Network Simulation

7.7.1

Network Simulation and VIRL

Network simulation provides a means to test network configurations, debug configuration code, and to work with and learn Cisco infrastructure and APIs in a safe, convenient, and non-cost-prohibitive way.

Cisco Virtual Internet Routing Laboratory (VIRL, pronounced 'viral') is a commercial product originally developed for internal use at Cisco, with broad and active community support. Now in version 2, VIRL can run on bare metal, or on large virtual machines on several hypervisor platforms (ESXi and VMware Workstation 12+ among them). The official name for VIRL v 2.0 is Cisco Modeling Labs – Personal (or CML-Personal). You may find the tool referenced as both VIRL 2.0 and CML - P.

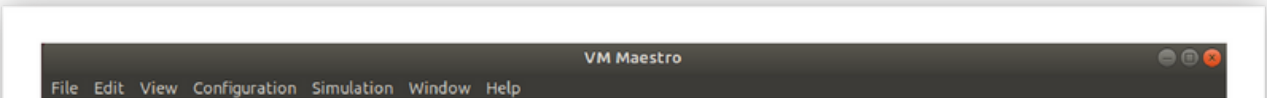
Though VIRL cannot duplicate the performance of elite hardware or SDN software components running in optimal production conditions, it mirrors Cisco functionality perfectly. The virtual equipment that runs inside VIRL uses the same code that runs inside actual Cisco products. This makes VIRL an ideal tool for learning, as well as a useful mechanism for trialing network configurations and fine-tuning automation for building and testing them.

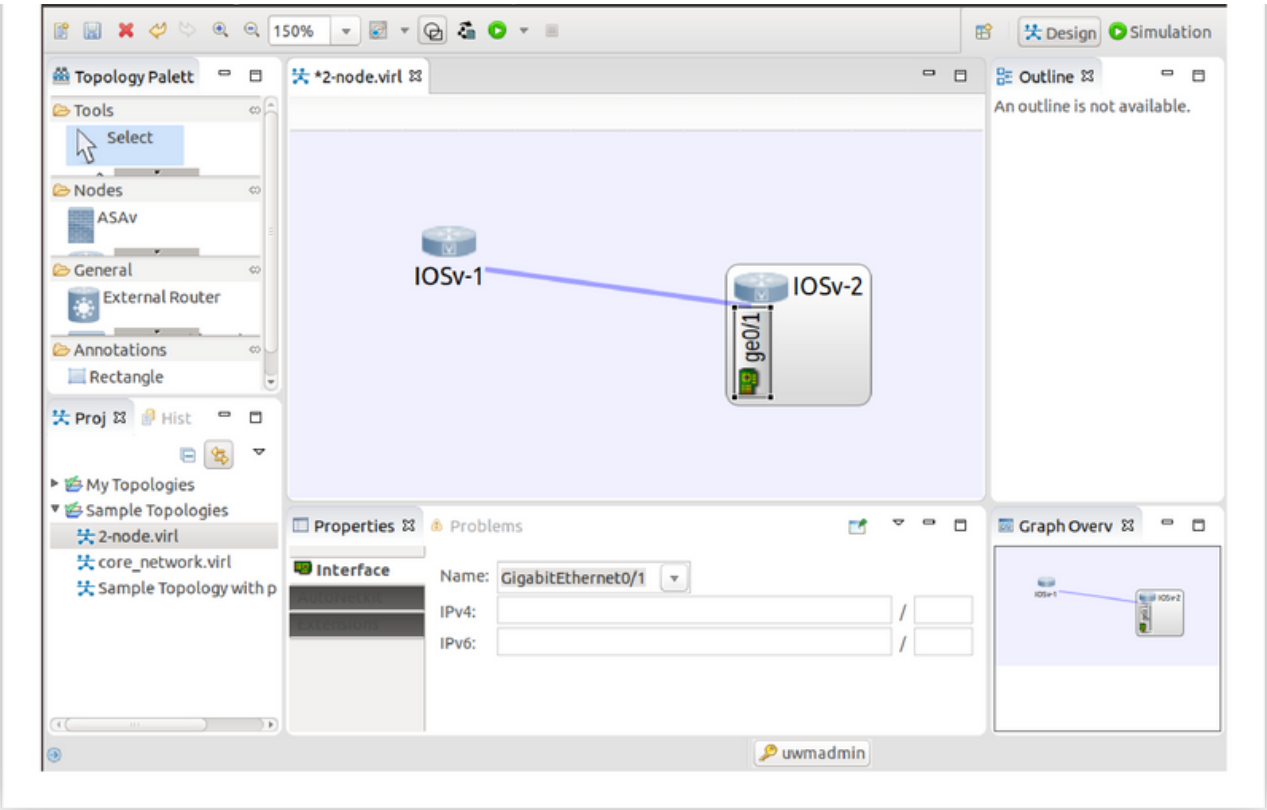
VIRL components and workflow

VIRL provides a local CLI for system management, a REST interface for integration with automation, and a powerful UI that offers a complete graphical environment for building and configuring simulation topologies.

The UI comes with several sample topologies to get you started. Among these is a two-router IOS network simulation that can quickly be made active and explored. VIRL's **Design Perspective** view lets you modify existing simulations (after stopping them) or compose new simulations by dragging, dropping, and connecting network entities, configuring them as you go.

VIRL Displays a Visualization of Your Simulation





The visualization has clickable elements that let you explore configuration of entities and make changes via the WebUI, or by connecting to network elements via console. You can also extract individual device configurations, or entire simulated network configs, as `.virl` files.

VIRL files

VIRL also enables you to define simulations as code, enabling both-ways integration with other software platforms for network management and testing.

VIRL's native configuration format is called a `.virl` file, which is a human-readable YAML file. The `.virl` file contains complete descriptions of the IOS routers, their interface configurations and connection (plus other configuration information), credentials for accessing them, and other details. These files can be used to launch simulations via the VIRL REST API, and you can convert `.virl` files to and from "testbed" files for use with PyATS and Genie.

In the VIRL UI, you select a simulation, make VIRL read the device's configuration, and then it composes a `.virl` file to represent it. VIRL offers to save the topology in a new file that you can then open in an editor for review.

The `.virl` file provides a method for determining if configuration drift has occurred on the simulation. A simple `diff` command can compare a newly-extracted `.virl` file with the original `.virl` file used to launch the simulation, and differences will be apparent.

This technique, comparing a known-good configuration manifest with an extracted manifest describing current network state, helps debug real-world networks for which authoritative, complete PyATS

topologies are available.

7.6
Automating Testing

7.8
Infrastructure and Automation Summary