

ПОСТРОЕНИЕ ЛАБОРАТОРНОГО СТЕНДА ДЛЯ ИССЛЕДОВАНИЯ ПРОГРАММНО КОНФИГУРИРУЕМЫХ СЕТЕВЫХ ИНФРАСТРУКТУР

А. Ю. ФИЛИМОНОВ, К. А. АКСЁНОВ, А. С. КЛИМОВА, С. Д. КОЛОДОВ

*Уральский федеральный университет, 620002, Екатеринбург, Россия,
E-mail: a.filimonov@urfu.ru*

Активное использование технологий виртуализации при построении сетевых инфраструктур позволяет применять новые методы и инструменты для управления компонентами этих структур, что, в свою очередь, расширяет область применения решений, основанных на концепции программируемых (программно конфигурируемых) сетей. По мере расширения сферы внедрения подобных решений возрастает потребность в специалистах, сочетающих знание сетевых технологий с навыками программирования. Предложен подход к построению лабораторного комплекса учебного заведения, который позволяет прививать учащимся практические навыки использования современных методов управления виртуальными компонентами сетевой инфраструктуры. Рассматривается применение таких инструментов, как платформа Ansible, библиотеки программ NCCLIENT и PyEZ для динамического управления виртуальными компонентами аппаратно-программного стенда в составе лабораторного комплекса программно конфигурируемых сетевых инфраструктур ИРИТ-РтФ УрФУ.

***Ключевые слова:** программно конфигурируемые сети, виртуальное разделение сетевой инфраструктуры, технологии виртуализации, лабораторный комплекс*

Введение. Базовым принципом виртуализации является разделение физического ресурса между независимыми виртуальными объектами. Применение решений, которые основаны на принципах виртуализации сетевых функций и ресурсов, позволяет не только существенно повысить эффективность использования сетевой инфраструктуры, но и обеспечивает возможность улучшения качества предоставляемых на ней информационных сервисов [1]. Повсеместное внедрение подобных решений в инфраструктуру центров обработки данных привело к необходимости кардинального изменения традиционных подходов к мониторингу и управлению системами телекоммуникаций и к появлению концепции программируемых или программно конфигурируемых сетей (ПКС, Software-Defined Network, SDN). Классический вариант реализации концепции ПКС предполагает использование контроллера в качестве посредника в процессе информационного взаимодействия между сервисом-приложением и коммуникационной инфраструктурой по конфигурационному протоколу OpenFlow — OF-CONFIG [2].

В последнее время приобретают популярность альтернативные варианты, которые основаны на других протоколах управления коммуникационной инфраструктурой или не предполагают использования контроллеров [3, 4]. Наиболее плодотворной идеей из тех, что были предложены в рамках этой концепции, оказалось предложение о разделении процесса управления коммуникационным оборудованием на отдельные процессы управления (Control Plane) и обработки данных (Data Plane) [5]. Именно это разделение ускорило модификацию процесса управления компонентами сетевой инфраструктуры и привело к появлению целого семейства универсальных конфигурационных платформ (например, Ansible) [6], что, в свою очередь, сделало возможной модельно-ориентированную автоматизацию этого процесса [7]. Сегодня область внедрения технологий программного конфигурирования постоянно расширя-

ется, захватывая все новые направления телекоммуникаций [8] и смежные ИТ-области, что позволяет говорить о программно определяемом конфигурировании как об универсальном подходе (Software-Defined Everything/Anything-SDx) [9], который задает современные направления развития информационных технологий. Именно поэтому знание базовых принципов построения ПКС становится основой профессионального образования современного ИТ-специалиста, что делает актуальным скорейшее внедрение соответствующих курсов в учебный процесс высших образовательных учреждений.

Инструменты для изучения ПКС. По прогнозу компании Gartner [10], которая специализируется на рынках информационных технологий, доля коммуникационного оборудования, управляемого через командную строку (Command Line Interface, CLI), в ближайшие 2—4 года сократится на 50 % [11]. Это дает веские основания полагать, что ИТ-отрасль нуждается в специалистах, которые в дополнение к знаниям основ построения систем телекоммуникаций обладают знанием базовых принципов создания ПКС и навыками программного управления сетевыми инфраструктурами.

