

3. Docker SDK, URL: <https://docs.docker.com/engine/api/sdk/> (дата обращения 23.01.2022).

4. Raspberry Pi boards - QEMU documentation, URL: <https://www.qemu.org/docs/master/system/arm/raspi.html> (дата обращения 17.12.2021).

5. BCM2835 ARM Peripherals, URL: <https://www.raspberrypi.org/app/uploads/2012/02/BCM2835-ARM-Peripherals.pdf> (дата обращения 17.12.2021).

ВЕБ-ТРЕНАЖЁР ДЛЯ ОБУЧЕНИЯ СЕТЕВЫМ ТЕХНОЛОГИЯМ

*Золотухин Д.С., Климин Н.А.,
Наумова Е.А., Якименко С.И.*

*Национальный исследовательский университет
«Высшая школа экономики»,
департамент компьютерной инженерии
МИЭМ НИУ ВШЭ*

Аннотация

Перевод образовательного процесса на дистанционный формат является приоритетной задачей многих университетов в последние годы. Одним из решений такой задачи стал формат проведения практических работ определённых дисциплин в веб-тренажёрах [1], в которых студентам предстояло бы работать с эмуляцией некоторой активности в браузере. Именно одному из таких веб-тренажёров и посвящена данная статья.

Введение

На данный момент прохождение лабораторных и практических работ по многим дисциплинам в условиях пандемии стало очень трудоёмким. Студенты должны связываться с преподавателем удалённо и задавать ему вопросы с меньшей степенью взаимопонимания из-за невозможности физически присутствовать в одной аудитории. Более того, работа с некоторыми видами оборудования, такими как серверные шкафы, коммутаторы и маршрутизаторы в контексте дисциплины сетевых технологий, оказалась недостижимой для студентов. Веб-тренажёр позволяет как можно сильнее приблизиться к условиям выполнения работы с необходимым уровнем интерактивности и к тому же, при необходимости, позволяет автоматизировать процесс выставления оценок студентам. Помимо прочего, одной из важнейших задач во время обучения является повышение эффективности усвоения материала студентами. Автоматизация проверки работ позволяет ускорить получение знаний учащимися и за кратчайшие сроки отобразить результат в виде оценки.

В данной работе рассматривается способ создания обучающих систем, а также исследуются темы и задачи, которые могут быть переведены на автоматическую проверку на примере дисциплины «Проектирование компьютерных сетей».

Существующие аналоги

Для обучения компьютерным сетям используются различные методы, в том числе обучение с использованием специализированного программного обеспечения или веб-сервисов.

На сегодняшний день популярные эмуляторы сетевых взаимодействий распространяются в формате desktop-приложений, в которых уже настроена определённая коллекция лабораторных работ, которые необходимо будет пройти по заданному сценарию. Три самых популярных из не коммерческих программ – Cisco Packet Tracer [2], GNS3 [3], Dynamips [4]. Помимо очевидного недостатка в виде того, что каждый эмулятор — это программа для рабочего стола, присутствуют ещё некоторые неприятные для функциональности моменты:

– Невозможно манипулировать информацией о принадлежности студентов к определённым группам, выдавать задания необходимо каждому студенту персонально с методическими указаниями в том или ином виде.

– Некоторые аналоги при прочих плюсах, заточены только под эмуляцию оборудования конкретных компаний. Dynamips и Cisco Packet Tracer способны воспроизводить поведение только оборудования Cisco [3, 4].

– Отсутствие свободы создания заданий для выполнения. В тех средах, где предусмотрена не косвенная возможность прохождения практических работ, нет возможности создать свои работы для прохождения и проверки.

Наиболее функциональные аналоги, такие как Cisco Packet Tracer, вместе с тем являются проприетарными, что не позволяет использовать их без лицензии и не исключает возможности блокировки со стороны владельца.

Цель проекта и задачи

Целью разработки является создание гибкого в контексте выдачи и составления практических работ по курсу сетевых технологий, а также их последующая автоматизированная оценка. Преподаватель и команда поддержки проекта должны иметь возможность быстро изменять наиболее непостоянные аспекты, такие как сценарий работы и группа студентов для выдачи работы. Основными задачами были выделены:

– Создание микро-сервисной архитектуры веб-приложения.

– Использование внутриуниверситетской авторизации для выполнения работ только студентами вуза.

– Создание независимого от прочих сервисов проекта генератора топологии сети.

– Создание эмулятора протоколов маршрутизации для использования в практических работах.

– Реализация интерактивного интерфейса веб-приложения для удобства пользования.

Микро-сервисная архитектура

Для работы веб-сервиса используется микро-сервисная архитектура. Данный подход позволяет разделить большие системы на небольшие функцио-

нальные блоки. Для развертывания сервиса был выбран подход «сервис как сущность в контейнере». При таком подходе каждый блок упаковывается в отдельный контейнер и изолируется от других с помощью виртуализации.

Для инкапсуляции каждого сервиса используется система контейнеризации Docker и соответственно Docker-контейнеры для каждого из следующих блоков: генератор заданий, генератор топологий, серверная часть проекта. Данная структура также позволяет путем взаимодействия блоков через REST API объединить все части системы.

