Using NETCONF-proxy server to integrate laboratory equipment into software-defined infrastructures

Journal of Physics: Conference Series **1680** (2020) 012022 doi:10.1088/1742-6596/1680/1/012022

Using NETCONF-proxy server to integrate laboratory equipment into software-defined infrastructures

S D Kodolov, A S Klimova, K A Aksyonov and A Yu Filimonov
Ural Federal University, Ekaterinburg, Russia

E-mail: sergey.kodolov@urfu.ru, alina_klimova1503@mail.ru, wiper99@mail.ru, a.filimonov@urfu.ru

Abstract. The essential changes that have taken place over the past decade in the field of telecommunications present new requirements for the educational institutions laboratory complexes management. The modern management concept "Infrastructure as Code" (Infrastructure as Code, IaC) proclaims the usage of a single and universal approach for programmatic management of all components of the communication and computing infrastructure. One of the most common ways to implement this approach is based on the representation of managed unit's configurations in the form of specially marked-up records that form the configuration management database. In this case, process of infrastructure components control is nothing more then sequence of transactions that can be performed for this database records, both locally or remotely - by using network management protocols. The implementation of solutions based on modern universal protocols and network management tools will be complicated when the controlled components do not support modern network management protocols and are separated by the institution's intranet. As one of the possible approaches to solving these problems, we consider the use of gateway communication servers as part of the training classes, which will be able to implement dynamic configuration management of special laboratory equipment of the training class and to provide information interaction between the components of the laboratory complex. The paper considers the choice of control protocols for the gateway server, as well as tools for managing communication infrastructures, and presents an implementation option for this approach for integrating special laboratory equipment of the IRIT RTF at Ural Federal University laboratory classes into a single software-defined laboratory complex.

1. Introduction

The fundamental changes that have taken place over the past decade in the field of telecommunications, at first glance, cast doubt on the possibility of the productive use of traditional laboratory complexes (LC) that are present today in educational institutions. However, it is important to note that the use of new approaches and solutions based on universal configuration protocols allows a completely new view at the role what such LC will be able to play in educational institutions in the near future. Thanks to use modern "Infrastructure as Code" (IaC) method, which involves the exploitation of a single and universal approach for programmatic management of all components of the communication and computing infrastructure, these complexes will be able to integrate into software-defined infrastructure. The implementation of the IaC approach allows us to bring completely new solutions to the tasks of managing the LC infrastructure to a whole new level thanks to the use of universal

management tools and software applications that can reconfigure components in real time [1]. One of the common ways to implement this approach is based on the representation of configurations in the form of XML files that are generated using YANG models [2] to form a configuration management database. Equipment management, in this case, is presented in a form of sequence of transactions that can be performed both locally and remotely by using network management protocols such as NETCONF [3] and RESTCONF [4]. The implementation of integrated solutions based on virtualization technologies for network components, universal management protocols, and network management tools will make it possible to transform traditional laboratory classes into distributed hardware and software systems where students can fulfill not only laboratory classes, but also independent researches [5]. The exploitation of such complexes in the educational process will allow institution to continue the effective use of the presented laboratory equipment and, in addition, makes it possible to smoothly shift education process from the traditional (CLI-based) method to software control for network equipment. To ensure the future unification and optimization of communication infrastructure management systems, the search seems to be important for solutions that allow to provide the integration in such systems of outdated laboratory equipment that is incompatible in control mechanisms with the basic protocols and IaC principles. The following sections discuss the application of gateway communication servers as part of the LC, which implement dynamic configuration management of special classroom laboratory equipment and provide information interaction between the components of the laboratory complex. The paper presents a selection of control protocols for such gateway server building and presents an example of LC construction based on such server for the organization of laboratory classes to study the basics of building software-controlled communication infrastructures.

2. Problems and methods of distributed laboratory complex equipment integration

The basic version of the laboratory complex included training classes in which laboratory classes were conducted to study VoIP, WLAN, and the basics of building software-configured network infrastructures. As part of the complex, both virtual and hardware components of the network infrastructure are applied, which makes it possible to increase the efficiency of using existing equipment, intensify the learning process and, at the same time, avoid the use of non-essential material components [6]. The structure of the Distributed Laboratory Complex (DLC) is shown in figure 1.

