

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ О.В. Непомнящий
подпись инициалы, фамилия

« ____ » _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование направления

Генератор конфигураций для кольцевой структуры ядра сети

тема

Руководитель

подпись, дата

ст. преподаватель

должность, ученая степень

О.В.Шмелев

инициалы, фамилия

Выпускник

подпись, дата

С.В.Брюханов

инициалы, фамилия

Нормоконтролер

подпись, дата

ст. преподаватель

должность, ученая степень

О.В.Шмелев

инициалы, фамилия

Красноярск 2021

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

О.В. Непомнящий

подпись инициалы, фамилия

« ____ » _____ 2021 г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме бакалаврской работы
бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту Брюханову Сергею Владимировичу
фамилия, имя, отчество

Группа КИ17-06Б Направление (специальность) 09.03.01
номер код

Информатика и вычислительная техника
наименование

Тема выпускной квалификационной работы Генератор конфигураций для
кольцевой структуры ядра сети

Утверждена приказом по университету № _____ от _____

Руководитель ВКР О.В.Шмелев, старший преподаватель кафедры

«Информатика и вычислительная техника» ИКИТ СФУ
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: задание для ВКР

Перечень разделов ВКР: Введение, анализ существующих решений и
постановка задач, описание программы и документация, описание
выполненных работ, заключение, список использованных источников.

Перечень графического материала: презентация в формате PowerPoint

Руководитель ВКР _____ О.В.Шмелев
подпись, дата инициалы, фамилия

Задание принял к исполнению _____ С.В.Брюханов
подпись, дата инициалы, фамилия

« ____ » _____ 2021 г.

РЕФЕРАТ

Отчет о практике по теме «Генератор конфигураций для кольцевой структуры ядра сети» содержит 20 страниц текстового документа, 8 рисунков, 2 приложения и 2 использованных источника.

ГЕНЕРАТОР КОНФИГУРАЦИЙ, СЕТЕВОЕ ОБОРУДОВАНИЕ, CISCO, PHP, ВЕБ-ПРИЛОЖЕНИЕ.

Объект проектирования: Генератор конфигураций для кольцевой структуры ядра сети.

Цель работы: разработать конфигурацию бюджетной кольцевой структуры ядра сети, а также генератор конфигураций для кольцевой структуры ядра сети.

В выпускную квалификационную работу входит введение, 3 главы и заключение.

Содержание работы: в результате работы была разработана конфигурация бюджетной структуры ядра сети с резервированием и оптимизацией загрузки на основе коммутаторов Cisco 3560-24PS и было разработано решение для автоматической генерации конфигураций кольцевой структуры ядра сети на языке программирования PHP.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Анализ существующих решений и постановка задач	5
1.1 Анализ существующих решений	5
1.1.1 SolarWinds	5
1.1.2 ManageEngine Network Configuration Manager	6
1.1.3 GEN.IT	7
1.2 Постановка задач	8
1.3 Вывод по главе	9
2 Описание программы и инструкции	10
2.1 Описание программы	10
3.2 Инструкция пользователя	12
3.3 Инструкция системного администратора	13
3.3 Вывод по главе	14
3 Описание выполненных работ	15
3.1 Разработка топологии кольцо	15
3.2 Разработка шаблонов конфигураций	16
3.3 Компоновка и настройка сервера приложения	18
3.4 Вывод по главе	19
ЗАКЛЮЧЕНИЕ	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	22
ПРИЛОЖЕНИЕ А	23
ПРИЛОЖЕНИЕ Б	24

ВВЕДЕНИЕ

В современном мире большие компании могут себе позволить держать штат сетевых инженеров или системных администраторов, но не всегда это необходимо. Что касается системного администрирования - конечно, необходима поддержка в течении всего рабочего дня, но в маленьких компаниях настройка сетей - это очень редкая задача, которую можно частично автоматизировать.

Существуют тысячи команд для настройки и управления сетевым оборудованием, но запомнить все эти команды невозможно. А если даже запомнить их все, то существует большая вероятность сделать ошибку в процессе настройки оборудования, что может привести к неработоспособности сети или сбоях в ее работе. Если сеть состоит из большого количества узлов, то поиск ошибки в ней будет достаточно долгим и кропотливым процессом.