В современных системах коммуникаций применяются разнообразные программные средства и информационные ресурсы, которые могут быть использованы при изучении базовых компонентов технологического комплекса ПКС. К таким средствам и ресурсам относятся: порталы повышения квалификации разработчиков, такие, например, как DevNet [12] или EngNet [13], программные платформы, предназначенные для управления конфигурациями [14], например, Ansible, Saltstack, Chef, Puppet, специальные библиотеки для универсальных языков программирования — например NETCONF Client (NCCLIENT) [14], которые предназначены для автоматизации управления сетевыми инфраструктурами. Поэтому представляется целесообразным более подробно рассмотреть некоторые средства и ресурсы с точки зрения возможности их интеграции в учебный процесс для подготовки специалистов в учебном заведении.

Порталам повышения квалификации разработчиков всегда уделяли большое внимание ведущие производители коммуникационного оборудования. Размещая на таких порталах информацию о какой-либо новой или перспективной технологии, они способствовали продвижению своей продукции, которая поддерживает эту технологию. На портале Cisco DevNet [12] разработчики могут представлять свои работы и знакомиться с работами своих коллег. Аналогичный по функциям портал EngNet [13] компании Juniper объединяет полный спектр средств автоматизации процесса разработки и средств взаимодействия разработчиков. На портале представлены описания программных интерфейсов приложений (Application Programming Interface, API), организован доступ к виртуальным лабораторным комплексам, обеспечивается обмен опытом применения инструментов сетевой автоматизации. Использование представленных выше порталов представляется безусловно перспективным для организации аудиторных практических занятий на виртуальном оборудовании. Следует, однако, отметить, что применение подобных ресурсов в решениях, которые предполагают выполнение практических работ, может вызвать проблемы, связанные с необходимостью резервирования оборудования на стороне портала.

Среди платформ для управления конфигурациями наиболее популярна Ansible [14]. Ansible обычно используется для управления вычислительной инфраструктурой в виртуальной или физической среде [15]. Среди преимуществ Ansible стоит отметить отсутствие необходимости развертывания агента на конфигурируемых устройствах, низкую стоимость по сравнению с конкурентами, простоту установки и администрирования [16]. Автоматизация построения сетевой инфраструктуры с помощью Ansible не требует глубоких познаний в области программирования, поскольку наиболее часто используемые операции уже реализованы в виде стандартных модулей. В случае необходимости существует возможность разработать собственный модуль, выполняющий специфические задачи [16]. Благодаря использованию популярных протоколов связи Ansible позволяет автоматизировать управление

конфигурациями как виртуального, так и реального оборудования. В качестве протоколов связи, в зависимости от платформы и назначения модуля, могут быть использованы NETCONF [17], CLI по SSH или API по HTTPS (RESTCONF) [18]. Учитывая то, что Ansible хорошо документирована, удобна в использовании, достаточно универсальна и доступна, следует считать безусловно целесообразным ее включение в образовательный процесс.

Протоколы NETCONF и RESTCONF сегодня активно используются для автоматизации управления сетевым оборудованием [19]. Применение основанных на них приложений позволяет перевести решение задач управления сетевым оборудованием на качественно новый уровень, делая возможной динамическую реконфигурацию сетевой инфраструктуры в реальном масштабе времени [7]. При разработке таких приложений обычно используются библиотеки [20], которые делают процесс автоматизации управления сетевым оборудованием доступным для широкого круга специалистов. Внедрение комплексных решений, основанных на технологиях виртуализации сетевых компонентов [21], универсальных протоколах сетевого управления и соответствующих программных библиотеках, позволит преобразовать традиционные лабораторные комплексы в аппаратно-программные стенды, на которых студенты смогут проводить не только лабораторные занятия, но и самостоятельные исследования.