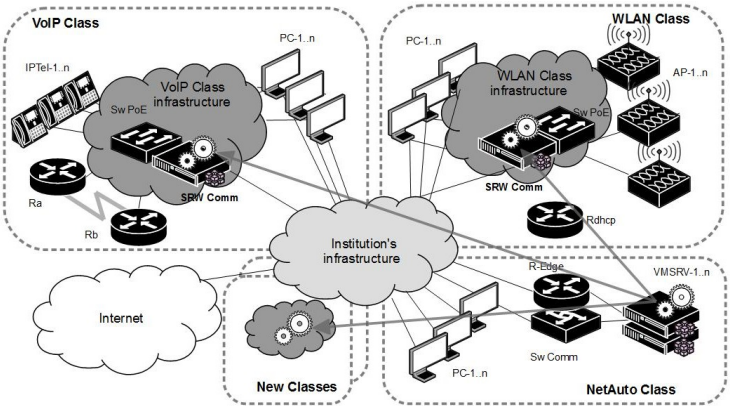

Figure 1. Topology of a distributed laboratory complex.

The DLC consists of several classrooms, each of which contains special laboratory equipment used for laboratory classes, such as IPTel-1..n IP telephones and AP-1..n wireless access points.

The classrooms also host student workstations (PC-1..n) connected to the communication infrastructures of the classrooms and institution intranet (shown in Fig. 1 in dark gray and light gray, respectively). General management and coordination of the interaction of DLC components is carried out by control applications that run on servers of the network automation class SRV_NM. These servers also host virtual network infrastructure components, such as, for example, virtual WLAN controllers or routers, which can be used for laboratory work fulfillment. In this case, the program-controlled communication infrastructure of each of the classrooms is formed by Sw_PoE switches, which provide power to the laboratory equipment and connect workstations of the classroom to it. The main problem that arises in the implementation of program management of the distributed infrastructure described above is that its components are separated by the internal network of the educational institution (intranet), which excludes the possibility of direct information interaction between SRV_NM and special laboratory equipment of classrooms. As one of the approaches to solve this problem, in this paper we consider the inclusion of SRW_Comm gateway communication servers into laboratory classes, which will be intended to perform two main functions:

- implementing dynamic configuration management of special laboratory equipment of the classroom, for example, to perform basic settings of this equipment before the start of the lesson;
- providing information interaction between the DLC components during the lesson, for example, the organization of virtual channels between components of the laboratory complex located in different classes.

The gateway communication server is the intermediary in the process of managing the components of the communication infrastructure of the classroom and the managing application. To perform its functions, this server must use two groups of network management protocols. In our paper, the abbreviation CMP (Central Management Protocols) will be used as the general name for the group of protocols that provide interaction between the gateway server and the management application. For the protocol of this group, the gateway server should act as a server. The abbreviation PMP (Peripheral Management Protocols) will be used to denote a group of protocols that provide the interaction of the gateway server with the components of the communication infrastructure and laboratory equipment of classrooms; for the protocol of this group, the gateway server should act as a client. The "basic protocol" hereinafter is identified as the control protocol, which is simultaneously used by the gateway server both as a CMP and a PMP. The paper discusses the selection of CMP, PMP protocols, the basic protocol of a gateway server, as well as tools for managing DLC communication infrastructures, and presents an implementation option for this approach for integrating special laboratory equipment of the IRIT RTF Ural Federal University training classes into a single software-defined laboratory complex.