Большинство системных администраторов хранят шаблоны конфигураций сетевых устройств в текстовых файлах, и для того, чтобы настроить новое оборудование появляется необходимость искать эти файлы и что-то менять. Повседневные задачи для поддержания нормальной работы сети не являются исключением и бой с этими задачами обычно ведется с помощью файлов-шаблонов конфигураций. Безусловно, есть различные приложения по управлению сетью, но их используют далеко не все из-за стоимости, либо задачи по конфигурированию оборудования выполняются настолько редко, что данная покупка будет не обоснована [5].

Крупные сетевые компании готовы предоставлять свое оборудование и программные решения для любых нужд, но зачастую это не могут себе позволить небольшие компании. В большинстве случаев компании выбирают между производительностью и стоимостью. Иногда появляется возможность выбрать более дешевое оборудование, но настройка всей сети займет продолжительное время. Для того, чтобы автоматизировать рутинную работу

и исключить ошибки, можно создать шаблон конфигурации и изменять в нем только входные параметры. Это позволит автоматизировать дальнейшую настройку сети, создав один раз список всех команд для настройки определенного устройства. Генератор конфигураций позволит упростить и свести к минимуму время настройки оборудования.

В связи с тем, что конфигурация разных устройств различается, невозможно сделать один шаблон для всех устройств даже одного производителя.

1 Анализ существующих решений и постановка задач

1.1 Анализ существующих решений

За последнее время многое было сделано для облегчения работы системных администраторов. Существует множество ручных процессов, которые были автоматизированы с использованием различных сетевых инструментов. Программное обеспечение, которое автоматически генерирует параметры конфигураций сетевого оборудования стали очень популярными благодаря своей необходимости, ведь ручной процесс настройки оборудования подвержен очень многим ошибкам и расходуется много времени. В результате исследования существующих решений в открытом доступе было найдено 3 менеджера конфигураций для автоматического генерирования конфигурации, которые являются лидерами среди таковых.

1.1.1 SolarWinds

SolarWinds Network Config Generator это простой, но полезный инструмент сетевого администрирования. Он автоматически создает сценарии конфигурации на основе шаблонов для каждого устройства в сети, если оно имеет командную строку. Это решение уже имеет некоторые встроенные шаблоны, а также можно создавать собственные сценарии для данного инструмента для запуска неограниченного количества команд в сети. Для того, чтобы выполнить сценарий конфигурации, необходимо для каждого устройства войти в интерфейс командной строки.

Генератор сетевой конфигурации SolarWinds использует интернет-протокол для управления устройствами SNMP для связи с устройствами в сети, поэтому необходимо включить SNMP на устройствах для получения шаблонов конфигурации. Решение имеет как бесплатную версию с ограниченным функционалом, так и платную.

