design Actions on page DA-1-12) • Add new control attributes • Change embedded C/C++ code • Rebuild action
Create custom action	• Create entirely new action in Design Action Editor (Custom Design Actions on page DA-1-12) • Specify all information, including attributes and C/C++ code
End of Table 1-1	

Design Menu

The Design menu includes operations for selecting, configuring, running, and the viewing the results of design actions.

Table 1-2 Design Menu Operations

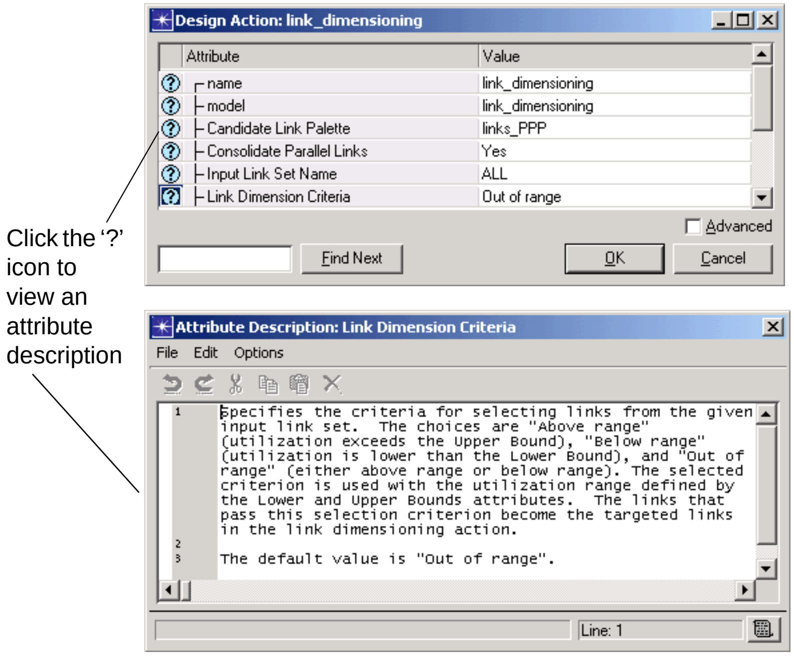
Menu item		**Reference**
Configure/Run Design Action	Select, configure, and run a design action	Configuring a Design Action on page DA-1-3
Run Design Action	Select and run a design action without configuring it	Running a Design Action on page DA-1-8
Quick Actions	Run a design action directly from the menu, without configuring it	Quick Actions on page DA-1-8
Results	Open Log—View log messages generated during the design action run View Output Tables—View result tables generated by the design action run	—
End of Table 1-2		

Configuring a Design Action

To configure a design action, choose Design > Configure/Run Design Action. The Configure/Run Design Action dialog box (Figure 1-3) shows a treeview of all available actions, a description of the selected action, and the list of visible attributes for the selected actions.

Every design action has one or more attributes that you can configure from this dialog box. You can also save a configured action under a new name, and thus create your own set of configured actions.

Figure 1-1 Attribute Table for a Design Action (Example)



Viewing Information about a Design Action

Because a design action can cause extensive changes to a scenario, it is good practice to research the effects and options of a design action thoroughly before you run it. There are three primary sources of information about a specific design action:

- Documentation—This manual includes chapters about specific design actions such as link dimensioning and MPLS traffic engineering.
- Action description—To view the action description, open the design action in the Configure / Run Action Dialog box and click the View Comments button.
- Attribute descriptions—To view descriptions of individual attributes, open the design action in the Configure / Run Action Dialog box; then click Edit Attributes. To view information about a specific attribute, click the Help ('?') icon as illustrated in Figure 1-1

Compound Actions, Roles, and Subactions

A *compound action* is a design action that can run other actions. A compound action has the following characteristics:

- Each compound action has one or more *roles* that define function blocks in the flowchart of the action. The function block is known by its role name, and runs other actions according to the internal flowchart of the compound action.

- The order in which the roles are listed in the user interface is not significant. Each compound action has its own internal flowchart that defines when and how each role is used.

- When an action is run by a role, it is called a *subaction*. If a compound action needs to execute a function block, it retrieves the subactions assigned to that role and runs each subaction in sequence.

- You can assign zero, one, or multiple subactions to a role. You might not want or need to assign a subaction to every role; a compound action skips over any role that has no subaction assigned. If a role has multiple subactions assigned, it runs subactions in the order in which they are listed.

Figure 1-2 Subactions Table (Example)

Role	Action(s)	
Initializer	link_pricer_2kb_100km	
Link Selector	link_price_selector_geq_25	
Link Action	remove_links	
Link Analyzer	link_price_analyzer	Click to configure...
Finalizer		

The Subactions table appears in the bottom right in the Configure Design Action dialog box. This table lists the roles and subactions defined for a specific compound action. To assign one or more subactions to a role, click in the Action(s) field for that role. In Figure 1-2, the compound action is configured to run four subactions:

- Set the financial cost on each link (“Initializer”)

- Select all links with a financial cost greater than 25 (“Link Selector”)

- Remove these links from the network (“Link Action”)

- Analyze the cost of the remaining links (“Link Analyzer”)

Derived Actions

A *derived action* is an action with customized attribute settings. You might want to create your own derived actions to perform common actions. For example, you might want to create actions to price links according to different criteria. You can derive actions from the link_pricer base action and then run the derived actions as needed.

Procedure 1-1 Creating a Derived Action

1 Choose Design > Configure/Run Design Action and select the action you want to use as a base model.

2 Configure the attributes and subactions (if any) for the base model.

3 Click Save As and save the action under a new name.

Note—You cannot overwrite a base action in the Project Editor; you can only save it under a new name. For this reason, the Save button is disabled when you configure a base action.

End of Procedure 1-1

Configure/Run Design Action Dialog Box

Figure 1-3 shows the Configure/Run Design Action dialog box and indicates the types of information that appear in the panes of the dialog box. The display and control elements in this dialog box are described in Table 1-3 on page DA-1-7.

