

# SOFTWARE SYSTEM FOR CONFIGURATION MANAGEMENT OF NETWORKING DEVICES

Delyan Genkov<sup>1</sup>

<sup>1</sup>Technical University of Gabrovo, Department of Computer Systems and Technologies, Gabrovo, Bulgaria

## Abstract

Configuration files are critical for most networking devices – without configuration they are not functioning at all, or can't perform some essential functions. There is reach variety of networking devices, and different approaches for configuration management. Present paper aims to propose a software application for centralized configuration management of some networking devices, which can assist network administrators in backup and restore tasks for their networks.

**Keywords:** Network device, Configuration, Management, Backup, Restore, Software

-----\*\*\*-----

## 1. INTRODUCTION

Configurations are important part for networking devices functionality. Without configuration some network devices can't perform some or all of their functions. For example Cisco Systems routers without initial configuration even don't have IP addresses and cannot route. Most of the home appliances have default configuration enough to make a connection to Internet, but have no security for their wireless network.

Distinct vendors have different approaches for configuration files management for their devices. Some of them, including Cisco Systems, Juniper and Hewlett Packard store the configuration as a text file, containing set of commands and residing in non-volatile memory (NVRAM). During power-on the device reads this file loading it into RAM memory, then executes the commands and become to perform functionality, defined by the administrator. Some cheaper devices like Belkin and Netgear store configuration into binary instead of text file [1]. Some vendors, like TP-Link combines both approaches in their devices – for example their home routers like TL-WR1043ND use binary file, but some manageable switches like TL-SG3210 use text file with commands.

Maintaining actual copies of the configuration files is an important task, especially in organizations that don't have a skilled network administrator employed. It is also necessary in training network laboratories, where is presumed that configurations may be changed regularly for every different lab setup.

Normal way to save configuration of network devices suggests manual connection to every single device and saving the configuration manually to external server, executing a command or pressing a button. Data exchange with the server is some file-oriented protocol, as FTP, TFTP, RCP, HTTP or the encrypted versions SFTP, RCP and HTTPS. This approach is distributed and stores every device's configuration separately. When many devices in a

network exist, this process can be time and effort-consuming. Sometimes when many different configurations for every device have to be maintained, for example in a networking laboratory with different scenarios and setups, this approach can be difficult to use and can easy lead to misconfiguration of devices.

There are some vendor-specific approaches for configuration management, for example Cisco LAN Management Solution, SolarWinds and HP OpenView are complex and expensive network management platforms, which can perform many tasks and manage device configurations. There are also some great free tools, like rConfig [2] and Rancid [3], which are mainly Cisco and Juniper-oriented, and offers only telnet and SSH connections.

The goal of this work is to describe a software system for configuration management of different network devices from different vendors with user profiles and extendable functions.

## 2. SYSTEM ARCHITECTURE

Architecture and main modules of the system are shown at Fig. 1.