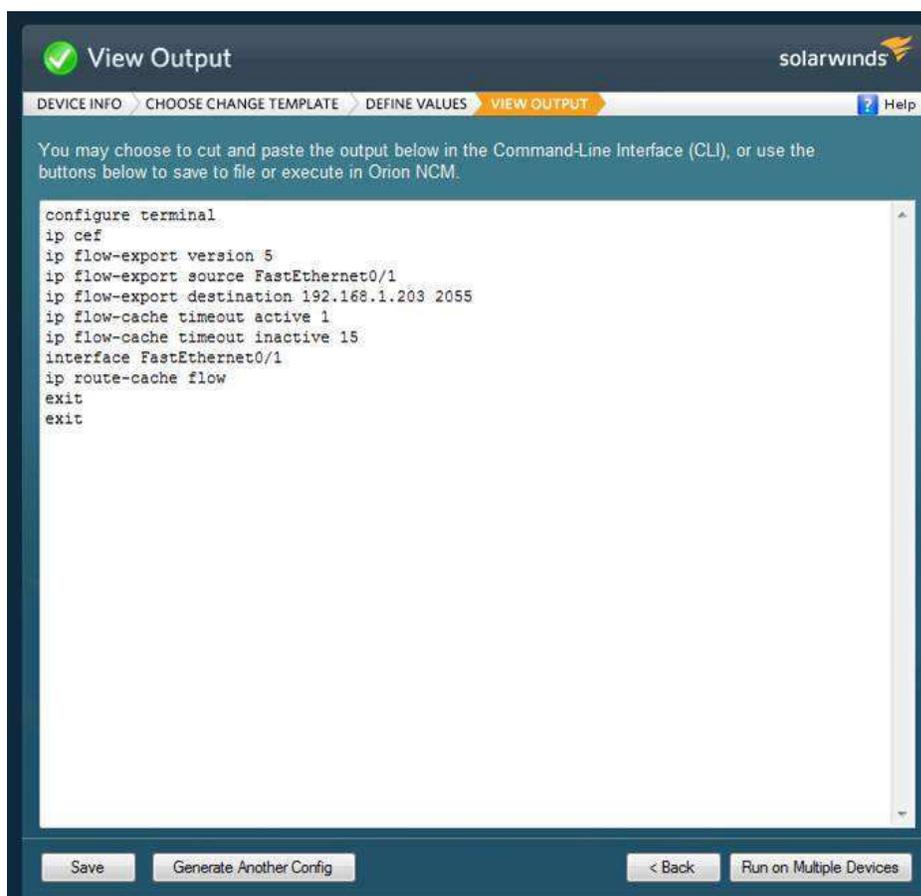


Рисунок 1 – SolarWinds

1.1.2 ManageEngine Network Configuration Manager

ManageEngine- это комплексный инструмент управления изменениями, конфигурацией и соответствием сети (NCCM), который будет работать на различных сетевых устройствах. Он имеет централизованный веб-интерфейс, из которого можно отслеживать все настройки устройства и определять, когда вносятся какие-то изменения. Данный инструмент позволяет применять параметры конфигурации сразу ко всем устройствам, а также автоматизирует некоторые повторяющиеся задачи по настройке.

Конфигуратор сети ManageEngine имеет недостаток: бесплатная версия ограничена конфигурацией только двух устройств.

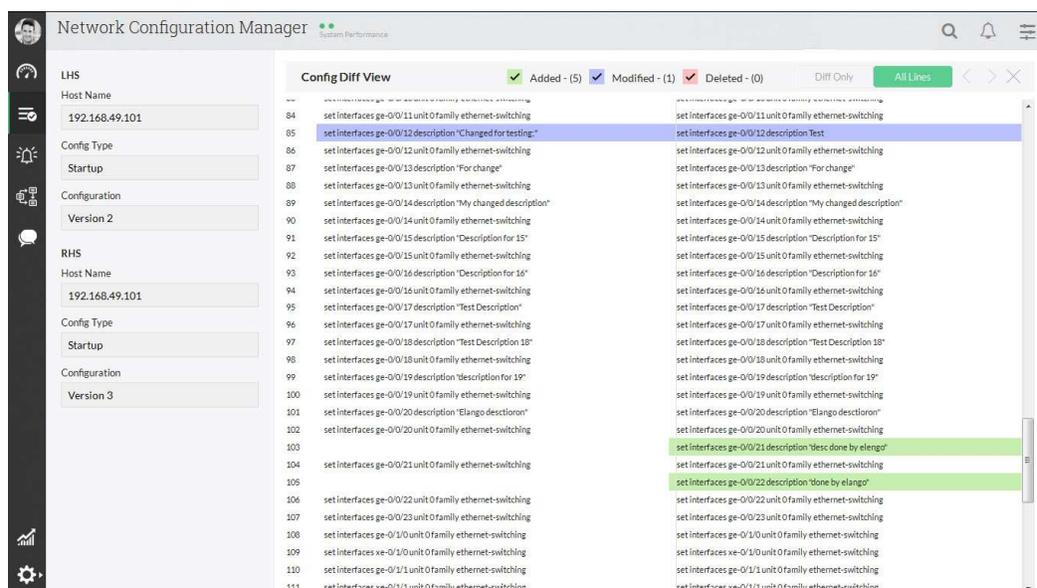


Рисунок 2 – ManageEngine

1.1.3 GEN.IT

GEN.IT- это инструмент, который можно использовать для генерации параметров конфигурации для множества устройств от разных производителей. Также его можно использовать для массового развертывания настроек в короткие сроки. GEN.IT представляет простой способ сохранить настройки после их развертывания, включая возможность их изменения. Для работы программы не требуется доступ к сети.

Данное программное обеспечение не имеет встроенных шаблонов. Пользователь должен сам их создать и сохранить. Затем данное решение считывает информацию из электронной таблицы и объединяет переменные и шаблоны в полезные файлы конфигурации. Затем эти файлы сохраняются в виде текстовых файлов, которые могут быть развернуты на сетевых устройствах.

GEN.IT доступна для бесплатного ознакомления. Функционал полностью сохраняется, но при сохранении шифрует некоторые из сгенерированных конфигураций.