Figure 1-3 Configure/Run Design Action Dialog Box

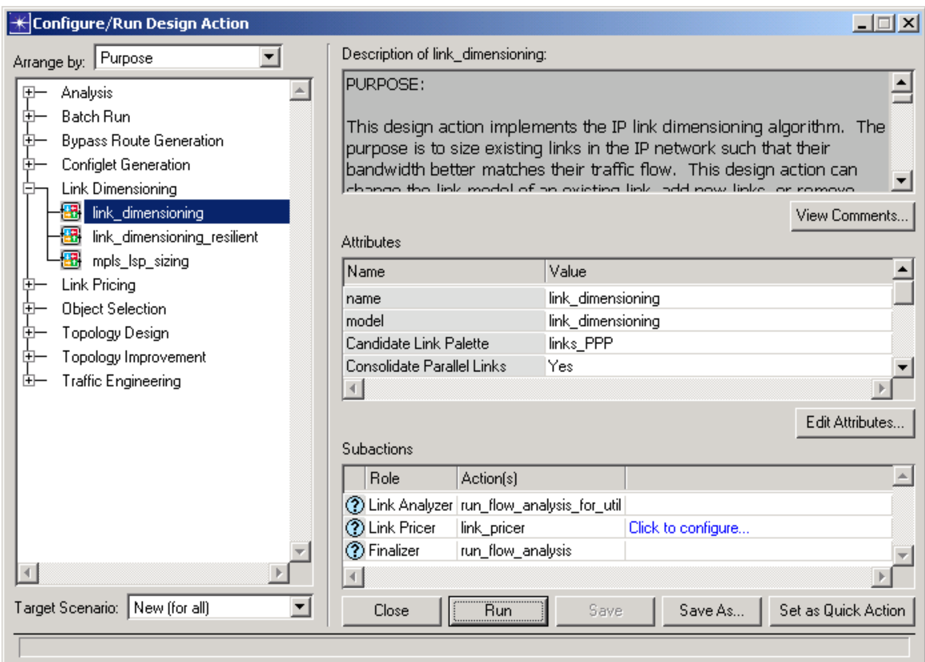


Table 1-3 Configure/Run, Design Action Dialog Box

Item	Description
Panes:	
• Treeview (left)	Select an action in this pane to view, configure, or run.
• Action description (right top)	Description of the selected action.
• Attributes table (right center)	Visible attributes of the selected action. Click Edit to configure these attributes.
• Subactions table (right bottom, compound actions only)	To assign one or more subactions to a role, click in the Action(s) field for that role. The Choose Subactions dialog box lists all assignable action models. To be assignable, an action model must have a Subaction Type self-description value that matches the role type.
Arrange By menu	Arrange the treeview based on the self-descriptions of action models. Thus you can arrange actions based on author, technology, or some other criteria.
Run	Run the selected action. If the action cannot be run directly, this button is disabled.
Save	Save the action based on the current configuration. If you change the configuration of a standard action, this button is disabled. You must choose Save As and save the current configuration under a new name.
Save As	Save the current configuration under a new name.
Set as Quick Action	Create a Quick Action from the current action and configuration. For more information, see Quick Actions on page DA-1-8.
Target Scenario menu	Choose the target scenario for the design action. The target scenario can be either the current scenario or a new scenario. **NOTE—**It is good practice to choose a new target scenario for any design action that can modify the network. This ensures that you retain a copy of the original (pre-design action) network state.
End of Table 1-3	

Running a Design Action

Every design action has an attribute called “runnable,” which specifies whether you can run the action directly. If this attribute is set to FALSE, the action can be run by a compound action only (that is, as a subaction). This means that

- when you choose Design > Run Design Action, only runnable actions appear.
- when you choose Design > Configure/Run Design Action and select an unrunnable action, the Run operation is unavailable.

You can run an action using any of the following operations:

1) Choose Design > Run Design Action (see Table 1-4 Run Action Dialog Box on page DA-1-9)

2) If you want to configure an action before running it, choose Design > Configure/Run Design Action (see Table 1-3 Configure/Run, Design Action Dialog Box on page DA-1-7)

3) If you have an action that you want to run quickly and repeatedly, you might want to create a quick action. You can then run the action from the Design > Quick Actions submenu (see Quick Actions on page DA-1-8)

Quick Actions

A *quick action* is an action that you can run directly from the Design > Quick Actions submenu. When you choose an action in this submenu, the action executes immediately. You can create your own quick actions, and thereby create your own “action toolbox” for a specific scenario.

Note—Each Project Editor session maintains its own Quick Actions list, so the Design > Quick Actions submenu is persistent within the current Project Editor session only.

Procedure 1-2 Creating a Quick Action

1 Choose Design > Quick Actions > Create New Quick Action to open the Configure/Run Design Action dialog box.

2 Select the action you want to use as the base for your quick action.

Note—You can only create a quick action from an action whose “runnable” attribute is set to TRUE. If you select an action and the Run button appears dimmed, this means that the action (as currently configured) is not runnable. If the “runnable” attribute is not visible, then you cannot use that action to create a quick action.

3 Click Edit.

4 Configure the attributes so that the action performs the behavior you want.

Note—Because a quick action executes automatically, you must specify the exact behavior you want the quick action to perform.

5 Verify that the “runnable” attribute is set to TRUE, then click OK to return to the Configure/Run Design Action dialog box.

6 Click Save As and save the action under the name you want to use for the quick action.

7 Click Quick Action.

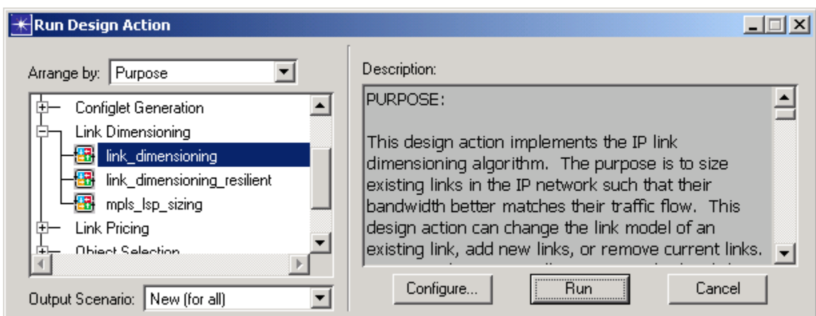
➥ The action appears in the Design > Quick Actions submenu.

End of Procedure 1-2

Run Action Dialog Box

Figure 1-4 shows the Run Action dialog box and indicates the types of information that appear in the panes of the dialog box.

Figure 1-4 Run Action Dialog Box



Treeview pane (only runnable actions appear in this tree)

Table 1-4 lists the display and control elements in the Run Action dialog box.

Table 1-4 Run Action Dialog Box

Item	Description
Treeview pane (left)	Select an action in this pane to configure or run.
Action description (right top)	Description of the selected action
Configure	Configure the attributes and subactions (if any) before running the action.
Run Design Action	Run the selected action.
End of Table 1-4	

Viewing Results

You can view the results of a design action using two different reporting mechanisms:

- The Design Action Log shows all network changes and significant events generated by the last action run; see Design Action Log for more information.
- In some cases, a design action might generate an output table of results. You can view these tables in the Results Viewer (Design > Results > View Output Table). For general information about viewing results, see Results Viewer on page ITU-14-46 of the *Guru User Guide*.