Построение аппаратно-программного стенда ПКС УрФУ. Для построения аппаратной инфраструктуры лабораторного комплекса использованы маршрутизаторы RA, RB и Edge Router, телефоны IP-Tel1, IP-Tel2, а также сервер Srv, основные характеристики которых приведены в таблице.

Модуль	Оборудование	Программное обеспечение
vR1, vR2	VRF-инстанции на маршрутизаторах Juniper M7i	JUNOS 11.1R4.4 built 2011-07-30 10:27:32 UTC
Edge Router	Маршрутизатор Cisco 2801	Cisco IOS Software, 2801 Software, Version 15.1 (2) T1
Srv	Сервер HP ProLiant DL380 G5 (Intel Xeon E5440 2.83GHz, 64 Гб)	VMware ESXi 5.5.0
vPBX1, vPBX2	Виртуальная АТС на Srv	CentOS Linux release 7.6.1810 (Core), Kernel: Linux 3.10.0-957.10.1.el7.x86_64 Asterisk 16.3.0/FreePBX 15.0.7
IP-Tel1, 2	Cisco Unified IP Phone 9971	9971 SIP IP Phone CUCM Versions: 7.1(5)+Release: 9.4(2)SR4 [9]
PC1, PC2	ПК	Windows 7 x86 (.NET Framework 4.6.1) X-Lite 5.6.0 Mozilla Firefox 68.0.2 vSphere Client 5.5.0

Сервер Srv используется для создания виртуальных телефонных станций vPBX1 и vPBX2, на RA и RB организованы виртуальные маршрутизаторы vR1 и vR2, Edge Router обеспечивает доступ в сеть Интернет. Топология виртуального слоя лабораторного комплекса представлена на рис. 1 (черным цветом обозначены реальные, а серым — виртуальные компоненты инфраструктуры).

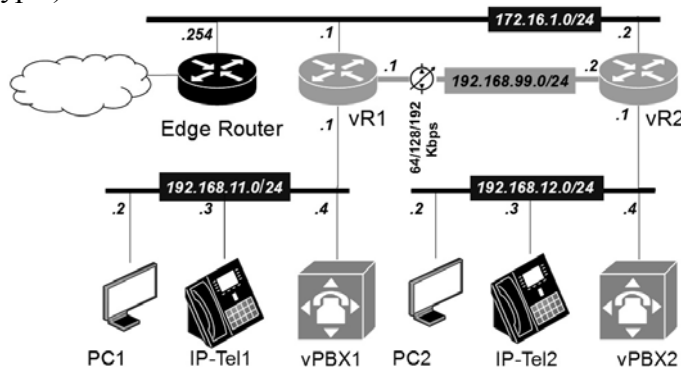


Рис. 1

Для оценки возможности реализации в составе комплекса системы, которая могла бы обеспечить динамическое управление пропускной способностью виртуального канала в процессе аудиовидеовызова, следовало определить характеристики информационного потока. Измерения проводились в одном из виртуальных слоев представленной на рис. 1 топологии с использованием телефонов Cisco Unified IP Phone 9971 [22], в которых для кодирования/декодирования видеопотока используется кодек H.264/AVC [23]. В алгоритме этого кодека применяются разнообразные техники сжатия видеоизображений и компенсации движения, поэтому параметры формируемого им информационного потока сильно зависят от динамических характеристик передаваемого изображения (сцены). Измерения проводились с использованием двух типов сцен: квазистатической, когда в поле зрения камеры практически не попадали подвижные объекты, и динамической, когда в поле зрения камеры находился работающий офисный вентилятор. Результаты этих измерений приведены на рис. 2, а и б соответственно. Представленные гистограммы сформированы по результатам 500 измерений при трех значениях пропускной способности виртуального канала, который связывает vR1 и vR2: 128, 192 и 256 кбит/с (вертикальная штриховка, наклонная штриховка и равномерная заливка соответственно).