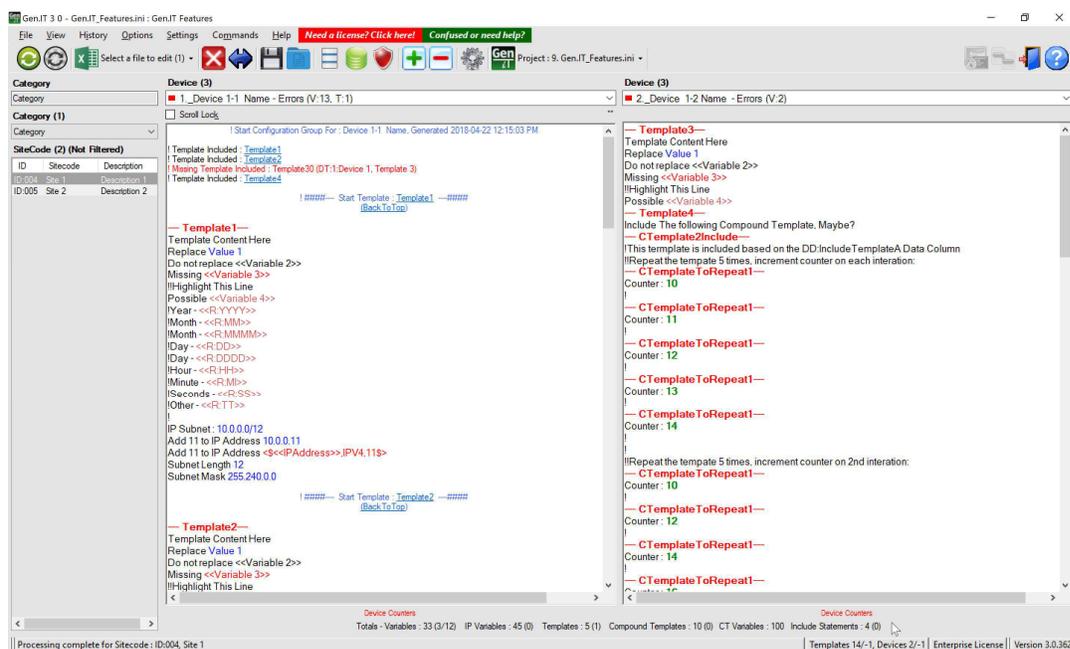


Рисунок 3 – GEN.IT

1.2 Постановка задач

Задачи проектирования:

- Разработка структуры генератора конфигураций для сетевого оборудования;
- Разработка и тестирование бюджетной кольцевой структуры ядра сети;
- Разработка шаблона конфигураций для кольцевой структуры ядра сети;
- Тестирование полученных конфигураций в симуляторе сети передачи данных Cisco Packet Tracer и на лабораторном стенде;
- Компоновка и настройка сервера приложения.

Генератор конфигураций должен удовлетворять следующим требованиям:

- Возможность выбора шаблона из зарегистрированных модулей для создания конфигурации;
- Возможность использования функций при создании конфигураций;

- В зависимости от поставленной задачи, ввод только необходимых данных от пользователя (для определенных шаблонов- определенные входные данные);
- Валидация вводимых данных;
- Возможность сохранения полученных конфигураций в системе;
- Возможность просмотра и удаления сохраненных конфигураций.

1.3 Вывод по главе

- 1) Проведен анализ существующих аналогов разрабатываемого генератора.
- 2) Сформулированы требования для генератора конфигураций и поставлены задачи проектирования.

2 Описание программы и инструкции

2.1 Описание программы

В данной работе рассматривается вариант дешевого программного обеспечения конфигурации сетевого оборудования для упрощенной настройки сетевого оборудования и хранения созданных конфигураций. Данный вариант может подойти для небольших и средних компаний.

В интернете есть готовые генераторы конфигураций на языках Bash и Python. Как правило, приложение представляет собой консоль либо терминал, в которой необходимо ввести данные строго по порядку и без ошибок. Проблема состоит в том, что данной программе нужно будет обучать персонал, и нет гарантии, что в процессе пользовательского ввода не будет ошибок, из-за которых нужно будет повторить ввод данных с самого начала.

Для решения рутинных задач было решено сделать генератор конфигураций с графическим интерфейсом пользователя. Реализовать это можно на языке программирования PHP.

Важным преимуществом данного языка программирования перед такими языками, как Perl и C заключается в возможности создания HTML документов с внедренными командами PHP [8]. Также преимуществами данного языка программирования являются высокая скорость работы, бюджетность и гибкость.

Генератор конфигураций для автоматического создания конфигураций по готовому шаблону был разработан на языке программирования PHP. Генератор состоит из трех компонентов: `app`, `templates` и `views`. `App` является функциональной частью приложения. `Views` отвечает за отображение и ввод пользователем данных. `Templates` является шаблоном конфигурации оборудования.