3. Network management tools

Communication infrastructures are becoming larger and more complex every year. Technical tasks require more computing resources and data storage. Modern technologies allow the virtualization of systems and equipment of varying complexity. The popularity of virtualization is growing. Virtual components are increasingly being used to build modern complex and productive infrastructures, providing a wide range of capabilities for managing and scaling these infrastructures. But the consequence of these advantages is the increasing complexity of managing virtual components, especially when you need to configure hundreds of devices of the same type. Current trends in network administration are aimed at abandoning device management using the command line [7] in favor of program management using universal protocols, orchestration and automated control systems, including applying configuration

management tools. Configuration management is aimed at maintaining the integrity of the system after the changes, preventing unwanted and unpredictable effects, systematically taking into account the modifications that developers make to the software product during its development and maintenance, formalizing the process of making changes. To perform these tasks, special tools are required, namely, SCM (Software Configuration Management) tools. Both open and proprietary ones and configuration management systems are available on the market. Among the most popular open source tools, Puppet, Chef, Ansible, SaltStack stand out. These tools automate the management of infrastructure components of a modern enterprise in compliance with information security and reliability requirements [8]. Automation of equipment management can also be implemented using developed software scripts and software that perform certain operations. This is convenient when it is necessary to find an effective solution for unique and atypical tasks. The described tools have various mechanisms for connecting to controlled devices. The most effective mechanism is the exploitation of unified protocols using model-oriented configuration repositories. Examples of such protocols are NETCONF and RESTCONF, adopted as standards for software control of equipment and using data models in the modeling language YANG. Support for these standards is implemented in the devices of the largest network equipment vendors, such as, for example, Cisco and Juniper. These standards are also supported by the configuration management tools listed above. Libraries implementing these standards have been created for many programming languages [9].

4. Network management protocols

Network management protocols define procedures that are used to monitor the status and manage configurations of network devices and applications. These protocols can be divided into two main categories: CLI-based and transaction-based. Examples of CLI-based network management protocols are Telnet and SSH. When using this type of protocol , the configuration is changed by executing a command on the managed device, and the configuration itself is an ordered list (script) of control commands. Currently, the share of network infrastructure components that implement the CLI-based configuration management approach is gradually decreasing [7]. This is primarily due to the complexity of integrating such devices into modern software-defined infrastructures. Since almost all components of the communication infrastructure of the training classes of the DLC under consideration (IOS-based switches, routers, access points of Cisco Systems) use this particular type of control protocol (table 1), it is reasonable to consider including these protocols in the list of PMP protocols of the gateway server.

Table 1. Network management protocol support on device software

Component	Software	Network management protocol				
		Telnet	SSH	SNMP	NETCONF	RESTCONF
Juniper M7i	JUNOS 11.1R4.4	+	+	+	+	-
Cisco 2801	Cisco IOS 15.1 (2) T1	+	+	+	-	-
Cisco C3750	Cisco IOS 12.2 (55) SE11	+	+	+	-	-
Cisco C2960	Cisco IOS 15.0 (2) SE4	+	+	+	-	-
Cisco CSR 1000v	Cisco IOS XE 3.13S	+	+	+	+	+

As the name implies, transaction-based control protocols are based on standard mechanisms and procedures for interacting with databases, which makes the configuration process vendor-neutral and provides additional capabilities for managing communication architectures. Today, these functions can be implemented using universal configuration platforms (such as, for example, Ansible) [8]. The emergence and distribution of transaction-based protocols form the basis