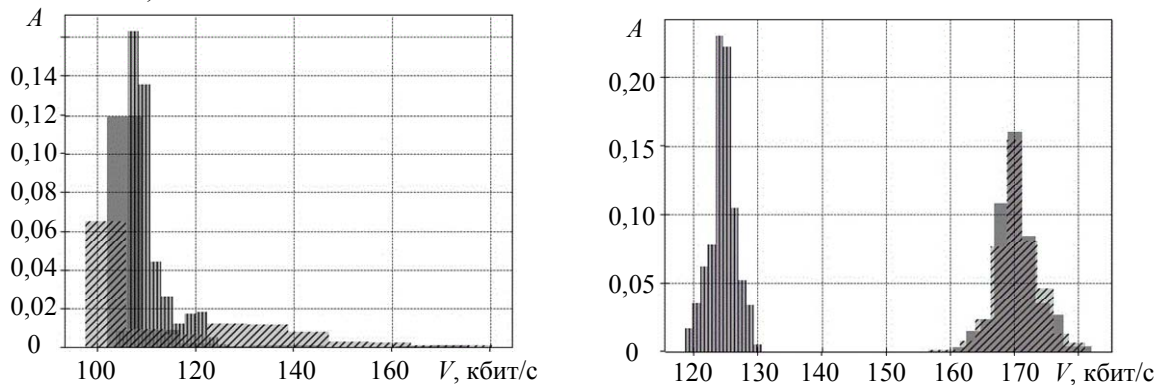


Рис. 2

Полученные результаты позволили установить, что при использовании квазистатического изображения формировался цифровой поток со средней скоростью передачи информации (110 ± 10) кбит/с, для передачи которого было вполне достаточно пропускной способности 128 кбит/с. Также было установлено, что при использовании динамического изображения формировался цифровой поток со средней скоростью (170 ± 10) кбит/с, который передавался без потерь по каналам связи с пропускной способностью 192 и 256 кбит/с. На рис. 3, а и б представлены результаты организации видеовызова при использовании виртуального канала с пропускной способностью 64 и 192 кбит/с соответственно.

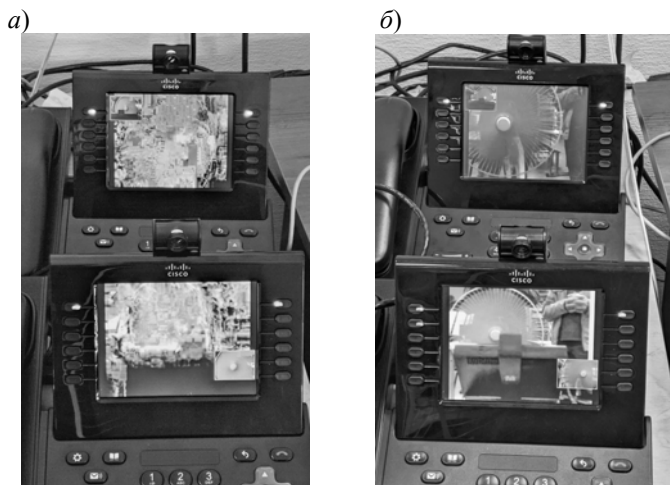


Рис. 3

В ходе испытаний установлено, что передача динамической сцены через канал с пропускной способностью 128 кбит/с приводила к потерям и ухудшению качества изображения из-за перегрузки этого канала. Следует также отметить, что канал с полосой пропускания 192 кбит/с обеспечивает практически безошибочную передачу такого потока. Данные, полученные в ходе испытаний, были использованы при разработке программы-контроллера, которая способна перераспределять ресурс компонента опорной сети (физического канала) между виртуальными слоями.

Программирование стенда ПКС УрФУ. Подготовлены технологические скрипты с использованием NCCLIENT (библиотека Python), позволяющие производить считывание, конфигурирование и мониторинг маршрутизаторов. Примеры технологических скриптов приведены в репозитории [24]. На следующем этапе была разработана программа-контроллер, которая позволяет динамически адаптировать полосу пропускания виртуального канала к требованиям передаваемого сигнала. Программа-контроллер представляет собой скрипт на языке Python, который использует процедуры библиотеки PyEZ (функционально аналогична NCCLIENT и предназначена для использования на устройствах Juniper) [25].