Design Action Log

The design-action log browser (Figure 1-5) shows all network changes and significant events generated by the last action run. The messages that appear in this log depend on the specific action. Generally, actions generate Summary messages to provide a high-level snapshot of the action results. Because a design action can generate many types of log messages, you might want to configure the log browser to show only messages of interest. You can open this browser using one of two methods:

- If a design action completes successfully, the Action Completed dialog box appears and includes a “View Log” button.
- To view the action log for the last run, choose Design > Results > View Log.

Figure 1-5 Design Action Log Browser

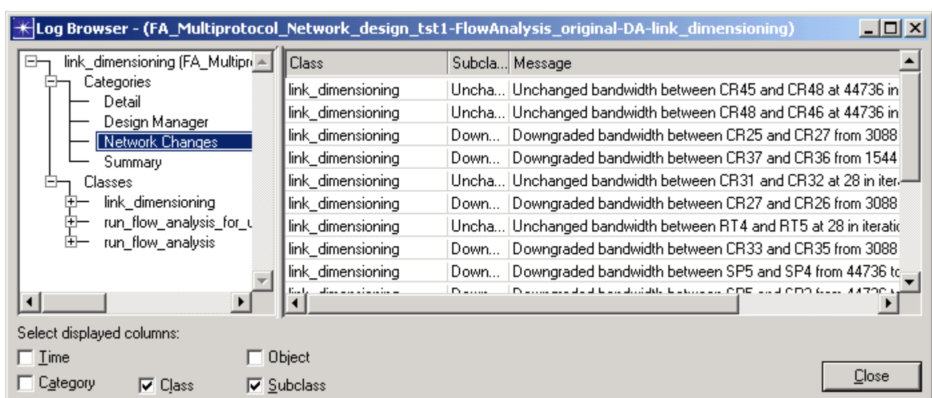


Table 1-5 lists the elements in the Design Action Log Fields.

Table 1-5 Design Action Log Fields

Label	Description
"Select columns" checkboxes	Specify which columns appear in the log table. You can specify a different set of columns for each view (for example, Categories or Classes): • Object—The network object to which the message refers • Class—Action or subaction that generated the message • Subset—Subset of the messages generated by that action or subaction
Log Treeview	Enables you to specify which messages appear the log table. If the top-level parent is selected, the table shows all messages. If a child is selected, the table shows relevant messages only.
Log Table	Lists action log messages. You can configure this table as follows: • To filter messages, set the Log Treeview. • To filter information about individual messages, set the "Select columns" dialog boxes.
End of Table 1-5	

Custom Design Actions

If you have a license for the OPNET Development Kit, you can create your own custom design models using the Design Action Editor. You can open this window in one of two ways:

- To create a new action model, open the New File dialog box (File > New) and select Design Action.
- To edit an existing action model, open the Open File dialog box (File > Open) and select Design Action.

A design action includes the following components:

- Implementation—The implementation is C or C++ code that defines the internal behavior of the model. Each base action must have an implementation for its run function. This implementation is stored within the model file. An action model can also use code defined in external C/C++ or object files. The external files may be shared among multiple action models. Derived models inherit their base models implementation and cannot define their own code.

  The OPNET Development Kit includes a set of Design Action-related functions that you can use to develop your own implementations. To view the declarations for these functions, go to the `<reldir>/sys/include` directory and look in the `odan*.h` set of header files.

- Attributes—To define the attributes for an action, choose Design Action > Edit Attributes.
- Self-Description—The self-description of a model determines how the action appears in the action treeview of the Run Design Action and Configure Design Action dialog boxes. To edit the self-description, choose Design Action > Edit Self-Description.
- Subactions—To edit the roles and subactions for a compound action, choose Design Action > Edit Subactions.

2 Link Dimensioning

The objective of a link_dimensioning run is to determine the best link configuration that accommodates the current traffic patterns in an IP network. This design action can add, remove, reconnect, or replace links, but does not change the existing network connectivity—that is, it will never remove the last link between two nodes or add a link between unconnected nodes.

A link_dimensioning run starts by analyzing the existing network and traffic patterns. Then it upgrades, downgrades, or leaves unchanged the bandwidth for each link bundle in the selection set; a *link bundle* is a set of one or more links between the same node pair. For each link bundle, the design action calculates the best solution according to the following criteria:

1) The link bundle accommodates current traffic patterns.

2) The dimensioned links are neither over-utilized nor underutilized—that is, link utilizations fall within a specified range.

3) If cost parameters are specified, the solution results in the lowest possible cost while still meeting the previous criteria.

To adjust the bandwidth of a link bundle, link_dimensioning can do any of the operations described in Table 2-1. If link_dimensioning cannot find a satisfactory solution for an individual link bundle, it generates a log message; for more information, see Log Messages on page DA-2-4.

Table 2-1 Operations on Individual Links

Operation Type	Details	Before	After
Replace	Change the data rate or model of the link, but do not change the port/interface connections to its end nodes	Link A	Link A with new model
Move	Change the data rate or model of the link, and also the port/interface connections to its end nodes This happens when the new link model cannot use the original ports.	Link A	Link A with new model and ports
Remove	Remove the link from the current scenario	Link A	None
Add	Create a new link and copy its configuration from a parallel link that existed before the current design-action run. Certain configurations such as IP address are not copied.	None	New link object with new model and ports. Configuration is based on the most recent target link.
End of Table 2-1			

Prerequisites

Like all design actions, link_dimensioning is an advanced feature.This section describes the recommended background knowledge you should have before you run this design action.

Link Dimensioning Tutorial

The Guru documentation includes a tutorial about link dimensioning. It is recommended that you complete this tutorial before you run a link_dimensioning design action on your own. For more information, see the Design Actions: Link Dimensioning tutorial.

Background Knowledge

To use this design action most effectively, you should have some familiarity with the following topics:

- Routers, ports, and their supported links
- The relationships between link utilizations and IP traffic flows
- IP routing protocols and their load-balancing options
- OPNET node and link models
- The Flow Analysis module

Design Action and Attribute Descriptions

In addition to the information in this chapter, you should also view the action description and individual attribute descriptions available in the Configure/Run Design Action dialog box. For more information, see Viewing Information about a Design Action on page DA-1-4.