of a model-oriented automation of the administration process [8]. Since NETCONF [3] and RESTCONF [4] are the most common transaction-based network management protocols today, it is appropriate to consider using them as the base CMP protocol for implementing a gateway server. Although SNMP can also be assigned to this class of network management protocols with some reservations, the possibility of its use in promising systems seems doubtful [10]. Since when choosing the basic network management protocol, it is primarily necessary to take into account not so much the possibility as the prospects for its use in a particular infrastructure, an assessment of the possibility of using SNMP as the basic CMP protocol was not carried out. It was of interest to consider the possibility of using gRPC as the basic CMP protocol. This protocol uses HTTP/2 as a transport protocol, has support for popular programming languages, has rich functionality (authentication, bidirectional streaming and flow control, blocking / non blocking bindings, cancellation, and timeouts), and uses GPB (Google Protocol Buffers) as a data representation scheme. At the time of this writing, however, gRPC have not had adopted [11] (in comparison with the NETCONF and RESTCONF protocols) implementations yet, therefore, consideration of its use as a basic CMP protocol had also not carried out. NETCONF was originally intended to replace the SNMP protocol in communications equipment configuration management systems. This protocol was developed with the active participation of Juniper specialists and was first implemented on the facilities of this company. Today, NETCONF is supported on the network devices of the vast majority of leading manufacturers of communication equipment (on Cisco Systems devices starting with IOS XE). A characteristic feature of this protocol is the representation of the equipment configuration management process in the form of a sequence of transactions that are performed by the NETCONF client (control object) over the contents of the configuration database (Configuration Database, CDB) of the managed object (NETCONF server). The values of the records of this database uniquely determine the configuration or condition of the nodes of the managed object [12] at the current time. Executing transactions <get ...> or <edit ...> for CDB records, in this case, allows you to analyze or change the state of an object, respectively. A functionally complete set of operations performed on the configuration database includes the operations of creating, reading, updating, and deleting records (Create, Read, Update, Delete – CRUD). The structure of CDB records is dynamically determined by the templates of models of configurable nodes, for the description of which the YANG language is used [13]. CDB records are hierarchical structures marked with XML tags, and the sequence of names of key XML tags acts as an address in the execution of transactions. The advantage of such a navigation scheme is that it can be used to manage the configurations of both a single node and a group of blocks of a managed device. The NETCONF client itself can be located either directly on the managed device itself or can be located in an external system. In the second case, the protocol stack “RPC over SSH” is used to organize sessions of remote interaction with NETCONF [12]. The RESTCONF Protocol is the result of further development of the main ideas that have been the basis of NETCONF. This Protocol can also be used to control the configuration of communication equipment and has many similarities with its predecessor. Similar to the NETCONF Protocol, the RESTCONF Protocol uses CDB to store the configuration of a managed object and a transaction engine to monitor and operate that object. The formal differences between the RESTCONF Protocol are the ability to use JSON format in addition to XML for creating configurations, and the use of HTTP/HTTPS protocols for transporting transactions. In this case, the managed object is an HTTP server that provides the REST API to the control object to perform a set of CRUD operations on JSON or XML-tagged records of its CDB. These features of RESTCONF allow you to significantly simplify the construction of software-defined infrastructures based on IT, which explains the recent growth in popularity of this protocol. It should also be noted that to ensure the security of information interaction between the client and the RESTCONF server, the standard https/TLS scheme is used, which is widely used today for accessing secure Internet resources. Since the

NETCONF and RESTCONF protocols have approximately the same characteristics, to choose one of THEM as the base CMP Protocol, you should consider in more detail the features of using the configuration Protocol as part of an intermediate DLC server

5. Purpose and implementation methods of the Proxy server

As shown in the previous sections, the primary function of an intermediate server is to integrate classroom hardware into a single software-defined DLC Suite. To implement this function, the server must perform the following tasks:

(i) laboratory class communications infrastructure management and monitoring;
(ii) management of interaction with the communication infrastructure of the institution;
(iii) management of non-standard laboratory class components that do not support traditional management protocols.

To perform the first task on the server, the NETCONF / RESTCONF server functions must be implemented when interacting with management applications of the control center. To perform its other tasks, the proxy server must perform the functions of a client of traditional (and morally obsolete today) protocols such as TELNET, SSH, SNMP. In addition, this server must be able to manage non-standard laboratory equipment, such as specialized sensors or test benches. The diagram of information interaction between the CMP Proxy server and the managed DLC components is shown in figure 2.

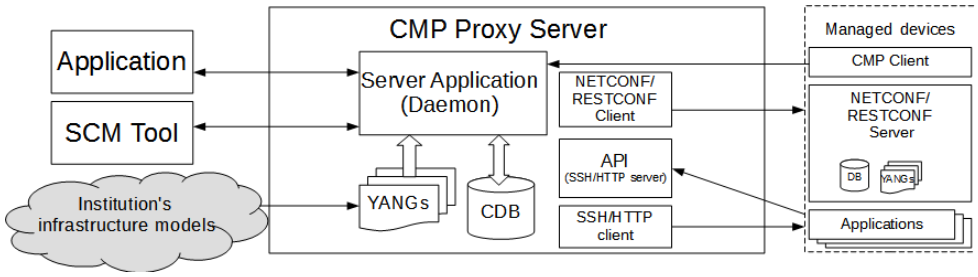

Figure 2. Diagram of information interaction between a CMP Proxy server and DLC components.

