

Easy configuration of NETCONF devices

David Alexa¹
FIT, CTU in Prague
Czech Republic

Tomas Cejka²
CESNET, a.l.e.
Czech Republic

alexadav@fit.cvut.cz

cejkat@cesnet.cz

Abstract. It is necessary for developers of devices or systems to supply a user interface that can be used for control and monitoring. Visualisation of device's configuration and state data belongs to non-trivial tasks as well as preparation of easy mechanism of configuration for end users. This paper is focused on universal graphical user interface for NETCONF protocol *NetopeerGUI* that is developed as an open-source project. *NetopeerGUI* is based on usage of standard technologies such as configuration protocol NETCONF and modeling language Yang, both standardized by IETF. *NetopeerGUI* is a NETCONF client that can be easily used as a user interface for configuration and control of any device supporting NETCONF protocol. *NetopeerGUI* provides basic universal way of data presentation that helps developers to concentrate on device development. This paper proposes *NetopeerGUI* as an interface that can be deployed on a system to supply remote configuration and monitoring through the web browser and that can increase the speed of development process.

Keywords. NetopeerGUI, NETCONF client, YANG, Graphical user interface, Netopeer, Libnetconf

1 Introduction

This paper is focused on a universal graphical user interface for network devices configuration called *NetopeerGUI*. *NetopeerGUI* is based on usage of standard technologies such as configuration protocol NETCONF and modeling language Yang, both standardized by IETF. *NetopeerGUI* works as a NETCONF client in web browser.

NetopeerGUI supplies a unified method of configuration of various remote devices (supporting NETCONF protocol) or configuration of one specific (local) device. Both kinds of configuration are tightly related, but each fits to different use case. *NetopeerGUI* helps users to configure any device without need to know all internal NETCONF commands and even without deep knowledge of device's configuration data model. It transforms responses from NETCONF server into user-friendly look and simplifies whole process of configuration.

The Network Configuration Protocol (NETCONF) protocol is proposed in RFC6241 [6]. NETCONF defines a mechanism through which a network device can be managed and configured. NETCONF specifies the way of configuration data retrieval and modification. The protocol allows the device to expose a full, formal application programming interface (API). Applications can use NETCONF to send and receive full and partial configuration data.

Yang modeling language is proposed in RFC6020 [2]. YANG is used to model the operations and content layers of NETCONF. It models configuration and state data of a device. The

data model describes data that can be manipulated by NETCONF. Additionally, Yang model can define remote procedure calls, and NETCONF notifications for advanced communication between NETCONF server and client.

2 Architecture of NETCONF-enabled device

Architecture of NETCONF-enabled device is shown in Fig. 1. The NETCONF protocol [6] “uses a remote procedure call (RPC) paradigm. A client encodes an RPC in XML and sends it to a server using a secure, connection-oriented session. The server responds with a reply encoded in XML.” The example of a command line interface NETCONF client is *netopeer-cli* [4]. The *netopeer-cli* utility is controlled via text input so that users need to write content of configuration manually in XML format.

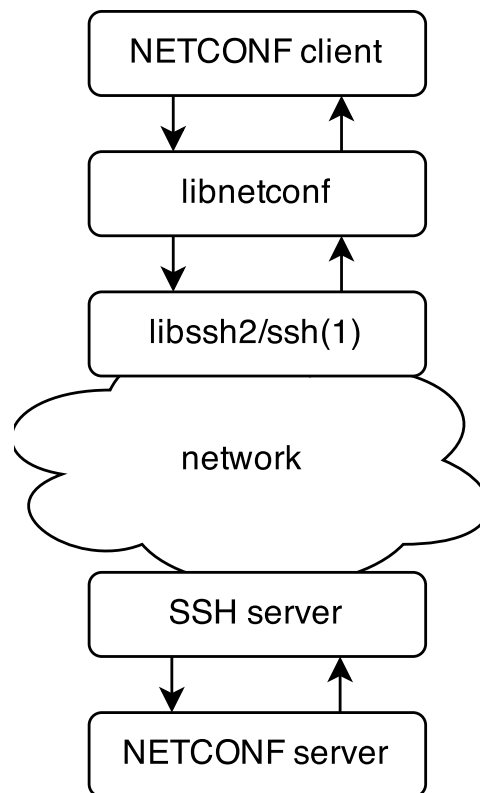


Figure 1: General architecture of NETCONF-enabled device. NETCONF client and NETCONF server (=configurable device) communicates together using *libnetconf* [5].