Configuring and Running the link_dimensioning Design Action

This design action sizes links based on the following inputs:

- In the initial IP network:
  - — Nodes, links, and link connectivity
  - — Available ports and supported interface types on existing nodes
  - — Initial traffic levels (specified by the existing traffic flows)
  - — Initial link utilizations (determined by running Flow Analysis)

- Attribute settings in the link_dimensioning design action. Attributes of interest include:
  - — The specified target utilization range (upper bound and lower bound)
  - — The set of input links (all links in the scenario or a subset of links)
  - — The available link models (defined by the candidate link palette for the design action)
  - — Link pricing criteria (optional–specified in the link_pricer subaction)

For more information, see

- Configuring a Design Action on page DA-1-3
- Running a Design Action on page DA-1-8

Log Messages

The link_dimensioning design action generates several types of log messages to show the summary and network changes. Table 2-2 and Table 2-3 describe these log messages.

Table 2-2 Categories of link_dimensioning Log Messages

Use This Category...	To View...
Detail	Candidate link models considered when adding or replacing individual links
Design Manager	Which subactions ran successfully.
Network Changes	Which links bundles were upgraded, downgraded, and unchanged.
Warning	Warning messages generated during the run.
Summary	A list of statistics that were generated before and after the run.
End of Table 2-2	

Table 2-3 Classes of link_dimensioning Log Messages

Use This Class...	To View...
Link Pool	Candidate link models considered when adding or replacing individual links.
Unchanged	Link bundles whose bandwidths were unchanged by the design action.
Upgraded	Link bundles whose bandwidths were upgraded by the design action.
Downgraded	Link bundles whose bandwidths were downgraded by the design action.
Summary	Summary statistics about the design action
Success	Whether the design action ran successfully.
End of Table 2-3	

Unsatisfactory Solutions

In some cases, the new solution for a specific link bundle still cannot satisfy the utilization range and criteria. These cases can be categorized in two types:

- Unchanged - A link bundle is selected because it fit in selection criteria, but link dimensioning action cannot change it due to different constraints. Detailed reasons are listed in Table 2-4.

- Upgrade/downgrade but still Invalid - The new solution is not valid because no valid solution can be found within current constraints. A best invalid solution is provided and all possible reasons described in Table 2-4.

Table 2-4 Messages for Unsatisfactory Solutions

ID	Bundle status	Utilization Range	Reason	Message
0	Unchanged		No candidate links are compatible	The initial link bundle has this message as default. For bundles below range, an additional reason is that no smaller links are available (log ID #5). In this case, the initial solution uses small-capacity links and does not meet the utilization range. During the process of finding another appropriate solution, the target link bundle retains this message by default. Whenever a compatible link model is selected and tested, the design action removes this message.
1	Upgraded or unchanged	Above or within	Limited port in the first node	These three reason codes are added during the solution search process. Whenever a compatible link model is considered for inclusion in the solution, these reason codes might be added into reason code of the new solution
2	Upgraded or unchanged	Above or within	Limited port in the second node	
3	Upgraded or unchanged	Above or within	Limited by the maximum number of parallel links	
4	Upgraded or unchanged	Above	Not compliant with the utilization upper bound	This message is generated when the action finds no feasible solution under current constraints, and uses the best infeasible solution: use the largest candidate links to give the lowest maximum link utilization.
5	Downgraded or unchanged	Below	No smaller links available	This message is generated when the action finds no feasible solution under current constraints, and uses the best infeasible solution: use the smallest candidate links to give the highest minimum link utilization.
6	Upgraded or Unchanged	Above	Link Analyzer problem	Current solution should be feasible but the Link Analyzer subaction might give different results than expected. For examples that illustrate how this can occur, see the Design Actions: Link Dimensioning tutorial
7	Unchanged	Above or Below	Limited by algorithm attribute	Current solution is unchanged because the maximum number of links dimensioned per iteration is reached.
8	Warning	Any state	Unspecified reason	Unknown reason, please report to OPNET technical support.
End of Table 2-4				

Viewing Reports

After a link_dimensioning run finishes, you can view reports about the run from the View Results window (Design > Results > View Output Reports). Table 2-5 describes the available reports.

Table 2-5 link_dimensioning Reports

Report Name	Description
Link Bundle Summary	Shows the operations performed on each link bundle during the run. The Link Bundle Operation column includes links that enable you to see details about an individual operation (how a link bundle was upgraded, or why a link bundle was unchanged).
Design Action Statistics	A list of statistics that were generated during the run.
Object Tables	Intermediate statistics for individual failure cases
End of Table 2-5	

3 Link Dimensioning Resilient

The objective of a link_dimensioning_resilient run is to determine the best link configuration that will accommodate current traffic patterns in the normal case (no link failures) and in a set of defined failure cases. Like link_dimensioning, link_dimensioning_resilient can add, remove, reconnect, or replace links, but does not change the existing network connectivity—that is, it never removes the last link between two nodes or adds a link between unconnected nodes. link_dimensioning and link_dimensioning_resilient differ in the following ways:

- link_dimensioning considers the normal case (no failures) only; link_dimensioning_resilient considers the normal case and a set of failure cases, and sizes the links to accommodate traffic for all failure cases.
- link_dimensioning can upgrade or downgrade links, or leave links unchanged; link_dimensioning_resilient can upgrade links or leave them unchanged (it never downgrades links).

Prerequisites

Like all design actions, link_dimensioning_resilient is an advanced feature. Before you work with link_dimensioning_resilient, you should be familiar with the link_dimensioning design action. For more information, see the Link Dimensioning chapter:

Configuring and Running the link_dimensioning_resilient Design Action

This design action sizes links based on the following inputs:

- In the initial IP network:
  - — Nodes, links, and link connectivity
  - — Available ports and supported interface types on existing nodes
  - — Initial traffic levels (specified by the existing traffic flows)
  - — Initial link utilizations (determined by running Flow Analysis)
- Attribute settings in the link_dimensioning_resilient design action. Attributes of interest include:
  - — Specified failure cases (see Failure Cases for more information)
  - — The maximum utilization (upper bound)
  - — The set of input links (all links in the scenario or a subset of links)
  - — The available link models (defined by the candidate link palette for the design action)
  - — Link pricing criteria (optional–specified in the link_pricer subaction)

Failure Cases

A *failure case* is a scenario in which a set of one or more network objects is failed. link_dimensioning_resilient includes attributes that define the failure cases to consider when the design action is run. During a run, link_dimensioning_resilient sizes each link to accommodate current traffic levels in the normal case (no link failures) and in each possible failure case.

You can configure the following characteristics of failure scenarios:

- Failure object type—A failure case can fail links, nodes, and shared risk groups (SRGs). A shared risk group defines a failure relationship between two sets of network objects, where failures in one set cause failures in the other set. For more information, see Shared Risk Groups on page ITU-6-39 of the IT Guru *User Guide*.
- Failure object set——This is the set of all objects to be failed during the run. You can include failure cases for all eligible objects or a subset of eligible objects.
- Failure object occurrence—You can specify a run to test single failure scenarios (one object failed), pairwise scenarios (two objects failed), or simultaneous scenarios (all eligible objects failed simultaneously).