Taking into account the list of functions that should be implemented on the intermediate server, it should be noted that the correct and reasonable choice of the CMP management Protocol for interaction with it plays a particularly important role in choosing technical solutions for building. These are the capabilities of this protocol that ultimately determine the effectiveness of managing DLC components and the transparency of monitoring their current status. When comparing the possibilities of using the protocol as a basic CMP one, NETCONF has a noticeable advantage over the RESTCONF protocol due to the fact that it is more suitable for managing the hardware components of the infrastructure [14]. This advantage is largely due to the fact that the NETCONF protocol has a significantly more developed range of operations that are performed on the contents of the CDB. Using NETCONF allows you to determine the candidate configuration, compare this configuration with the current one, apply this configuration, and finally replace it with the current configuration [14]. That is why in modern communication systems, intermediate NETCONF servers are used for implementing a variety of functions. An example of such a solution is the intermediate full-featured NETCONF proxy server, which acts as an intermediary between the element management system (EMS) and the target device, and ensures the organization of a NETCONF session between them, in cases where the organization

of a direct session is impossible. Such solutions are necessary in cases where the EMS and the managed entity are separated by a public network or exchange data with each other using non-standard protocols [15]. An intermediate NETCONF server can also work as a relay bridge providing model-oriented control in cases where the operated object does not support the NETCONF protocol [16]. Limited in their functions NETCONF-Proxy servers today are used even in the field of IoT, despite the stringent requirements that apply to solutions that are implemented in this field [17]. Thus, in modern communication systems, there are enough examples of successful application of proxy servers based on the NETCONF protocol, and therefore there is a fairly wide list of proven solutions, tools and developer kits for its implementation.

6. Using NETCONF-Proxy Server to Integrate Laboratory Equipment

NETCONF-Proxy server as an intermediate server as part of the DLC must implement the following set of functions:

- provide a NETCONF interface for managing and monitoring the entire system;
- connect to remote (local) repositories in order to download the required initial models and their configurations;
- provide storage for the current infrastructure configuration;
- support model-oriented languages like YANG to represent the entire infrastructure as a single model;
- provide configurations for managed device NETCONF clients;
- configure devices that are NETCONF servers;
- provide an API for integrating devices that do not support the NETCONF protocol.

The following tools are considered during design for implementing a proxy server:

- platforms that implement the data storage and YANG models with API support for the integration of equipment that does not support the NETCONF protocol and NETCONF servers, such as, for example, Sysrepo [18];
- open-source NETCONF servers such as Yuma123 [19], Netopeer2 [20];
- end-to-end solutions supporting both the NETCONF server and data storage, such as YumaPro SDK Basic [21].

The use of a proxy server to initialize the starting configuration of the peripheral equipment of a laboratory complex that implements dynamic control of network infrastructure elements [5] based on YANG-models of elements using the NETCONF protocol is considered as a solution in this paper. The proxy server is a Debian 9 virtual machine on which the Yuma123 NETCONF server, a file server, a repository of configuration files in the JSON, and YANG models of the LC equipment are deployed, as well as hardware configuration scripts for use by the automated configuration systems in the YAML format. The use of the Yuma123 NETCONF server is due to the open source code of this tool along with active development of functionality and bug fixes, as well as detailed documentation [19, 22]. A test bench implementing the prototype NETCONF proxy server was deployed during the work. The configuration management tool (Ansible) is installed and the NETCONF server (Yuma123) is deployed on the Debian 9 operating system. Then, configurable device models, configuration templates, device instance parameters, and Ansible script files corresponding to these models were prepared and loaded into the GitHub repository [23]. Ansible script represents the following sequence of tasks:

(i) The test YANG model (figure 3) of the laboratory complex is loaded into the NETCONF server storage