На начальном шаге программа-контроллер определяет среднее количество отброшенных пакетов и разность между максимальной и текущей скоростями передачи данных на виртуальном интерфейсе. На следующем шаге принимается решение о необходимости изменения пропускной способности. Программа-контроллер увеличивает пропускную способность, если среднее число отброшенных пакетов на интерфейсе превышает пороговое значение X . Если разность между максимальной и текущей скоростью передачи данных превышает пороговое значение Y , программа-контроллер уменьшает пропускную способность. В завершение проведены динамические испытания программы-контроллера в составе аппаратно-программного стенда. В процессе испытаний попеременно использовались квазистатические и динамические изображения и фиксировались значения максимально возможной скорости (Limit Rate, LR), текущей информационной скорости (Current Rate, CR) и скорости изменения числа отброшенных из-за перегрузки пакетов (Drop Rate, DR). Временная диаграмма одного из динамических испытаний программы-контроллера представлена на рис. 4. По оси ординат на этой диаграмме откладываются значения информационных скоростей отброшенных за текущий отрезок времени пакетов.

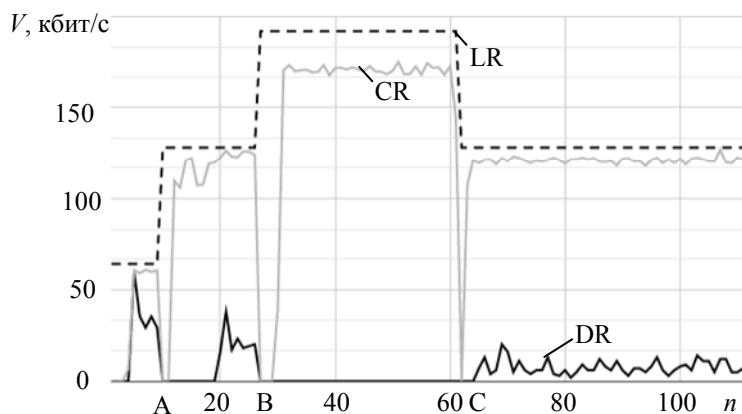


Рис. 4

В начальной фазе испытания пропускная способность виртуального канала установлена равной 64 кбит/с, используется квазистатическая сцена. Поскольку такой полосы пропускания недостаточно для передачи изображения сцены, DR превышает пороговое значение, и программа-контроллер принимает решение об увеличении предельной информационной скорости до 128 кбит/с (точка „А“ на диаграмме). После этого процесс передачи данных приобретает стационарный характер с небольшими флуктуациями относительно среднего значения текущей информационной скорости 110 кбит/с. Примерно через минуту после начала испы-

тания в поле камер видеотелефонов помещался подвижный объект (напольный вентилятор), что, как было показано выше, приводило к увеличению среднего значения информационной скорости до 170 кбит/с. В этом случае, как и в предыдущем, DR превышает пороговое значение, и программа-контроллер принимает решение об увеличении предельной информационной скорости, на этот раз до значения 192 кбит/с (точка „В“ на диаграмме). Еще через минуту подвижный объект удалялся из поля камер видеотелефонов, что приводило к снижению информационной скорости до 110 кбит/с. Поскольку в этом случае расхождение между предельной и текущей скоростями превышает пороговое значение, программа-контроллер принимает решение об уменьшении предельной информационной скорости до 128 кбит/с (точка „С“ на диаграмме).

Многочисленные повторные испытания показали аналогичные результаты, что подтверждает работоспособность аппаратных и программных решений, которые положены в основу описанного стенда. Процесс разработки программно-аппаратного стенда не потребовал дополнительных ресурсов. Его программирование, благодаря наличию полной документации для библиотек NCCLIENT и PyEZ и большого количества примеров, также не вызвало больших затруднений. Все это позволило провести работы по созданию и вводу в эксплуатацию стенда в достаточно сжатые сроки (май—октябрь 2019 года).