Note—A link_dimensioning_resilient run can take several minutes or more to complete, depending on the size and complexity of your network. Factors that can increase run times include: a large number of input links, a complex network with many links and node, a run that tests pairwise failures, and a large number of repetitions for each failure case.

For general information about configuring and running design actions, see

- Configuring a Design Action on page DA-1-3
- Running a Design Action on page DA-1-8

Viewing Logs

The link_dimensioning_resilient design action generates several types of log messages to show the summary and network changes. Table 3-1 and Table 3-2 describe these log messages.

Table 3-1 Categories of link_dimensioning_resilient Log Messages

Use This Category...	To View...
Detail	Intermediate statistics for individual failure cases
Design Manager	Which subactions ran successfully
Network Changes	Which links bundles were upgraded, downgraded, and unchanged
Warning	Warning messages generated during the run
Summary	A list of statistics that were generated before and after the run
End of Table 3-1	

Table 3-2 Classes of link_dimensioning_resilient Log Messages

Use This Class...	To View...
Link Pool	Candidate link models considered when adding or replacing individual links
Unchanged	Link bundles whose bandwidths were unchanged by the design action
Upgraded	Link bundles whose bandwidths were upgraded by the design action
Summary	Summary statistics about the design action
Statistics	Intermediate statistics for individual failure cases
Success	Whether the design action ran successfully
End of Table 3-2	

Unsatisfactory Solutions

When it determines an unsatisfactory solutions, the link_dimensioning_resilient design action generates messages similar to link_dimensioning; for more information, see Unsatisfactory Solutions on page DA-2-4. (Because it never downgrades links, link_dimensioning_resilient does not generate “Downgrade” messages.)

Viewing Reports

After a link_dimensioning_resilient run finishes, you can view reports about the run from the View Results window (Design > Results > View Output Reports). Table 3-3 describes the available reports.

Table 3-3 link_dimensioning_resilient Reports

Report Name	Description
Failure Cases	Provides a summary of all failure cases considered in the link_dimensioning_resilient run. If a single failure case has multiple failed objects, the failure case contains multiple rows.
Link Bundle Summary	Shows the operations performed on each link bundle during the run. The Link Bundle Operation column includes links that enable you to see details about an individual operation (how a link bundle was upgraded, or why a link bundle was unchanged).
Summary Statistics	A list of statistics that were generated before and after the run.
Intermediate Statistics	Intermediate statistics for individual failure cases
End of Table 3-3	

4 Ring Backbone Topology Design

The design actions for topology design help you determine where to place links in a network. Three design actions are available, depending on the type of network you are building. You can use the topology design actions to a create a ring-based backbone, a spanning tree, or dual spanning trees. This chapter describes the ring_backbone design action.

The topology design actions work on network topologies that contain only nodes. This type of design is also referred to as a Greenfield design. You can also use the design actions on networks that have existing links or traffic demands. In this type of incremental design, the design action can use the existing links when it determines where to place additional links.

When the design action adds links to a network, it considers the following:

- cost
- connectivity
- hop count
- utilization
- port constraints
- distance

The ring_backbone design action lets you build network designs that are resilient to failure and insensitive to traffic. These network designs provide paths that have low delays and low cost. The design action builds the network designs by constructing a near-symmetric 2-connected or 3-connected topology with a small diameter. The diameter is the maximum number of hops between any node pair.

Symmetric topologies are less sensitive to traffic fluctuations in demands. A symmetric topology can be an effective design for a backbone when you do not have any traffic information, when the traffic is very variable, or when you are planning for future applications. The design action can use existing traffic information to bias the design to have direct links between high-traffic pairs.

Prerequisites

This section describes what you need to know before you run the ring_backbone design action.

Background Knowledge

To use this design action most effectively, you should be familiar with the following topics:

- Topology design, including greenfield and brownfield design
- Routers, ports, and their supported links
- OPNET node and link models

Design Action and Attribute Descriptions

In addition to the information in this chapter, you should also view the action description and individual attribute descriptions available in the Configure/Run Design Action dialog box. For more information, see Viewing Information about a Design Action on page DA-1-4.

Configuring and Running the Ring Backbone Design Action

This design action creates links based on the following inputs:

- In the initial network:
  - — Nodes
  - — Existing links, if any
  - — Traffic demands, if any
- Attribute settings in the ring_backbone design action. Attributes of interest include:
  - — The available link models (defined by the candidate link palette for the design action)
  - — The target utilization
  - — The type of ring topology (defined by the Graph Engine attribute)
  - — Link pricing criteria (optional–specified in the link_pricer_custom_db subaction)

For more information, see

- Configuring a Design Action on page DA-1-3
- Running a Design Action on page DA-1-8