(ii) The NETCONF server starts and the above model loads in RAM
(iii) Configurations are generated based on templates (figure 4) and parameters of device instances (figure 5)
(iv) The generated configurations are deployed to the NETCONF server (figure 6)
(v) A list of configured devices is displayed

```
container devices {
  list device {
    key "name";
    leaf name {
      type string;
      mandatory "true";
      description
        "Device's name. Example value: GigabitEthernet 0/0/0";
    }
    leaf address {
      type dotted-quad;
      mandatory "true";
      description
        "Device's IP address. Example value: 10.10.10.1";
    }
    leaf subnet-mask {
      type dotted-quad;
      mandatory "true";
      description
        "Device's subnet mask. Example value: 255.255.255.0";
    }
    leaf location {
      type string;
      default "P-344";
      description
        "Device's location";
    }
  }
```

Figure 3. Container "Devices".

```
<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <devices xmlns="http://rtf.urfu.ru/ns/yang/rtf-virtlab">
        {% for router in routers %}
        <device>
            <name>{{ router.name }}</name>
            <address>{{ router.ip }}</address>
            <subnet-mask>{{ router.mask }}</subnet-mask>
            <location>{{ router.loc }}</location>
        </device>
        {% endfor %}
    </devices>
</config>
```

Figure 4. Jinja2 Template of configuration.

```
---
netconf_servers:
  hosts:
    localhost:
      ansible_connection: netconf
      ansible_user: root
      ansible_password: {{ password }}
      routers:
        - name: Router1
          ip: 1.1.1.1
          mask: 255.255.255.0
          loc: R-111
        - name: Router2
          ip: 1.1.1.2
          mask: 255.255.255.0
          loc: R-112
        - name: Router3
          ip: 1.1.1.3
          mask: 255.255.255.0
          loc: R-113
```

Figure 5. Configuration parameters.

```
#896
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  message-id="urn:uuid:e6a9a90f-9ca9-4ee1-83c1-7a0dc630b3f8"
  last-modified="2020-04-01T10:45:39Z"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <devices xmlns="http://rtf.urfu.ru/ns/yang/rtf-virtlab">
      <device>
        <name>Router2</name>
        <address>1.1.1.2</address>
        <subnet-mask>255.255.255.0</subnet-mask>
        <location>R-112</location>
      </device>
      <device>
        <name>Router3</name>
        <address>1.1.1.3</address>
        <subnet-mask>255.255.255.0</subnet-mask>
        <location>R-113</location>
      </device>
      <device>
        <name>Router1</name>
        <address>1.1.1.1</address>
        <subnet-mask>255.255.255.0</subnet-mask>
        <location>R-111</location>
      </device>
    </devices>
  </data>
</rpc-reply>
```

Figure 6. Result of configuration.

7. Conclusion

The use of solutions based on universal configuration protocols and software libraries that support these protocols allows us to take a completely new look at the role of laboratory complexes

(LC) of educational institutions. The implementation of the IaC concept, thanks to the use of universal tools and software applications that can reconfigure components in real time, allows us to take managing the LC infrastructure at brand new level. Unfortunately, implementation of solutions based on universal protocols and network management tools becomes complicated in case components do not support modern network management protocols and are separated from control center by the institution intranet. As one of the possible approaches to solving these problems, the work proposes the use of gateway communication servers as part of the training classes, which will be able to implement dynamic configuration control of special laboratory equipment of the training class and to provide information interaction between the laboratory complex components.The paper gives the rationale for the choice of gateway server protocols, as well as tools for managing laboratory complex infrastructures. Paper also concludes an implementation option for this approach for integrating special laboratory equipment of the IRIT RTF Ural Federal University laboratory classes into a single software-controlled laboratory complex is presented. It is shown how the use of such servers makes it possible to unify the management process and, therefore, to provide further development for this complex.