К тому же, эта система контейнеризации завоевала популярность в последние годы в связи со следующими факторами:

- Более быстрая работа контейнеров, по сравнению с виртуальными машинами [5].
- Относительно лёгкое развёртывание сервера, на базе Docker [6].

Внутриуниверситетская авторизация

Для обеспечения доступа к лабораторным работам только студентам и преподавателям НИУ ВШЭ в веб-тренажёр была интегрирована система авторизации портала Smart LMS. Данная система обладает одним существенным плюсом: она независима и находится полностью под контролем университета. В то время, как авторизация таких гигантов, как Google с интеграцией через OAuth2 [7] пользуется большой популярностью [8], она может быть заблокирована извне. Использование этой функции извне сразу сделало бы невозможным функционирование веб-тренажёра.

Генерация топологии

Генератор топологии реализован в виде отдельного сервиса, который будет предлагать способы взаимодействия посредством REST API. Делается это с той идеей, что в будущем его можно будет использовать и с другими сервисами, а не только с веб-тренажёром. Все необходимые зависимости для генератора также предоставлены ему в виде инкапсулированных сервисов. Команда разработки придерживалась принципа «отдельная задача – отдельный контейнер».

Что касается реализации генератора топологий, был выбран язык программирования Python, а микросервис построен на сервере Flask. Основой алгоритма генерации топологии и подсетей для заданий являются модели Эрдёша-Реньи и Барабаши-Альберта [9]. Работа генератора представляет собой создание случайного графа на матрице смежности и нормализация матрицы путем соединения устройств с маршрутизаторами. Были рассмотрены и другие варианты генерации графов с предпочтительным связыванием [10], однако большая часть алгоритмов имели неоптимизированную структуру или не подходили для работы.

Для программной эмуляции топологий, состоящих из подсетей и маршрутизаторов, было использовано модель представления данных в виде двудольно-

го графа или биграфа. Пример представления сети в виде биграфа изображен на рис. 1.

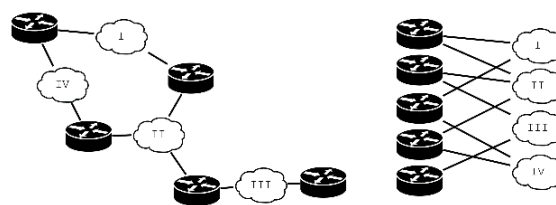


Рис.2. Топология сети в виде двудольного графа

Реализуемые протоколы маршрутизации

В веб-тренажёре предлагается реализовать эмуляцию по 3-м протоколам маршрутизации и протоколу удаления петель или циклов в коммутационной сети. RIP – первый протокол динамической маршрутизации, из ошибок и идей которого развились другие протоколы. OSPF – самый популярный на данный момент протокол динамической маршрутизации [11], понимание которого пригодится в работе почти с любой компьютерной сетью. EIGRP – относительно молодой протокол, созданный компанией Cisco, знание которого необходимо, ведь зачастую он показывает результаты лучше [12], чем у OSPF, который распространён повсеместно. STP – протокол, особенно важный к изучению, так как являлся прообразом RSTP, PVSTP, MSTP, SPB и других, которые сейчас формируют основную группу инструментов для решения проблем в сетях.

Заключение

Как итог, конечным результатом работы выступит гибкое и функциональное веб-приложение, в котором на этапе проектирования были выделены наиболее часто изменяемые части и обработаны с точки зрения архитектуры. Ядром эмулятора будет генератор сетевой топологии, которые полностью независим и может использоваться в других проектах посредством REST API. Генерация будет основываться на модифицированном методе Барабаши-Альберта, а топология представляться в виде двудольного графа. Проект будет использовать авторизацию SmartLMS, находящийся под полным контролем университета, и не будет зависим от сторонних провайдеров авторизации.

Список литературы

1. Mkrttchian V., Miroljub K., Škraba A., Yernanosyan H., Borštnar M. Development of Educational Web-Based Simulator and its Evaluation.//IFAC. - Proceedings Volumes. - 2008. - Vol.41. - №2. - URL: <https://www.sciencedirect.com/science/article/pii/S1474667016405343?via%3Dihub>
2. Кобылянский В.Г., Семенцова А.Г. Анализ компьютерных сетей в программном пакете Cisco Packet tracer // Технические науки – от теории к практике. - 2017. - №3 – с.63. - URL: <https://cyberleninka.ru/article/n/analiz-kompyuternyh-setey-v-programmnom-pakete-cisco-packet-tracer>

3. Davies J. N., Comerford P., Grout V., Verovko M. V., Stasiuk S. S. Comparison of network simulators in ip networks // ММС. - 2014. - №4. - URL: <https://cyberleninka.ru/article/n/comparison-of-network-simulators-in-ip-networks>