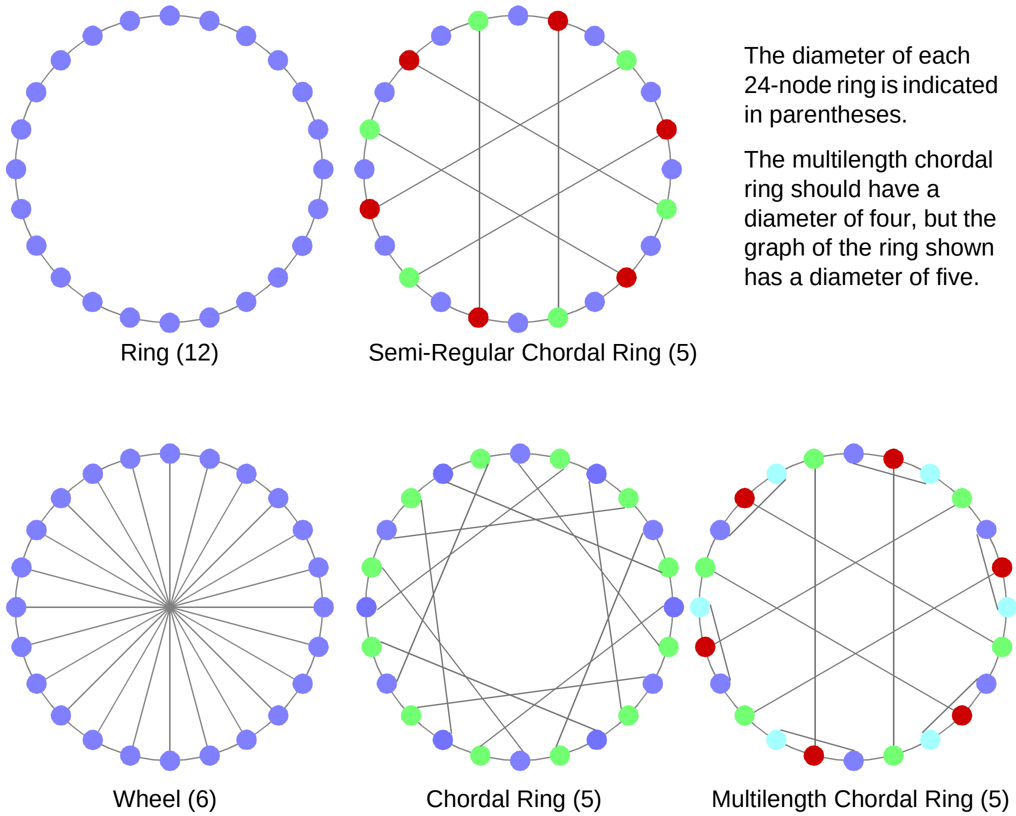
Graph Engine

The design action can create five types of ring backbone designs. All designs consists of a ring that contains all of the input nodes. The degree of connectivity within the ring differentiates ring types.

Table 4-1 Ring Backbone Graph Engine Characteristics

Graph Engine	Connectivity	Symmetry	Number of Edges	Description
Ring	2-connected	Symmetric	N	A ring that passes through each node
Semi-Regular Chordal Ring	2-connected	Near symmetric	5N/4	A ring with chords incident to every other node
Wheel	3-connected	Symmetric	3N/2	A ring with chords connecting each node to the node N/2 away in rotational distance around the ring
Chordal Ring	3-connected	Near symmetric	3N/2	A ring with chords connecting each node to the node +/- w away in rotational distance around the ring.
Multilength Chordal Ring	3-connected	Near symmetric	3N/2	Similar to the chordal ring, but with two chord lengths, w1 and w2, which minimize the diameter of the network.
End of Table 4-1				

Figure 4-1 Types of Ring Designs



The diameter of each 24-node ring is indicated in parentheses.

The multilength chordal ring should have a diameter of four, but the graph of the ring shown has a diameter of five.

Log Messages

The ring_backbone design action generates several types of log messages to show the summary and network changes. Table 4-2 and Table 4-3 list these log messages.

Table 4-2 Categories of Ring Backbone Design Log Messages

Use This Category...	To View...
Detail	Candidate link models considered when adding or replacing individual links
Design Manager	A list of subactions that ran successfully.
Network Changes	A list of the links that were created.
Error	Error messages generated during the run.
Warning	Warning messages generated during the run.
Summary	A list of statistics that were generated before and after the run. Statistics include: average cost per link, cost of new links, and the number of links added and reused.
End of Table 4-2	

Table 4-3 Classes of Ring Backbone Design Log Messages

Use This Class...	To View...
Link Pool	Candidate link models considered when adding or replacing individual links.
Filtered	Nodes that were excluded from the design. These are usually configuration objects or nodes with no available ports.
Created	Links that were added by the design action.
Reused	Links that existed in the topology and were reused by the design action.
Summary	Summary statistics about the design action.
Success	Whether or not the design action ran successfully.
End of Table 4-3	

Available Reports

Detailed results are displayed as output tables in the Results Viewer, as described in Viewing Results on page DA-1-10. The mpls_te design action generates the following reports:

- Link Summary
- Summary

Link Summary

The link summary report contains results for each link in the topology. The results given in the link summary report are described in the following table.

Table 4-4 Link Summary Report Statistics

Result	Description
Link Name	The name of the link that was added, reused, or upgraded.
Node A, Node B	The endpoints of the link.
Operation	Indicates the action taken on the link by the design action: created, reused, upgraded.
Model	The link model assigned to the link by the design action.
BW (kbps)	The bandwidth, or data rate, of the link after any modifications by the design action.
Cost	The cost of the link after any modifications by the design action. Costs are calculated using the Link Pricer subaction; by default, this is the link_pricer_custom_db action.
End of Table 4-4	

Summary

The summary report contains a general statistics about the design action run. The results given in the summary report are described in the following table.

Table 4-5 Summary Report Statistics (Part 1 of 2)

Result	Description
Vertices in Graph	The number of vertices in the graph that was used by the design action. The design action creates a graph to represent the network and uses this graph when constructing the ring design. The graph is then mapped back to the network topology. The graph contains all nodes from the input node object set, with the exception of filtered nodes.
Links in Graph	The number of links in the final (output) graph. This is the graph that results after the design algorithm has run.
Links Added	The number of links added to the topology.
Links Reused	The number of links that existed in the topology and were used by the design action.
Links Upgraded	The number of existing links that were upgraded to use a different model by the design action.
Cost of New Links	The cost of all links added to the topology.
Cost of Upgraded Links	The net cost of upgrading existing links in the topology. This is the new cost minus the existing cost for all upgraded links.
Value of Reused Links	The total cost of all reused links.
Cost of Reused Links (discounted)	The cost of reusing links in the network. If the discount factor is 100 percent, the cost is 0. Otherwise, the cost is: (1 – discount factor) * Value of Reused Links
Total Cost of Topology Change	The cost of the new, upgraded, and reused (discounted) links in the topology. The Total Cost of Topology Change equals the sum of the following statistics: • Cost of New Links • Cost of Upgraded Links • Cost of Reused Links (discounted)
Average Cost Per Link	Total cost of topology change divided by the number of added, reused, and upgraded links.
Graph Diameter	The maximum number of hops between any node pair in the graph.
Graph Average Distance	The average number of hops between any node pair in the graph (excludes 0-hop paths from source to source).

Table 4-5 Summary Report Statistics (Part 2 of 2)

Result	Description
Traffic Averaged Distance	The traffic averaged number of hops between any node pair in the graph. For each pair, the minimum hop distance is weighted by the total traffic between the node pair.
Graph Connectivity	The minimum number of node disjoint paths between all of the node pairs.
Graph Average Connectivity	The average number of node disjoint paths over all of the node pairs.
Traffic Averaged Connectivity	The connectivity for each node pair is weighted by the total traffic between the node pair.
End of Table 4-5	

5 Spanning Tree Topology Design

The design actions for topology design help you determine where to place links in a network. Three design actions are available, depending on the type of network you are building. You can use the topology design actions to a create a ring-based backbone, a spanning tree, or dual spanning trees. This chapter describes the spanning_tree design action.

The topology design actions work on network topologies that contain only nodes. This type of design is also referred to as a Greenfield design. You can also use the design actions on networks that have existing links or traffic demands. In this type of incremental design, the design action can use the existing links when it determines where to place additional links.