Acknowledgments

This work is supported by Act 211 Government of the Russian Federation, contract #02.A03.21.0006

References

[1] Kundrat J, Vojtech J, Skoda P, Vohnout R, Radil J and Havlis O 2018 *J. Lightwave Technol.* **36** 3105–3114 ISSN 0733-8724, 1558-2213 URL https://ieeexplore.ieee.org/document/8329499/

[2] An Architecture for Network Management Using NETCONF and YANG published: IETF RFC6244, Jun. 2011 URL https://tools.ietf.org/html/rfc6244

[3] Network Configuration Protocol (NETCONF) published: IETF RFC6421, Jun. 2011 URL https://tools.ietf.org/html/rfc6241

[4] RESTCONF Protocol published: IETF RFC8040, Jan. 2017 URL https://tools.ietf.org/html/rfc8040

[5] Filimonov A, Kodolov S, Klimova A and Aksyonov K A 2019 Implementing dynamic management of virtual network infrastructure components *ITTCS*

[6] Filimonov A, Medvedev D, Klimova A and Muravyov A 2018 *Izvestija Vysshih Uchebnyh Zavedenij. Priborostroenie (in Russian)* **61** 1092–1099 ISSN 00213454 URL http://pribor.ifmo.ru/en/article/18361/primenenie_komponentov_virtualnoy_infrastruktury_pri_postroenii_laboratornogo_kompleksa_v_uchebnom_zavedenii.htm

[7] Lerner A 2018 Checking in on the Death of the CLI URL https://blogs.gartner.com/andrew-lerner/2018/01/04/checking-in-on-the-death-of-the-cli/

[8] Johari A 2017 Chef vs Puppet vs Ansible vs Saltstack: Which One to Choose URL https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/

[9] Netconf Central URL http://www.netconfcentral.org/

[10] 2017 It's time to move away from SNMP and CLI and use Model-Driven Telemetry URL https://blogs.cisco.com/developer/its-time-to-move-away-from-snmp-and-cli-and-use-model-driven-telemetry

[11] JamesNK Sravnenie sluzhb gRPC s API-interfejsami HTTP library Catalog: docs.microsoft.com URL https://docs.microsoft.com/ru-ru/aspnet/core/grpc/comparison

[12] Moberg C NETCONF: A new approach to network management URL http://picmg.mil-embedded.com/articles/netconf-new-approach-network-management/

[13] Bjorklund M YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF) library Catalog: tools.ietf.org URL https://tools.ietf.org/html/rfc6020

[14] 2017 NETCONF versus RESTCONF: Capabilitity Comparisons for Data Model-driven Management – Benoît Claise library Catalog: www.claise.be URL https://www.claise.be/netconf-versus-restconf-capabilitity-comparisons-for-data-model-driven-management-2/

[15] Vallin S and Wikström C 2011 *Automating network and service configuration using NETCONF and YANG* pp 267–279

[16] Wang Z and Zheng G Network Configuration Protocol (NETCONF) Proxy library Catalog: tools.ietf.org URL https://tools.ietf.org/html/draft-wangzheng-netconf-proxy-01
[17] Scheffler T and Bonneb O 2017 Manage resource-constrained IoT devices through dynamically generated and deployed YANG models *Proceedings of the Applied Networking Research Workshop on - ANRW '17* (Prague, Czech Republic: ACM Press) pp 42–47 ISBN 978-1-4503-5108-9 URL http://dl.acm.org/citation.cfm?doid=3106328.3106331
[18] 2020 Yang-based configuration and operational state data store for unix/linux applications original-date: 2015-12-09T12:27:03Z URL https://github.com/sysrepo/sysrepo
[19] Vassilev V 2020 The yuma123 repository. Contribute to vlvassilev/yuma123 development by creating an account on GitHub. original-date: 2015-07-19T13:41:44Z URL https://github.com/vlvassilev/yuma123
[20] 2020 NETCONF toolset. Contribute to CESNET/netopeer2 development by creating an account on GitHub original-date: 2015-12-03T10:05:33Z URL https://github.com/CESNET/netopeer2
[21] Multi-protocol Server - YumaWorks library Catalog: www.yumaworks.com URL https://www.yumaworks.com/tools/multi-protocol-server/
[22] Yuma123 Wiki URL https://yuma123.org/wiki/index.php/main_page/
[23] KlimovaAlina 2020 RTF hardware and software stand repository original-date: 2019-10-16T12:01:55Z URL https://github.com/KlimovaAlina/SDN