4. Лапони́на О.Р., Сизов М.Р. Лабораторный стенд для тестирования возможностей интеграции ПКС-сетей и традиционных сетей // International Journal of Open Information Technologies. - 2017. - Vol.5. - №9. - URL: <https://cyberleninka.ru/article/n/laboratornyy-stend-dlya-testirovaniya-vozmozhnostey-integratsii-pks-setey-i-traditsionnyh-setey>

5. Potdar A. M., Narayan D.G. Performance Evaluation of Docker Container and Virtual Machine // Procedia Computer Science. - 2020. - Vol.171. - URL: <https://www.sciencedirect.com/science/article/pii/S1877050920311315>

6. Васильев П.А. Развертывание сервера с помощью технологии docker // Вестник науки и образования. - 2016. - Vol.24. - №12. - URL: <https://cyberleninka.ru/article/n/razvertyvanie-servera-s-pomoschyu-tehnologii-docker>

7. Шейкина Е.П., Лазарева О.Ю. Проблема безопасности авторизации аккаунта через социальные сети // Вестник МГУП. - 2015. - №1. - URL: <https://cyberleninka.ru/article/n/problema-bezopasnosti-avtorizatsii-akkaunta-cherez-sotsialnye-seti>

8. Torroglosa-García E., Pérez-Morales A.D., Martínez-Julia P., Lopez D.R., Integration of the OAuth and Web Service family security standards // Computer Networks. - 2013. - Vol.57. - №10. - URL: <https://www.sciencedirect.com/science/article/abs/pii/S1389128613001138>

9. Берновски М.М., Кузюрин Н.Н. Случайные графы, модели и генераторы безмасштабных графов // Труды ИСП РАН. 2012. №. URL: <https://cyberleninka.ru/article/n/sluchaynye-grafy-modeli-i-generatory-bezmasshtabnyh-grafov>

10. Юдин Е. Б. Генерация случайных графов предпочтительного связывания // ОНВ. 2010. №2 (90). URL: <https://cyberleninka.ru/article/n/generatsiya-sluchaynyh-grafov-predpochtitelnogo-svyazyvaniya>

11. Redmond T. Exchange transport and routing // HP Technologies - Microsoft Exchange 2007 - Digital Press. - 2007. - URL: <https://www.sciencedirect.com/science/article/pii/B9781555583477500077?via%3Dihub>

12. Абрамки́на О.А. Анализ эффективности протокола EIGRP // T-Comm. 2017. - №10. - URL: <https://cyberleninka.ru/article/n/analysis-of-the-effectiveness-of-the-eigrp-routing-protocol>

СТАБИЛИЗАЦИЯ ОПТОВОЛОКОННОГО ПРИЕМНИКА КВАНТОВЫХ КОГЕРЕНТНЫХ ОПТИЧЕСКИХ СИГНАЛОВ

*Венедиктов И.О., Пестов Ю.С.,
Ганиев Р.Р., Сыч Д.В.*

*Национальный исследовательский университет
«Высшая школа экономики»
Базовая кафедра квантовой оптики и
телекоммуникаций ЗАО «Сконтел»,
МИЭМ НИУ ВШЭ*

Аннотация

В данной работе представлены математическая модель волоконного приемника квантовых оптических когерентных сигналов и расчет допустимых фазовых и поляризационных флуктуаций в приемнике для достижения максимально возможного отношения оптической мощности при конструктивной и деструктивной интерференции.

Введение

Проблема различения квантовых оптических когерентных сигналов возникает в квантовой и классической коммуникации на дальние расстояния. При большой интенсивности этих сигналов (или иначе при большом среднем числе фотонов) когерентные сигналы легко различимы, хотя и не являются ортогональными. Но с уменьшением среднего числа фотонов в сигнале состояния будут становиться все менее различимыми из за Пуассоновского распределения числа фотонов в сигнале. Существуют методы, которые позволяют уменьшить ошибку различения когерентных состояний. Один из таких методов — приемник Кеннеди, который основан на смещении фазово-модулированных состояний $|\alpha\rangle$ и $|-\alpha\rangle$ в оптическом фазовом пространстве в состояния $|2\alpha\rangle$ и $|0\rangle$ посредством оператора смещения $\hat{D}(\alpha)$. На практике оператор смещения реализуется при помощи интерференции сигнала с некоторым локальным осциллятором. Поскольку добиться когерентности двух независимых источников довольно сложно, то используется один источник излучения, который попадая на светоделитель делится на сигнал, который смещается по фазе на 0 или π (состояния $|\alpha\rangle$, $|-\alpha\rangle$ соответственно) и на локальный осциллятор, который интерферирует с сигналом давая на выходе состояния $|2\alpha\rangle$ или $|0\rangle$. Но в реальном эксперименте невозможно получить вакуумное состояние, поскольку нельзя идеально согласовать по фазе и поляризации сигнал и локальный осциллятор, что увеличивает ошибку различения состояний. В случае оптоволоконной реализации приемника данные флуктуации возникают из за температурных флуктуаций и механических вибраций в волокне, приводящих к флуктуациям фазы и поляризации излучения. Для уменьшения данных флуктуаций существуют различные подходы, реализующие активную стабилизацию фазы для увеличения точности работы приемника.