The spanning tree design action is useful for finding a near minimum cost topology that connects a set of nodes. This can serve as a baseline for understanding the lowest cost network option. The spanning tree may not meet the performance or redundancy requirements for your operational network design, but it may serve as a good starting point for building a final design.

When the design action adds links to a network, it considers the following requirements:

- cost
- connectivity
- hop count
- utilization
- port constraints
- distance

The spanning tree design action is very similar in functionality to the ring_backbone design action, which is described in Chapter 4 Ring Backbone Topology Design on page DA-4-1. The difference is that instead of using a ring-based graph as the network structure, the algorithm builds a spanning tree graph. A spanning tree graph is a connected, acyclic graph.

Prerequisites

This section describes what you need to know before you run the spanning_tree design action.

Background Knowledge

To use this design action most effectively, you should be familiar with the following topics:

- Topology design, including greenfield and brownfield design
- Routers, ports, and their supported links
- OPNET node and link models

Design Action and Attribute Descriptions

In addition to the information in this chapter, you should also view the action description and individual attribute descriptions available in the Configure/Run Design Action dialog box. For more information, see Viewing Information about a Design Action on page DA-1-4.

Configuring and Running the Spanning Tree Design Action

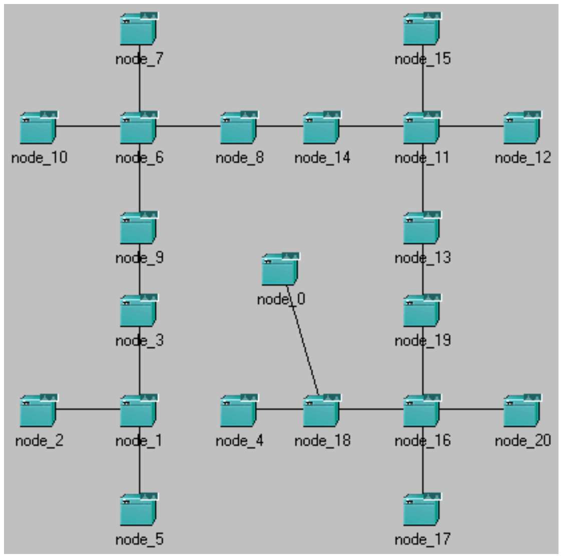
This design action creates links based on the following inputs:

- In the initial network:
  - Nodes
  - Existing links, if any
  - Traffic demands, if any
- Attribute settings in the spanning_tree design action. Attributes of interest include:
  - The available link models (defined by the candidate link palette for the design action)
  - The target utilization
  - The type of spanning tree (defined by the Spanning Tree Weight attribute)
  - The root node (defined by the Root Criteria attribute)
  - Link pricing criteria (optional–specified in the link_pricer_custom_db subaction)

For more information, see

- Configuring a Design Action on page DA-1-3
- Running a Design Action on page DA-1-8

Figure 5-1 Types of Spanning Trees



Log Messages

The spanning tree design action generates several types of log messages to show the summary and network changes. Table 5-1 and Table 5-2 list these log messages.

Table 5-1 Categories of Spanning Tree Design Action Log Messages

Use This Category...	To View...
Detail	Candidate link models considered when adding or replacing individual links
Design Manager	A list of subactions that ran successfully.
Network Changes	A list of the links that were created.
Error	Error messages generated during the run.
Warning	Warning messages generated during the run.
Summary	A list of statistics that were generated before and after the run. Statistics include: average cost per link, cost of new links, and the number of links added and reused.
End of Table 5-1	

Table 5-2 Classes of Spanning Tree Design Action Log Messages

Use This Class...	To View...
Link Pool	Candidate link models considered when adding or replacing individual links.
Filtered	Nodes that were excluded from the design. These are usually configuration objects or nodes with no available ports.
Created	Links that were added by the design action.
Reused	Links that existed in the topology and were reused by the design action.
Summary	Summary statistics about the design action.
Success	Whether or not the design action ran successfully.
End of Table 5-2	

Available Reports

Detailed results are displayed as output tables in the Results Viewer, as described in Viewing Results on page DA-1-10. The mpls_te design action generates the following reports:

- Link Summary
- Summary

Link Summary

The link summary report contains results for each link in the topology. The results given in the link summary report are described in the following table.

Table 5-3 Link Summary Report Statistics

Result	**Description**
Link Name	The name of the link that was added, reused, or upgraded.
Node A, Node B	The endpoints of the link.
Operation	Indicates the action taken on the link by the design action: created, reused, upgraded.
Model	The link model assigned to the link by the design action.
BW (kbps)	The bandwidth, or data rate, of the link after any modifications by the design action.
Cost	The cost of the link after any modifications by the design action. Costs are calculated using the Link Pricer subaction; by default, this is the link_pricer_custom_db action.
End of Table 5-3	

Summary

The summary report contains a general statistics about the design action run. The results given in the summary report are described in the following table.

Table 5-4 Summary Report Statistics (Part 1 of 2)

Result	Description
Vertices in Graph	The number of vertices in the graph that was used by the design action. The design action creates a graph to represent the network and uses this graph when constructing the tree. The graph is then mapped back to the network topology. The graph contains all nodes from the input node object set, with the exception of filtered nodes.
Links in Graph	The number of links in the final (output) graph. This is the graph that results after the design algorithm has run.
Links Added	The number of links added to the topology.
Links Reused	The number of links that existed in the topology and were used by the design action.
Links Upgraded	The number of existing links that were upgraded to use a different model by the design action.
Cost of New Links	The cost of all links added to the topology.
Cost of Upgraded Links	The net cost of upgrading existing links in the topology. This is the new cost minus the existing cost for all upgraded links.
Value of Reused Links	The total cost of all reused links.
Cost of Reused Links (discounted)	The cost of reusing links in the network. If the discount factor is 100 percent, the cost is 0. Otherwise, the cost is: (1 – discount factor) * Value of Reused Links
Total Cost of Topology Change	The cost of the new, upgraded, and reused (discounted) links in the topology. The Total Cost of Topology Change equals the sum of the following statistics: • Cost of New Links • Cost of Upgraded Links • Cost of Reused Links (discounted)
Average Cost Per Link	Total cost of topology change divided by the number of added, reused, and upgraded links.
Graph Diameter	The maximum number of hops between any node pair in the graph.
Graph Average Distance	The average number of hops between any node pair in the graph (excludes 0-hop paths from source to source).