NetopeerGUI behaves like NETCONF client (described in Fig. 1). In *NetopeerGUI*, we focus on configuration data visualisation and representation, using data models, bringing usable user interface. *NetopeerGUI* helps users to configure any device without need to know all NETCONF internal commands, their format and behaviour. It generates all requests internally and presents all information (responses from NETCONF server) and guidance of user by context help gained from model in user-friendly look.

NetopeerGUI uses data models for adding semantic information to response from NETCONF server, to every node described in data model. These information are presented in GUI automatically in well-arranged form.

The *NetopeerGUI* project contains Apache module called *mod_netconf* [3] as a backend. *NetopeerGUI* is an PHP application based on Symfony2 framework. *NetopeerGUI* communicates via *UNIX* socket with backend that persists opened NETCONF sessions. Schema of this communication is described in Fig. 2.

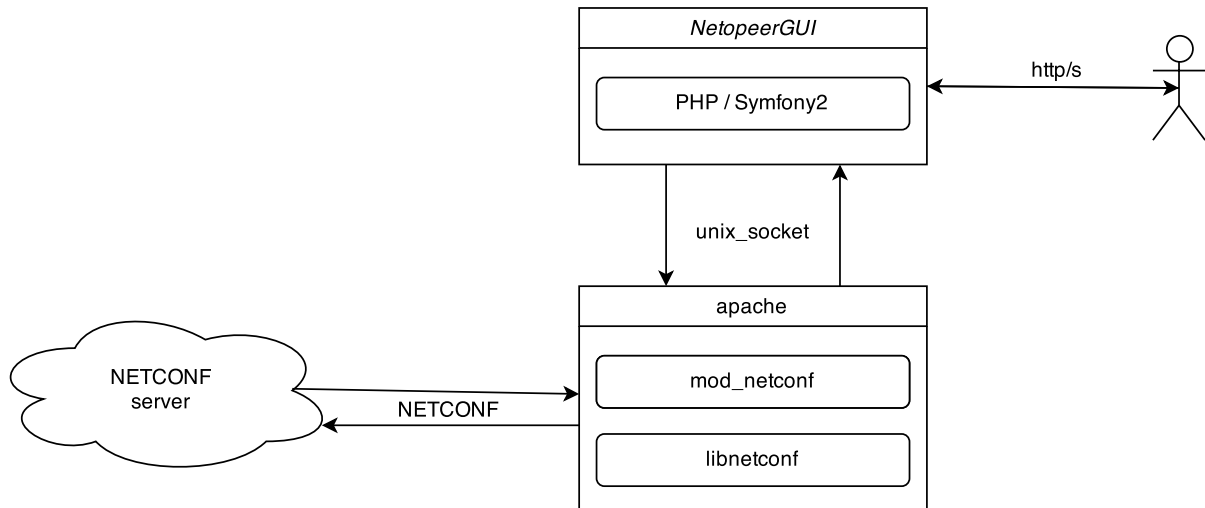


Figure 2: Architecture of *NetopeerGUI*, which communicates with *mod_netconf* (=Apache module) using *unix_socket*.

2.1 Related Work

NetopeerGUI is not the only one NETCONF client. List of known NETCONF implementations is shown in [1]. We will consider only NETCONF clients for comparison with our work.

List of NETCONF clients using CLI

- YUMAPro ¹
- NuDesign ²
- Netopeer ³

List of NETCONF clients using GUI

- MG-SOFT NETCONF Browser Pro ⁴
- NETCONFc ⁵

NetopeerGUI differs from other solutions mainly in architectural approach. Related projects are developed as desktop applications whereas *NetopeerGUI* is web application. In addition,

¹<http://www.yumaworks.com>

²<http://www.ndt-inc.com/>

³<https://www.liberouter.org/technologies/netconf/>

⁴<http://www.mg-soft.com/mgNetConfBrowser.html>

⁵<http://www.seguesoft.com>

NetopeerGUI is deployable on standalone device in order to create independent system that contains graphical user interface for remote configuration.

Configuration data models in Yang are usually used for validation of configuration data. Models can be also used as a source of help or information about configured device. In comparison to other related work, *NetopeerGUI* uses configuration data models to generate visually represented configuration data. Models are used for automatic generation of control elements for creation of new configuration.

3 NetopeerGUI — Interface for any Device

NetopeerGUI can communicate with any device, which supports NETCONF protocol. It could be for example Toaster [7] or module for linux system configuration. *NetopeerGUI* unifies the look, how to present configuration information in the universal way. Users do not have to write any XML configuration by hand, but they can create it by GUI.