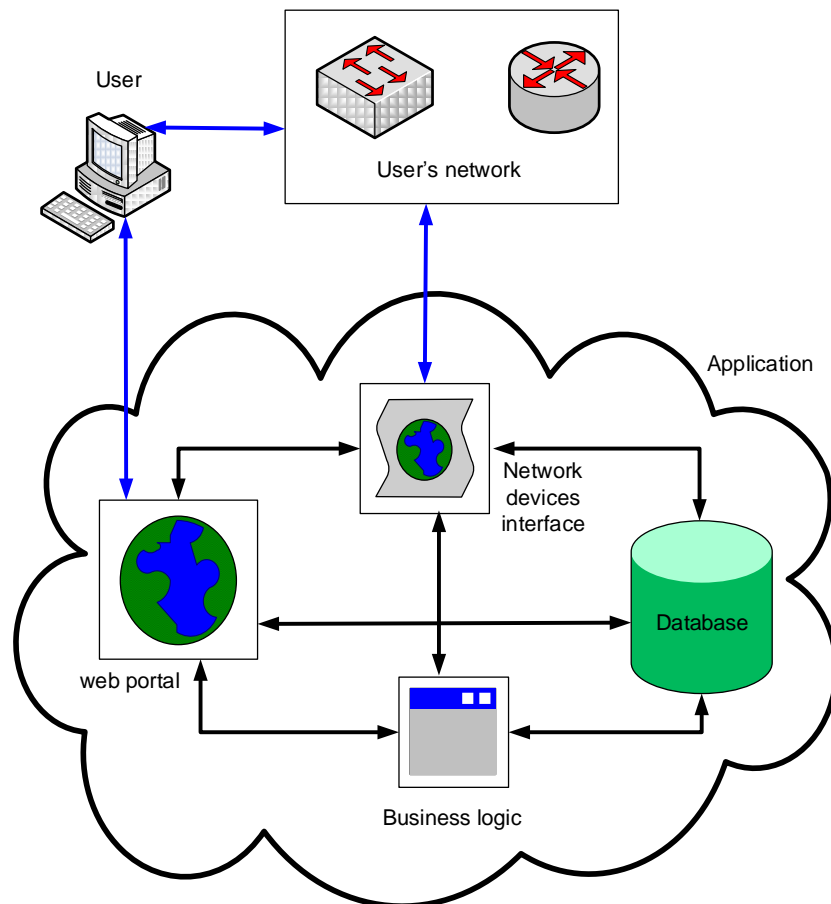


Fig 1 System architecture

To use the system any user should first make some initial setup of its devices – to assign IP addresses, usernames and passwords, as well as to configure the protocol for transport of configurations. Next s/he have to do a registration in the web portal choosing username and password, and to register his devices. For any device an IP address, username and password must be selected. Then alternate connection protocols, configuration transfer protocols, port for connection and backup time, different than defaults may be chosen. These parameters are stored into the database and used by the other modules. Through the web portal the user may also initiate manual restore of configuration for any device, and can choose which from all the stored versions in the database to be restored. For establishing the restoration process, an initial configuration with IP address parameters, usernames and passwords have to be done on the device. When the device is Cisco or Juniper, typically two passwords might be needed – first for establishment of the connection and second for access to privileged mode, from which the file transfer can be started [4].

The business logic module performs the autonomous functions of the system. It reads required parameters from the database and when needed it sends commands to

network interface module for connection to a device and backup of the configuration.

The network device interface module implements various methods for connection to devices and transfer the configuration file to the database. After analysis of available networking devices from different vendors, the following protocols are chosen and implemented: for connection to the device and start of configuration copying process Telnet, SSH and SNMP protocols, and for transfer of the configuration files TFTP is the default protocol, because of its simplify, but FTP, SFTP and SNMP might also be chosen. For better compatibility with different devices a future extension is planned – to store a custom script in the database, that will be started after connection and to initiate custom backups – for example to backup important files of a Linux computer.

### 3. IMPLEMENTATION

Web portal is implemented in PHP. It consists of nine PHP pages and have the structure, shown at Fig. 2.

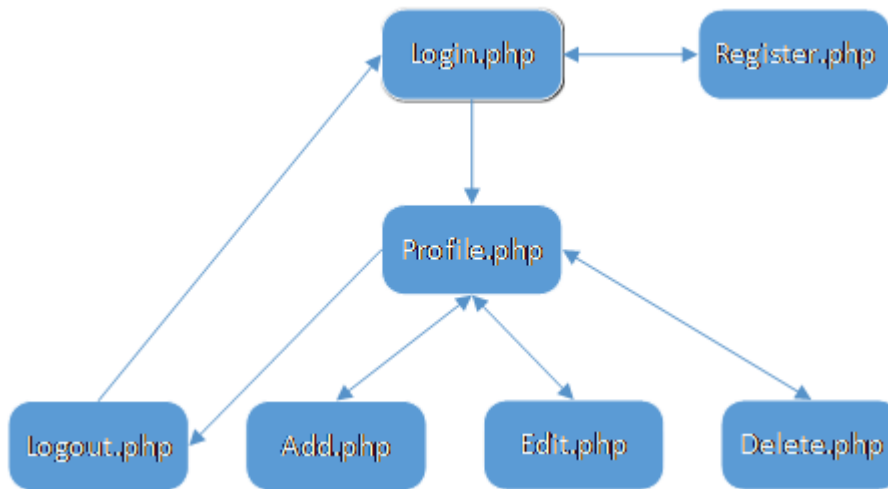


Fig 2 Web portal structure

From login page the user may access their profile or register one. At this time administrators approval of registrations is not implemented, but it is planned for future versions. Through the administrative panel now the administrator can disable access for some users or to define a quota in

meaning of count of devices for every user can be registered in the system. From their profile the user can add, edit or delete devices. A sample screen for users profile is shown at Fig. 3.