Заключение. Внедрение комплексных решений, основанных на технологиях виртуализации сетевых компонентов, универсальных протоколах сетевого управления и соответствующих программных библиотеках, позволяет преобразовать традиционные лабораторные комплексы в аппаратно-программные стенды, на которых студенты смогут проводить не только лабораторные занятия, но и самостоятельные исследования. В работе представлен пример построения подобного стенда и реализации на нем динамического управления сетевыми инфраструктурами в реальном масштабе времени. Применение подобных стендов в учебном процессе позволит продолжать эффективное использование имеющегося в распоряжении учебных заведений оборудования и, кроме того, сделает возможным плавный переход от обучения традиционному (CLI-based) способу управления к программному управлению компонентами сетевой инфраструктуры.

Авторы выражают благодарность Игорю Олеговичу и Илье Игоревичу Ситниковым за методическую поддержку и помощь, которые были оказаны при создании лабораторного стенда.

СПИСОК ЛИТЕРАТУРЫ

1. *Pedreno-Manresa J., Khodashenas P. S., Siddiqui M. S., Pavon-Marino P.* Dynamic QoS/QoE assurance in realistic NFV-enabled 5G Access Networks // 19th Intern. Conf. on Transparent Optical Networks (ICTON). Girona, 2017. P. 1—4.
2. OF-CONFIG 1.2 OpenFlow Management and-Configuration Protocol [Электронный ресурс]: <<https://www.opennetworking.org/images/stories/>>.
3. [downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf](https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf)>.
4. Perspective: Controller-based vs Controllerless-based SDN Solutions [Электронный ресурс]: <<https://www.pluribusnetworks.com/blog/perspective-controller-based-vs-controllerless-based-sdn-solutions/>>.
5. SDN: Where is it now and what is the future? [Электронный ресурс]: <<https://www.computerweekly.com/feature/SDN-Where-is-it-now-and-what-is-the-future>>.
6. *Hakiri A., Gokhale A., Berthou P., Schmidt D. C., Gayraud T.* Software-defined networking: Challenges and research opportunities for future internet // Computer Networks. 2014. Vol. 75, Pt. A. P. 453—471.
7. BEST Software Configuration Management Tools (SCM Tools in 2019) [Электронный ресурс]: <<https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>>.