***NetopeerGUI* offers these benefits:**

- users do not have to know all NETCONF specific commands, *NetopeerGUI* handles all commands internally in the user-friendly form,
- configuration of devices is transformed from XML into HTML form elements,
- easy configuration using GUI instead of using command line,
- no manual writing of configuration,
- configuration of one specific device (defined in application config file) vs. configuration of unlimited number of devices,
- CESNET offers all necessary parts needed to work with NETCONF protocol (more info in [5]),
- Netopeer-Demo — demo distribution as an virtual machine image with all necessary parts for using NETCONF.

***NetopeerGUI* brings these features:**

- history of connected devices,
- support of `<get-schema>`,
- schema of data model visible for every configuration,
- full support of CRUD operations on configuration,
- support for user-ordered lists,
- automatic filtering into modules,
- help and other useful semantic information for each node (from processed data model),
- automatic completion of available children nodes,
- asynchronous NETCONF notifications,

- calling RPC methods,
- full modular architecture (you can define own look for specific module = modular extensibility).

NetopeerGUI supports two application modes. The first one is *NetopeerGUI* as a service, which brings possibility of connecting more devices (behaves like a hub). The second mode offers so called “single instance” — applications automatically connects to one, predefined device. It is useful for configuration of only one devices, i. e. GUI for linux system configuration or universal GUI for home router configuration.

Difference between these two modes is described in Fig. 3.

3.1 Modular extensibility

NetopeerGUI behaves like a wrapper for communication with NETCONF server. It provides API for calling methods in *mod_netconf* and easy way how to get configuration data into templates. Every developer can provide own look of specific module configuration. There is no need to use only default, universal, look.

Every extension is developed as Symfony2 Bundle. Extension could be installed by composer [8] in few easy steps. Developers can override or extend default look to fit their individual needs too.

NetopeerGUI provides functionality for switching between pages (modules) using asynchronous walk-through. This asynchronous walk-through is prepared as a separate jQuery plugin, so everybody can use it in own extensions.

NetopeerGUI contains some useful services (using Dependency Injection [9]). The most important is service called *DataModel*, which provides methods for calling *mod_netconf* API. Through this service, all necessary request such as `<get>`, `<get-config>`, `<edit-config>` and so could be called and processed in own extension.

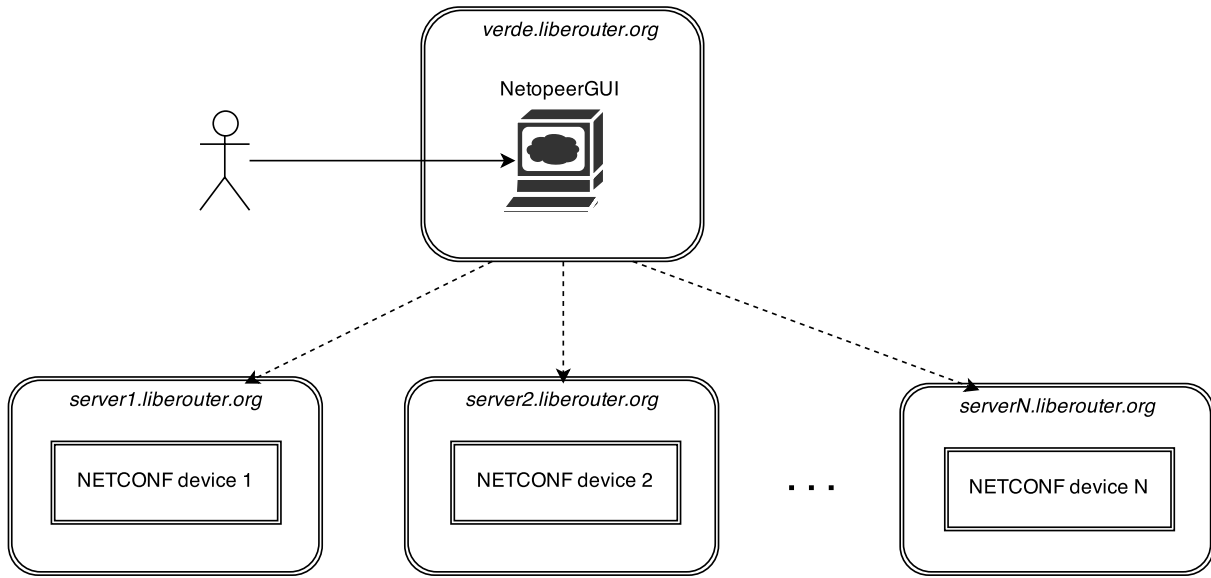
4 Conclusion

NetopeerGUI brings universal graphical user interface for configuration of NETCONF devices, both remotely or locally. It can operate in two different modes — “as a service” or in “single instance” mode.