Выполнение программы начинается с файла `index.php`. Сначала проводится регистрация собственного автозагрузчика в стеке автозагрузки, загружаются все классы, находящиеся по пути `"/src/app"`.

Далее, читаются необработанные данные из тела запроса. Если это JSON-запрос, и у него есть поле "action", которые принимают значения "check_file" или "save_file", то, выполняются, соответственно, проверка файла на наличие или сохранение по пути "/generated". Сохранение файла происходит в формате JSON. Иначе, создается объект класса FormGenerator, после чего вычисляется - какой именно шаг программы нужно показать в данный момент. Всего шагов 3.

Для шага 1 берется обычный html-файл, расположенный по пути "/views/step1.html", а далее генерируются опции для селекта (исходя из имен файлов, расположенных по пути "/templates" (первые два файла исключены)). Далее в форме просто заменяются опции у пустого селекта, и возвращается модифицированный файл "/views/step1.html".

На шаг 2 мы переходим из шага 1, путем отправки формы по методу POST. В POST-параметрах мы имеем имя того шаблона, над которым будем работать. Далее, получаем просто html-файл, расположенный по пути "/views/step2.html" (основа для отображаемой страницы), получаем шаблон (вычисляем исходя из параметра в POST и названиях файлов в директории /templates/*).

Далее мы производим парсинг шаблона на предмет нужных нам полей. Html-код полей находится по адресу "/views/fields", среди которых есть 3 нужных нам типа - option (находится внутри select), сам select, и text. Select и option - простые html формы ввода, а у типа text есть скрытые поля (type="hidden", полей с одним и тем же названием может быть несколько, все складывается в массив, который мы будем разбирать на следующем шаге). Имя поля, сгенерированного таким образом также является членом массива вида имя[идентификатор_группы]. Далее происходит группировка таких полей по идентификатору группы.

Важно заметить, что в шаблоне, расположенном по адресу /templates/* присутствуют циклы, которые начинаются с {%for <целое число>%} и заканчиваются {%endfor%}.

Далее генерируется html-код из нужных нам полей, на данном этапе, и им заполняется изначальный файл "/views/step2.html".

На основании введенных данных и полученных плейсхолдеров (и полученных циклов), мы рассчитываем итоговые значения - исходя из количества циклов и нужных нам IP-адресов, которые нам и нужны, в цикле, опять же, заменяем значения на нужные, и предоставляем итоговый результат, равно как и форму для скачивания (и сохранения) итоговых файлов.

На рисунке 4 отображен принцип генерации конфигураций. Пример файла template находится в приложении А.

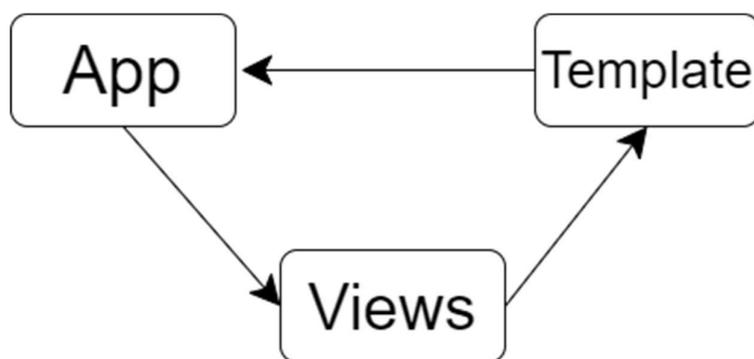


Рисунок 4 – Принцип генерации конфигураций

3.2 Инструкция пользователя

Пользователь должен получить у системного администратора ссылку для перехода в систему. После перехода по ссылке, пользователю предлагается выбрать шаблон для создания конфигурации. Шаблоны в систему загружает системный администратор. После выбора шаблона, пользователь должен ввести входные данные для генерации конфигурации. Система сделает необходимые действия и откроется страница с готовой конфигурацией. На данной странице есть возможность скопировать текст конфигурации, скачать его и сохранить в систему для дальнейшего использования.

3.3 Инструкция системного администратора

Для запуска сервера на операционной системе Windows необходимо осуществить следующие действия:

1. Загрузить и установить сервер Apache;
2. Загрузить пакет установки для версии PHP 5;
3. Запустить установочный файл .msi, запустится мастер установки;
4. Следовать инструкциям на экране для установки;
5. После завершения установки перезапустить сервер Apache;

После выполнения всех пунктов необходимо поместить в папку htdocs данное приложение и в браузере перейти по адресу <http://localhost/index.php>

Допускается использование серверных платформ, например, Open Server.