Table 5-4 Summary Report Statistics (Part 2 of 2)

Result	Description
Traffic Averaged Distance	The traffic averaged number of hops between any node pair in the graph. For each pair, the minimum hop distance is weighted by the total traffic between the node pair.
Graph Connectivity	The minimum number of node disjoint paths between all of the node pairs.
Graph Average Connectivity	The average number of node disjoint paths over all of the node pairs.
Traffic Averaged Connectivity	The connectivity for each node pair is weighted by the total traffic between the node pair.
End of Table 5-4	

6 Dual Tree Topology Design

The design actions for topology design help you determine where to place links in a network. Three design actions are available, depending on the type of network you are building. You can use the topology design actions to a create a ring-based backbone, a spanning tree, or dual spanning trees. This chapter describes the dual_tree design action.

The topology design actions work on network topologies that contain only nodes. This type of design is also referred to as a Greenfield design. You can also use the design actions on networks that have existing links or traffic demands. In this type of incremental design, the design action can use the existing links when it determines where to place additional links.

The dual tree design action is useful for finding a near minimum cost two-connected design for a given set of nodes. It builds on the low cost of the spanning tree design, but guarantees that there are at least 2 disjoint paths between all node pairs. The diameter and average hop distance are also generally significantly lower than the spanning tree.

When the design action adds links to a network, it considers the following requirements:

- cost
- connectivity
- hop count
- utilization
- port constraints
- distance

The dual tree design action is based on a spanning tree design. After constructing the first spanning tree, the design builds a second spanning tree by connecting the leaf nodes from the first tree. This creates a near-minimum cost 2-connected topology.

Prerequisites

This section describes what you need to know before you run the dual_tree design action.

Background Knowledge

To use this design action most effectively, you should be familiar with the following topics:

- Topology design, including greenfield and brownfield design
- Routers, ports, and their supported links
- OPNET node and link models

Design Action and Attribute Descriptions

In addition to the information in this chapter, you should also view the action description and individual attribute descriptions available in the Configure/Run Design Action dialog box. For more information, see Viewing Information about a Design Action on page DA-1-4.

Configuring and Running the Dual Tree Design Action

This design action creates links based on the following inputs:

- In the initial network:
  - Nodes
  - Existing links, if any
  - Traffic demands, if any
- Attribute settings in the dual_tree design action. Attributes of interest include:
  - The available link models (defined by the candidate link palette for the design action)
  - The target utilization
  - The types of spanning trees (defined by the Spanning Tree Weight attribute for the first tree and the Dual Tree Weight attribute for the second tree)
  - The root node (defined by the Root Criteria attribute)
  - Link pricing criteria (optional–specified in the link_pricer_custom_db subaction)

For more information, see

- Configuring a Design Action on page DA-1-3
- Running a Design Action on page DA-1-8

Log Messages

The dual tree design action generates several types of log messages to show the summary and network changes. Table 6-1 and Table 6-2 list these log messages.

Table 6-1 Categories of Dual Tree Design Action Log Messages

Use This Category...	To View...
Detail	Candidate link models considered when adding or replacing individual links
Design Manager	A list of subactions that ran successfully.
Network Changes	A list of the links that were created.
Error	Error messages generated during the run.
Warning	Warning messages generated during the run.
Summary	A list of statistics that were generated before and after the run. Statistics include: average cost per link, cost of new links, and the number of links added and reused.
End of Table 6-1	

Table 6-2 Classes of Dual Tree Design Action Log Messages

Use This Class...	To View...
Link Pool	Candidate link models considered when adding or replacing individual links.
Filtered	Nodes that were excluded from the design. These are usually configuration objects or nodes with no available ports.
Created	Links that were added by the design action.
Reused	Links that existed in the topology and were reused by the design action.
Summary	Summary statistics about the design action.
Success	Whether or not the design action ran successfully.
End of Table 6-2	

Available Reports

Detailed results are displayed as output tables in the Results Viewer, as described in Viewing Results on page DA-1-10. The mpls_te design action generates the following reports:

- Link Summary
- Summary

Link Summary

The link summary report contains results for each link in the topology. The results given in the link summary report are described in the following table.

Table 6-3 Link Summary Report Statistics

Result	**Description**
Link Name	The name of the link that was added, reused, or upgraded.
Node A, Node B	The endpoints of the link.
Operation	Indicates the action taken on the link by the design action: created, reused, upgraded.
Model	The link model assigned to the link by the design action.
BW (kbps)	The bandwidth, or data rate, of the link after any modifications by the design action.
Cost	The cost of the link after any modifications by the design action. Costs are calculated using the Link Pricer subaction; by default, this is the link_pricer_custom_db action.
End of Table 6-3	

Summary

The summary report contains a general statistics about the design action run. The results given in the summary report are described in the following table.

Table 6-4 Summary Report Statistics (Part 1 of 2)

Result	Description
Vertices in Graph	The number of vertices in the graph that was used by the design action. The design action creates a graph to represent the network and uses this graph when constructing the tree. The graph is then mapped back to the network topology. The graph contains all nodes from the input node object set, with the exception of filtered nodes.
Links in Graph	The number of links in the final (output) graph. This is the graph that results after the design algorithm has run.
Links Added	The number of links added to the topology.
Links Reused	The number of links that existed in the topology and were used by the design action.
Links Upgraded	The number of existing links that were upgraded to use a different model by the design action.
Cost of New Links	The cost of all links added to the topology.
Cost of Upgraded Links	The net cost of upgrading existing links in the topology. This is the new cost minus the existing cost for all upgraded links.
Value of Reused Links	The total cost of all reused links.
Cost of Reused Links (discounted)	The cost of reusing links in the network. If the discount factor is 100 percent, the cost is 0. Otherwise, the cost is: (1 – discount factor) * Value of Reused Links
Total Cost of Topology Change	The cost of the new, upgraded, and reused (discounted) links in the topology. The Total Cost of Topology Change equals the sum of the following statistics: • Cost of New Links • Cost of Upgraded Links • Cost of Reused Links (discounted)
Average Cost Per Link	Total cost of topology change divided by the number of added, reused, and upgraded links.
Graph Diameter	The maximum number of hops between any node pair in the graph.
Graph Average Distance	The average number of hops between any node pair in the graph (excludes 0-hop paths from source to source).

Table 6-4 Summary Report Statistics (Part 2 of 2)

Result	Description
Traffic Averaged Distance	The traffic averaged number of hops between any node pair in the graph. For each pair, the minimum hop distance is weighted by the total traffic between the node pair.
Graph Connectivity	The minimum number of node disjoint paths between all of the node pairs.
Graph Average Connectivity	The average number of node disjoint paths over all of the node pairs.
Traffic Averaged Connectivity	The connectivity for each node pair is weighted by the total traffic between the node pair.
End of Table 6-4	

Index