Fig 3 Profile page

Through the Edit page a user can also select a manual restore of configuration for some device and to select a configuration from set of stored files.

The Business logic module is a multi-threaded windows service application, implemented in C# with Microsoft

Visual Studio 2010 and has not user interface. Its main thread polls the database for upcoming events and when needed creates a separate thread for every device to be backed up. Then this thread reads the specific parameters for this device and sends commands to network interface module to do the backup. If this device has a limit for stored

configurations, after storing the new backup, it deletes the oldest file from database. If there is an error in any operation, e.g. unsuccessful connection to device or reading the configuration, this module sends an e-mail to the user with details of error. Every activity of the module is written into log table of the database for future reference.

The network devices interface module is also written in C# as Windows Forms application. Its main form is shown at Fig. 4.

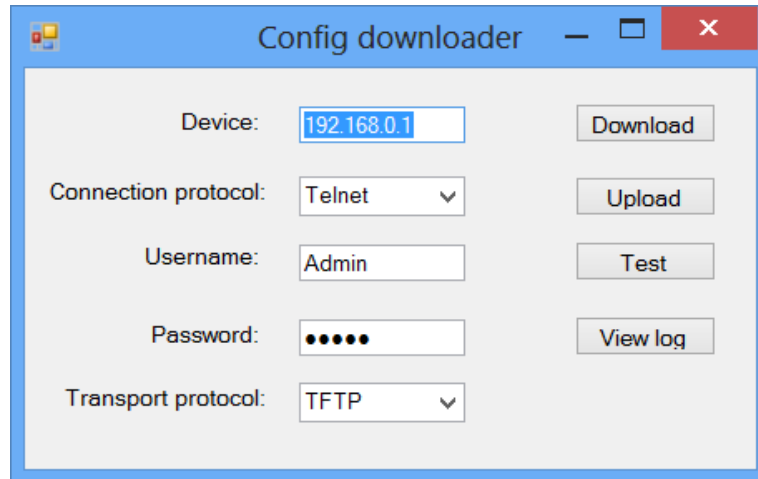


Fig 4 Network interface module

Although it is not necessary this module to have user interface, because it receives commands for its activities from the business logic module, it was easier to test separate communication modules with this user interface, and it allows manual download and upload of configuration, test the communication with the device and view the log stored in database. Except the user interface, this module

implements telnet and SSH client, as well as separate TFTP, FTP/SFTP and SNMP servers, which can receive the file sent by the device.

Database is relational and consists of four tables – **Devices**, **Users**, **Configs** and **Log** and its structure is shown at Fig.5.