Создание шаблонов осуществляется следующим образом:

В любом текстовом редакторе необходимо создать файл, имя которого начинается на «tmpl_», например, «tmpl_example.txt». Содержимое файла должно содержать все необходимые команды в нужной последовательности для настройки оборудования. Пример готового шаблона находится в приложении А.

Для того, чтобы использовать циклы, достаточно использовать ключевые фразы `for` и `endfor`, заключенные в фигурные скобки и знак `%`. (`{%for 5%}` и `{%endfor%}`). Для использования таких функций как изменение IP адреса, достаточно прописать имя необходимой функции и передать все необходимые параметры.

Для добавления новой переменной, необходимо в файл `config.yml` добавить имя новой переменной, ее тип и метку, которая будет отображаться на том месте, где необходимо вводить данные пользователю.

Сохраненные в систему конфигурации находятся в папке `generated`, которая находится в корневой папке проекта.

3.3 Вывод по главе

- 1) Разработано описание программы.
- 2) Разработаны инструкции для обычного пользователя и системного администратора.

3 Описание выполненных работ

3.1 Разработка топологии кольцо

Кольцевая структура представляет собой конфигурацию, в которой каждый узел соединяется ровно с двумя другими узлами, образуя единый непрерывный путь для сигналов через каждый компьютер- кольцо данные также передаются последовательно от компьютера к компьютеру, но по сравнению с простым последовательным соединением данные могут передаваться в двух направлениях, что повышает устойчивость к неполадкам сети. Один разрыв не выводит сеть из строя, но два разрыва делают сеть нерабочей. Кольцевая сеть достаточно широко применяется, в основном из-за высокой скорости передачи данных. Кольцевые сети самые скоростные [3]. На рисунке 5 отображена структура кольцевой топологии.

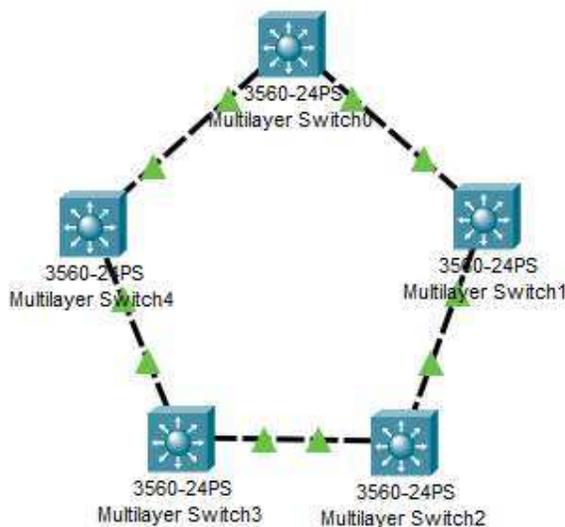


Рисунок 5 – Топология кольцо

Для настройки маршрутизации данной структуры можно использовать такие протоколы, как OSPF и HSRP.

OSPF- протокол динамической маршрутизации, основанный на технологии отслеживания канала и использующий для нахождения кратчайшего пути алгоритм Дейкстры.

HSRP- проприетарный протокол Cisco, предназначенный для увеличения доступности маршрутизаторов выполняющих роль шлюза по

умолчанию. Это достигается путём объединения маршрутизаторов в standby группу и назначения им общего IP-адреса, который и будет использоваться как шлюз по умолчанию для компьютеров в сети. [4]

Для реализации топологии кольцо была спроектирована схема в Cisco Packet Tracer. Конфигурации для всех устройств ядра сети идентичны между собой, меняются только IP-адреса и другие данные. В приложении Б находится пример готовой конфигурации одного из устройств. Для реализации конфигурации всех устройств кольцевой структуры сети необходимо создать шаблон конфигурации для одного устройства, меняя входные параметры в соответствии с заданием.

3.2 Разработка шаблонов конфигураций

В ходе разработки программного обеспечения были разработаны шаблоны конфигураций для сетевого оборудования. Все шаблоны были протестированы в симуляторе Cisco Packet Tracer и на лабораторном стенде. Данные шаблоны были созданы для настройки ядра сети с использованием таких протоколов, как HSRP и OSPF.

На рисунке 6 отображены входные данные для одного из шаблонов. На рисунке 7 отображен результат выполнения шаблона, который находится на рисунке 6.

