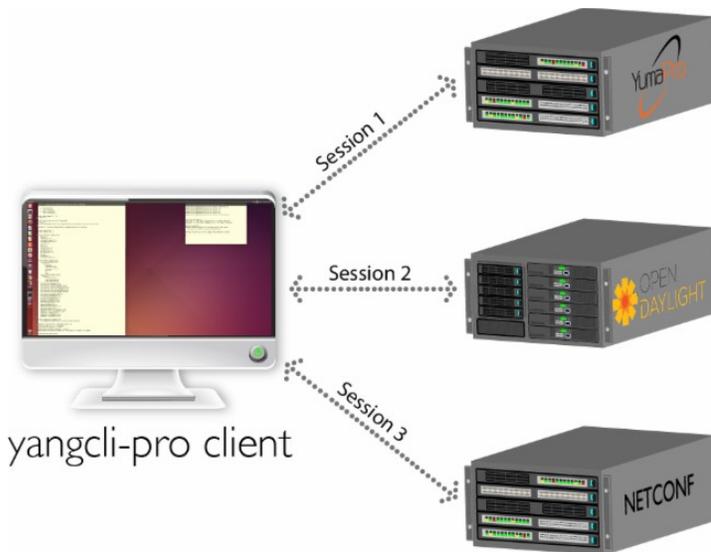**Multi-session, easy to use, and powerful NETCONF client with scripting**

Managed NETCONF sessions

yangcli-pro manages:

- **YumaWork's netconfd-pro multi-protocol server**
- **OpenDaylight**
- **Standard NETCONF servers**

yangcli-pro is designed to help automate many of your development and management tasks for NETCONF servers with automatic:

- **mirror of running config**
- **config-change handling**
- **notification setup**
- **transaction and lock management**

Scripting and Test Suites

yangcli-pro provides powerful feratures to develop and manage NETCONF servers:

- **Creation and playback of scripts**
- **Record session commands and responses and replay to different servers reporting if the test replay passes or fails**
- **Test suites can work for both positive and negative tests**
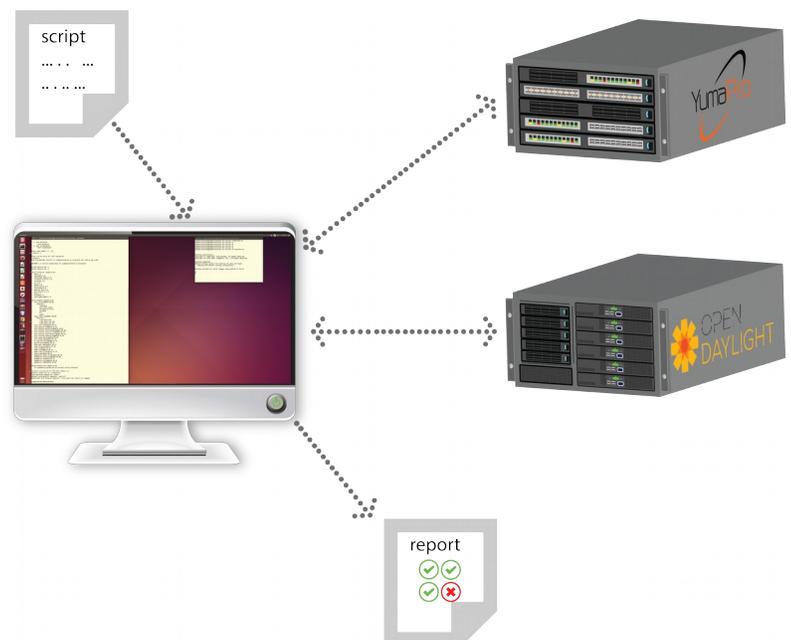
yangcli-pro is a full feature NETCONF client:

NETCONF Sessions

Concurrent Sessions	Multiple NETCONF sessions can be active at once, allowing interactive, user script, and test suite access to multiple servers at once.
Named Saved Sessions	Session configuration can be saved as named sessions. These saved sessions can be loaded manually or automatically, greatly simplifying the **connect** command.
YANG-Driven Command Interpreter	Any YANG module can be loaded into **yangcli-pro** and used right away. All **rpc** statements are automatically available as commands. All **data** statements are automatically available as database content or monitoring data.

Automation

Compile YANG Modules on the Fly	When a NETCONF session is started, the server <capability> statements can be used to compile the YANG modules exactly as that server is using them, so **yangcli-pro** can tailor the user interface independently for each session.
Configuration Shadowing	When a NETCONF session is started, a copy of the server's running configuration can be automatically cached for CLI tab completion and other operations.
Notification Monitoring	When a NETCONF session is started, notification monitoring can be started automatically. Various notifications are monitored, such as configuration change events to force shadow configuration caches to be properly updated.
Lock Handling	The high-level **get-locks** and **release-locks** commands allow all configuration locks for a NETCONF session to be managed automatically.

Programmability

Aliases	Unix-style command aliases allow user command short-cuts to be defined to save typing.
Data Variables	Powerful system variables can be used to change CLI parameters at run-time and user variables can be used to save almost any kind of data for reuse in commands, scripts, and tests.
File Variables	Data can be read and stored in XML or text files instead of data variables, allowing off-line tools to generate data for tests.
Scripts	Commands can be stored in text files and run automatically as scripts. Parameters can be passed to scripts. Scripts can call other scripts. Aliases, data variables and file variables can be used.
XPath based conditional statements	The **if, while, and eval** commands use XPath to allow customized conditional commands to be programmed into scripts. All data variables are available as XPath variables in these expressions.

Regression Testing

Test suites	The **test-suite** feature allows text test-suite configuration files to be used to control automated server regression testing. A test suite contains 0 or more tests with 0 or more steps. Test dependencies can be defined to prune tests that cannot be run on the test server.
Negative Testing	If a test step sends an operation to a NETCONF server, any type of reply can be expected, including **<rpc-error>**. Specific error fields can also be checked.
Data Testing	If a test step sends an operation to a NETCONF server, a data reply can be expected. These data replies can be checked against a saved control response.
Multiple Sessions	Tests can contain commands for multiple sessions to test locking scenarios, editing collisions, etc.
Test Reporting	Formatted test results can be generated and saved to a special log file.

Command Line Interface

YANG-based Help	The **help** command provides context-sensitive, user-friendly help text derived directly from YANG modules.
Smart Tab Completion	Pressing the tab key during interactive command line editing will cause a context-sensitive list of word completions to be displayed. List key values are fetched from the shadow configuration if available. Variable names and many other new tab completion features are supported.
Order Independent Parameters	Command and data parameters can be entered in any order. However, YANG lists and leaf-lists that are **ordered-by user** should be entered in the desired order.
Configuration Mode Editing	**Configuration mode** similar to router CLI interfaces, allowing configuration data to be edited directly.
Automatic Parameter Checking	Commands and data that are entered in interactive mode will be checked for any missing mandatory parameters. If so, the user will be prompted to fill in any missing parameter values, 1 at a time.
Command Validation	Any conditional YANG constructs like **if-feature** or **when** statements are automatically checked when a command is invoked so unsupported fields are not shown to the user.