8. Kundrať J., Vojtech J., Skoda P., Vohnout R., Radil J., Havlis O., YANG/NETCONF ROADM: Evolving Open DWDM towards SDN Applications // J. of Light wave Technology. 2018. Vol. 36, N 15. P. 3105—3114.
9. Dezfouli B., Esmaealzadeh V., Sheth J., Radi M. A Review of Software-Defined WLANs: Architectures and Central Control Mechanisms // IEEE Communications Surveys and Tutorials. 2019. Vol. 21, N 1. P. 431—463
10. Software Defined Everything (SDx) Series: What is SDx? [Электронный ресурс]: <<https://www.sdxcentral.com/cloud/definitions/software-defined-everything-sdx-part-1-definition/>>.
11. Lerner A. Checking in on the Death of the CLI [Электронный ресурс]: <<https://blogs.gartner.com/andrew-lerner/2018/01/04/checking-in-on-the-death-of-the-cli/>>.
12. Cisco DEVNET [Электронный ресурс]: <<https://developer.cisco.com/>>.
13. Juniper EngNet: Elevate development aptitude and break down the barrier associated with network application development [Электронный ресурс]: <<https://eng.juniper.net/site/global/home/index.gsp>>.
14. Chef vs Puppet vs Ansible vs Saltstack: Which Works Best For You? [Электронный ресурс]: <<https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/>>.
15. ANSIBLE ESSENTIALS [Электронный ресурс]: <<https://www.ansible.com/webinars-training/introduction-to-ansible>>.
16. Hall D. Ansible Configuration Management. Packt Publishing Ltd, 2013.
17. Sawtell S. DAY ONE: Automating JUNOS with Ansible. 2nd Edition. Juniper Networks Books, 2018. 398 p.
18. IETF RFC6421 Network Configuration Protocol (NETCONF) [Электронный ресурс] <<https://tools.ietf.org/html/rfc6241>>.
19. IETF RFC8040 RESTCONF Protocol [Электронный ресурс]: <<https://tools.ietf.org/html/rfc8040>>.
20. Wallin S., Wikström C. Automating network and service configuration using NETCONF and YANG // LISA'11 Proc. of the 25th Intern. Conf. on Large Installation System Administration. 2011. P. 22.
21. NCCLIENT: Python library for NETCONF clients [Электронный ресурс]: <<https://github.com/ncclient/ncclient>>.
22. Филимонов А. Ю., Медведев Д. А., Климова А. С., Муравьев А. А. Применение компонентов виртуальной инфраструктуры при построении лабораторного комплекса в учебном заведении // Изв. вузов. Приборостроение. 2018. Т. 61, № 12. С. 1092—1099.
23. Data Sheet Cisco Unified IP Phone 9971 [Электронный ресурс] <https://www.andovercg.com/datasheets/cisco-unified-ip-phone-9971.pdf>
24. ITU-T Rec.H.264 Advanced video coding for generic audiovisual services [Электронный ресурс]: <https://www.itu.int/rec/T-REC-H.264-201906-I/en>
25. Репозиторий аппаратно-программного стенда ПКС РТФ [Электронный ресурс]: <<https://github.com/KlimovaAlina/SDN>>.
26. Junos PyEZ Documentation [Электронный ресурс]: <https://www.juniper.net/documentation/product/en_US/junos-pyez>.

Сведения об авторах

Александр Юрьевич Филимонов

— Уральский федеральный университет, институт радиоэлектроники и информационных технологий, департамент информационных технологий и автоматике; старший преподаватель;
E-mail: a.filimonov@urfu.ru

Константин Александрович Аксёнов

— канд. техн. наук, доцент; Уральский федеральный университет, институт радиоэлектроники и информационных технологий, департамент информационных технологий и автоматике;
E-mail: wiper99@mail.ru

Алина Сергеевна Климова

— студентка; Уральский федеральный университет, институт радиоэлектроники и информационных технологий, департамент информационных технологий и автоматике;
E-mail: alina_klimova1503@mail.ru

Сергей Дмитриевич Колодов

— студент; Уральский федеральный университет, институт радиоэлектроники и информационных технологий, департамент информационных технологий и автоматике;
E-mail: sergey.kodolov@urfu.ru

Поступила в редакцию
18.11.19 г.

Ссылка для цитирования: Филимонов А. Ю., Аксёнов К. А., Климова А. С., Колодов С. Д. Построение лабораторного стенда для исследования программно конфигурируемых сетевых инфраструктур // Изв. вузов. Приборостроение. 2020. Т. 63, № 2. С. 178—186.

BUILDING A LABORATORY BENCH FOR THE STUDY OF SOFTWARE-CONFIGURABLE NETWORK INFRASTRUCTURES

A. Yu. Filimonov, A. K. Aksyonov, A. S. Klimova, S. D. Kodolov

Ural Federal University, 620002, Ekaterinburg, Russia
E-mail: a.filimonov@urfu.ru

The active use of virtualization technologies in the construction of network infrastructures provides the possibility of using new methods and tools to manage the components of these structures, which, in turn, expands the scope of solutions based on the concept of programmable (software-configurable) networks. Expanding the scope of implementation of such solutions, lead to increasing need for specialists who capable to combine knowledge of network technologies with programming skills. An approach to building the laboratory complex of an educational institution is proposed. The approach is reported to allow for continuity in teaching network technologies and for providing students with practical skills of using modern methods of managing virtual components of network infrastructure. Application of such tools as Ansible platform, the libraries of the NCCLIENT and PyEZ programs for dynamic management of virtual components of the hardware-software stand as part of the laboratory complex of software-configured network infrastructures of RTF UrFU is considered.