Vlan

IP Адрес

Маска подсети

Количество узлов в ядре

HELPER_IP

port

Рисунок 6 – Генератор конфигураций, пользовательский ввод

Скачать	Имя файла	Сохранить файл
---------	-----------	----------------

```
-----
switch_0
-----
enable
configure terminal
vlan 5
name vlan5
exit
interface vlan5
ip address 10.100.0.1 255.255.255.240
exit
vlan 10
name vlan10
exit
interface vlan10
ip address 10.200.0.1 255.255.255.240
standby 10 ip 10.200.0.14
standby 10 priority 120
exit
vlan 14
name vlan14
exit
interface vlan14
ip address 10.200.0.66 255.255.255.240
standby 14 ip 10.200.0.78
exit
interface fastEthernet0/15
switchport trunk encapsulation dot1q
switchport mode trunk
exit
interface fastEthernet0/16
switchport trunk encapsulation dot1q
switchport mode trunk
exit
interface fastEthernet0/19
switchport trunk encapsulation dot1q
switchport mode trunk
exit
spanning-tree vlan 5 root primary
spanning-tree vlan 10 root primary
spanning-tree vlan 14 root secondary
-----
switch_1
-----
enable
configure terminal
vlan 5
name vlan5
```

Рисунок 7 – Результат генерации конфигурации

3.3 Компоновка и настройка сервера приложения.

Для тестирования программы было принято решение использовать Open Server. Это портативный локальный сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов. Для

отладки скриптов в различном окружении, Open Server предлагает на выбор сразу два вида HTTP серверов, различные виды систем управления базами данных, а также возможность быстрого переключения между ними [7].

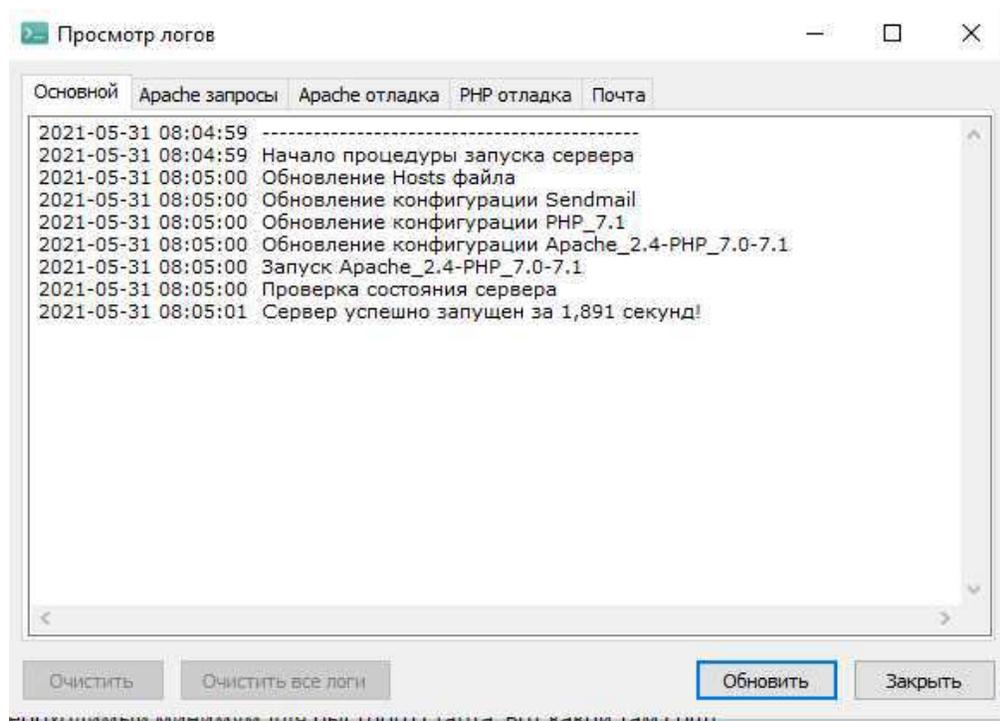


Рисунок 8 – Open Server

Для запуска проекта достаточно скопировать решение в папку с проектами и запустить сервер. Больше никаких действий от пользователя не требуется, можно открыть в браузере запущенный проект.

3.4 Вывод по главе

- 1) Разработан шаблон для генерации конфигурации кольцевой структуры ядра сети.
- 2) Разработаны шаблоны для создания конфигураций сетевого оборудования, включая шаблоны для кольцевой структуры ядра сети.

3) Приложение было запущено на локальном сервере Open Server, проблем при использовании приложения на данном сервере не обнаружено.