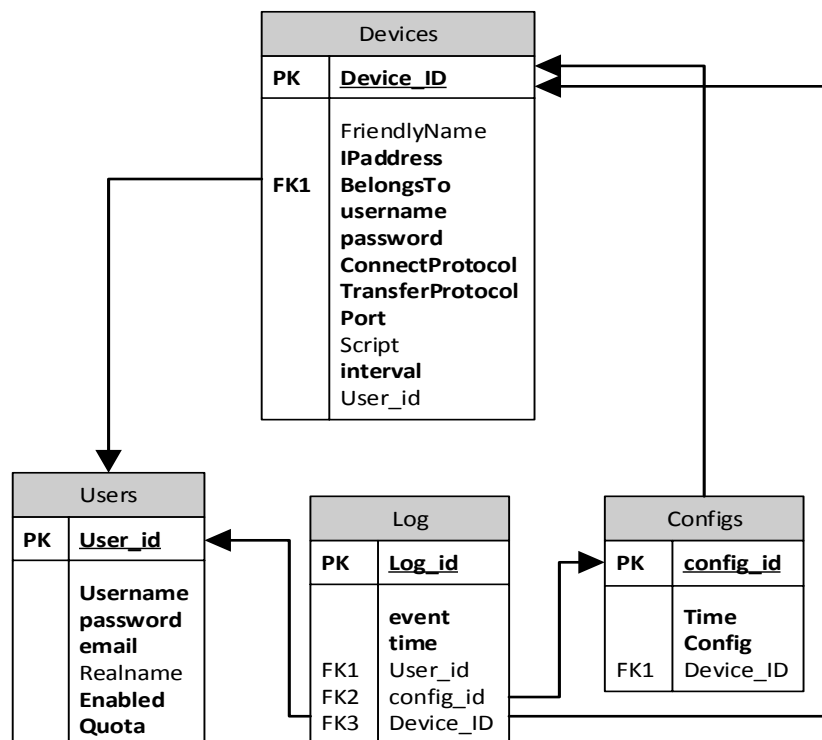


Fig 5 Database structure

In **Users** table the users data are stored – username, password, e-mail address and the user's real name. Through the administrators interface every user can be enabled or disabled and quota may be assigned. In **Devices** table the specific parameters for every device are stored – a friendly name, IP address, connection and transfer protocols, port, username and password for access, an optional script for compatibility and time interval for creating a backup. In **Configs** table the configuration are stored as a binary files. In **Log** table every activity is stored for future reference.

#### 4. RESULTS

The application is tested with few Cisco devices: routers 1721, 2821 and 886, switches: 2950 and 3560. It is also tested with HP 2524 switch, TP-Link TL-SG3210 and D-Link DES 3010G. It can perform successful automatic backups of these devices, connecting through Telnet, SSH or SNMP and transferring files through TFTP, FTP, SFTP and SNMP protocols. It can also perform a manual restore of a configuration when the device has a minimal configuration – IP address parameters and passwords. Although it is not tested, the application should work with Juniper, Nortel and Allied Telesis devices, maybe with much more.

If managed devices are in a remote network, they must have public IP addresses for connection, or a port-forwarding mechanism must be used on the router, connecting the remote network with Internet. In local networks it works well with private IP addresses.

#### 5. CONCLUSION AND FUTURE WORK

This paper proposes a software implementation of a centralized approach for configuration management of networking devices. The access is web-based, which enables to use the system through Internet without special tools. This software can be used from network administrators for backup and restore of configurations as well as in testing network laboratories where different sets of configurations simulating different network setups can be stored.

Future works include testing and integration of devices from more brands and models, experiments for backup and restore of important configuration files of Linux OS, as well as implementation of sets of configurations that can be restored simultaneously to bring a test lab into predefined configuration.

#### ACKNOWLEDGEMENTS

The present document has been produced with the financial assistance of the European Social Fund under Operational Programme "Human Resources Development". The contents of this document are the sole responsibility of "Angel Kanchev" University of Ruse and can under no circumstances be regarded as reflecting the position of the European Union or the Ministry of Education and Science of Republic of Bulgaria.

Project № BG051PO001-3.3.06-0008 "Supporting Academic Development of Scientific Personnel in Engineering and Information Science and Technologies".

#### REFERENCES

- [1] Netgear, NETGEAR Router Setup Manual, [ftp://downloads.netgear.com/files/rp614v3\\_setup\\_manual.pdf](ftp://downloads.netgear.com/files/rp614v3_setup_manual.pdf), date of usage 02.09.2014.
- [2] rConfig. Network Configuration Management, <http://www.rconfig.com/>, date of usage 20.08.2014.
- [3] Shrubbery Networks. RANCID - Really Awesome New Cisco config Differ, <http://www.shrubbery.net/rancid/>, date of usage 22.08.2014.
- [4] Cisco Systems, "Cisco IOS Configuration Fundamentals Configuration Guide", [www.cisco.com](http://www.cisco.com), date of usage 12.09.2013.