It simplifies the whole process of configuration by intuitive and user-friendly GUI. Users do not have to know all internal commands of NETCONF protocol. They use web forms and actions like they are used to from other websites. In addition, *NetopeerGUI* can be used as a user interface for new devices or systems that are developed. That can increase the speed of development.

Acknowledgments

This work was supported by the “CESNET Large Infrastructure” (LM2010005) funded by the Ministry of Education, Youth and Sports of the Czech Republic, MOBILITY 7AMB14SK177 Verification and dependability of digital systems design and by CTU in Prague that funded the grant No. SGS15/122/OHK3/1T/18.



*NetopeerGUI as a service vs.
GUI on device -- single instance mode*

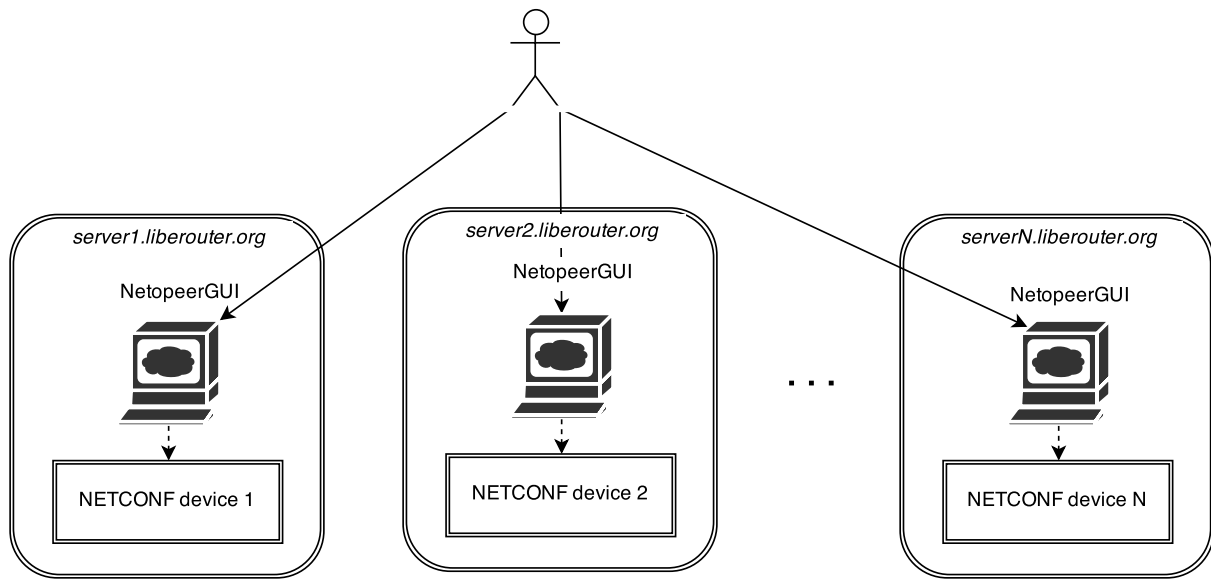


Figure 3: Modes of *NetopeerGUI*. *NetopeerGUI* can work in two ways — as a service for configuration of any devices or in single instance mode, when the *NetopeerGUI* is part of configurable device.

References

- [1] The NETCONF Wiki, <http://trac.tools.ietf.org/wg/netconf/trac/wiki>
- [2] Bjorklund, M.: YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF). RFC 6020 (Proposed Standard) (Oct 2010)
- [3] CESNET, a.l.e.: Mod_netconf, https://github.com/CESNET/mod_netconf

- [4] CESNET, a.l.e.: Netopeer, <https://code.google.com/p/netopeer/>
- [5] CESNET, a.l.e.: libnetconf - the NETCONF library in C (2012), <https://code.google.com/p/libnetconf/>
- [6] Enns, R., Bjorklund, M., Schoenwaelder, J., Bierman, A.: Network Configuration Protocol (NETCONF). RFC 6241 (Proposed Standard) (Jun 2011)
- [7] Toaster, <http://www.netconfcentral.org/modulereport/toaster>
- [8] Nils Adermann, Jordi Boggiano: Composer, <https://getcomposer.org>
- [9] The DependencyInjection Component, http://symfony.com/doc/current/components/dependency_injection/introduction.html