ЗАКЛЮЧЕНИЕ

В результате данной работы был произведен анализ существующих аналогов генератора конфигураций, сформулированы требования и поставлены задачи проектирования. После чего было разработано описание программы и созданы инструкции для пользователей. Затем были разработаны различные шаблоны для использования данной программы в целях настройки сетевого оборудования, доработано приложение, которое на основе шаблонов конфигураций создает готовую конфигурацию для настройки сетевого оборудования, и был запущен локальный сервер с данным приложением.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мэтт, З. PHP. Объекты, шаблоны и методики программирования / З. Мэтт. — Москва : Вильямс, 2011.
2. Стаффер, М. Laravel. Полное руководство / М. Стаффер. — Санкт-Петербург : Питер, 2020.
3. Беляев, М. А. Основы информатики / М. А. Беляев, В. В. Лысенко, Л. А. Малинина. — Ростов-на-Дону: Феникс, 2006. — 352 с.
4. HSRP [электронный ресурс] // Фрижурнал — Режим доступа: <https://ru.freejournal.org/211630/1/hsrp.html> (дата обращения: 30.05.2021)
5. Генератор конфигураций для сетевого оборудования и не только [электронный ресурс] // Хабр — Режим доступа: <https://habr.com/ru/post/307054/> (дата обращения: 01.05.2021)
6. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. — Введ. 09.01.2014. — Красноярск: ИПК СФУ, 2014. — 60с.
7. Архипов, М. Open Server — профессиональный инструмент веб-разработчика под Windows / М. Архипов. — [электронный ресурс] // Хабр : — Режим доступа: <https://habr.com/ru/post/137388/> (дата обращения: 11.05.2021).
8. Что такое PHP?. — [электронный ресурс] // PHP.SU: — Режим доступа: <http://www.php.su/php/?php> (дата обращения: 11.05.2021).

ПРИЛОЖЕНИЕ А

Шаблон кольцевой структуры ядра сети

```
1  enable
2  configure terminal
3  vlan {{Vlan|Vlan}}
4  name vlan{{Vlan|Vlan}}
5  exit
6  {%for 3%}
7  interface vlan{{Vlan|Vlan}}
8  ip address {{IP_ADDR|IP_addr}} {{MASK|Маска}}
9  exit
10 {%endfor%}
11 {%for 3%}
12 interface fastEthernet0/{%calc(port,i)%}
13 switchport trunk encapsulation dot1q
14 switchport mode trunk
15 exit
16 {%endfor%}
17 spanning-tree vlan 14 root primary
18 spanning-tree vlan 13 root secondary
```

ПРИЛОЖЕНИЕ Б

Конфигурация для настройки коммутатора в кольцевой топологии

```
1 enable
2 configure terminal
3 vlan 5
4 name vlan5
5 exit
6 interface vlan5
7 ip address 10.100.0.2 255.255.255.240
8 exit
9 vlan 10
10 name vlan10
11 exit
12 interface vlan10
13 ip address 10.200.0.2 255.255.255.240
14 standby 10 ip 10.200.0.14
15 exit
16 vlan 11
17 name vlan11
18 exit
19 interface vlan11
20 ip address 10.200.0.17 255.255.255.240
21 standby 11 ip 10.200.0.30
22 standby 11 priority 120
23 exit
24 interface fastEthernet0/15
25 switchport trunk encapsulation dot1q
26 switchport mode trunk
27 exit
28 interface fastEthernet0/16
29 switchport trunk encapsulation dot1q
30 switchport mode trunk
31 exit
32 interface fastEthernet0/19
33 switchport trunk encapsulation dot1q
34 switchport mode trunk
35 exit
36 spanning-tree vlan 11 root primary
37 spanning-tree vlan 5 root secondary
38 spanning-tree vlan 10 root secondary
```

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой



О.В. Непомнящий

подпись

инициалы, фамилия

« 08 » 06 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование направления

Генератор конфигураций для кольцевой структуры ядра сети

тема

Руководитель



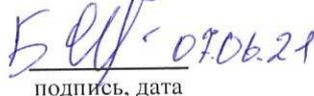
подпись, дата

ст. преподаватель
должность, ученая степень

О.В.Шмелев

инициалы, фамилия

Выпускник



подпись, дата

С.В.Брюханов

инициалы, фамилия

Нормоконтролер



подпись, дата

ст. преподаватель
должность, ученая степень

О.В.Шмелев

инициалы, фамилия

Красноярск 2021