Keywords: software-configurable networks, virtual separation of network infrastructure, virtualization technologies, laboratory complex

REFERENCES

1. Pedreno-Manresa J., Khodashenas P.S., Siddiqui M.S., Pavon-Marino P. *19th Intern. Conf. on Transparent Optical Networks (ICTON)*, Girona, 2017. P. 1–4.
2. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf>.
3. <https://www.pluribusnetworks.com/blog/perspective-controller-based-vs-controllerless-based-sdn-solutions/>.
4. <https://www.computerweekly.com/feature/SDN-Where-is-it-now-and-what-is-the-future>.
5. Hakiri A., Gokhale A., Berthou P., Schmidt D.C., Gayraud T. *Computer Networks*, 2014, vol. 75, pt. A, pp. 453–471.
6. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>.
7. Kundrát J., Vojtech J., Skoda P., Vohnout R., Radil J., Havlis O. *J. of Light wave Technology*, 2018, no. 15(36), pp. 3105–3114.
8. Dezfouli B., Esmaeelzadeh V., Sheth J., Radi M. *IEEE Communications Surveys and Tutorials*, 2019, no. 1(21), pp. 431–463
9. <https://www.sdxcentral.com/cloud/definitions/software-defined-everything-sdx-part-1-definition/>.
10. <https://blogs.gartner.com/andrew-lerner/2018/01/04/checking-in-on-the-death-of-the-cli/>.
11. <https://developer.cisco.com/>.
12. <https://eng.juniper.net/site/global/home/index.gsp>.
13. <https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/>.
14. <https://www.ansible.com/webinars-training/introduction-to-ansible>.
15. Hall D. *Ansible Configuration Management*, Packt Publishing Ltd, 2013.
16. Sawtell S. *DAY ONE: Automating JUNOS with Ansible*, 2nd Edition, Juniper Networks Books, 2018, 398 p.
17. <https://tools.ietf.org/html/rfc6241>.
18. <https://tools.ietf.org/html/rfc8040>.
19. Wallin S., Wikström C. *LISA'11 Proc. of the 25th Intern. Conf. on Large Installation System Administration*, 2011, pp. 22.
20. <https://github.com/ncclient/ncclient>.
21. Filimonov A.Yu., Medvedev D.A., Klimova A.S., Murav'yev A.A. *Journal of Instrument Engineering*, 2018, no. 12(61), pp. 1092–1099. (in Russ.)
22. <https://www.andovercg.com/datasheets/cisco-unified-ip-phone-9971.pdf>.
23. <https://www.itu.int/rec/T-REC-H.264-201906-I/en>.

24. <https://github.com/KlimovaAlina/SDN>. (in Russ.)
25. https://www.juniper.net/documentation/product/en_US/junos-pyez.

Data on authors

- Alexander Y. Filimonov** — Ural Federal University, Department of Information Technologies and Automation; Senior Lecturer; E-mail: a.filimonov@urfu.ru
- Konstantin A. Aksyonov** — PhD, Associate Professor; Ural Federal University, Department of Information Technologies and Automation; Head of The Department; E-mail: wiper99@mail.ru
- Alina S. Klimova** — Student; Ural Federal University, Department of Information Technologies and Automation; E-mail: alina_klimova1503@mail.ru
- Sergey D. Kodolov** — Student; Ural Federal University, Department of Information Technologies and Automation; E-mail: sergey.kodolov@urfu.ru

For citation: Filimonov A. Yu., Aksyonov A. K., Klimova A. S., Kodolov S. D. Building a laboratory bench for the study of software-configurable network infrastructures. *Journal of Instrument Engineering*. 2020. Vol. 63, N 2. P. 178—186 (in Russian).

DOI: 10.17586/0021-3454-2020-63-2-178-186