

Министерство образования и науки Российской Федерации  
ФГАОУ ВПО «УрФУ имени Президента России Б.Н. Ельцина»  
Институт радиоэлектроники и информационных технологий  
Кафедра информационных технологий

Оценка работы \_\_\_\_\_  
Руководитель от УрФУ \_\_\_\_\_

РАЗРАБОТКА ПО ДЛЯ АВТОМАТИЗАЦИИ КОНФИГУРИРОВАНИЯ  
СЕТЕВОГО ОБОРУДОВАНИЯ CISCO  
Отчет по преддипломной практике

Руководители от предприятия

Трофимов С.П.

Студент

Черетаев И.В.

Специальность

Информатика и вычислительная  
техника

Группа

РИ-420207

Екатеринбург, 2016

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ . . . . .	4
1. Техническое задание . . . . .	8
1.1 Общие положения . . . . .	8
1.1.1 Наименование программы . . . . .	8
1.1.2 Краткая характеристика области применения . . . . .	8
1.1.3 Основание для проведения разработки . . . . .	8
1.2 Назначение разработки . . . . .	9
1.2.1 Функциональное назначение . . . . .	9
1.2.2 Эксплуатационное назначение . . . . .	9
1.3 Требования к программному средству . . . . .	9
1.3.1 Требования к функциональным характеристикам . . . . .	9
1.3.2 Требования к надежности функционирования и без- опасности . . . . .	10
1.3.3 Требования к составу и параметрам технических средств	10
1.3.4 Требования к информационной и программной совме- стимости . . . . .	10
1.3.5 Требования к исходным кодам и языкам программи- рования . . . . .	10
1.3.6 Специальные требования . . . . .	10
1.4 Требования к программной докуменации . . . . .	11
1.4.1 Предварительный состав программной документации . . . . .	11
1.5 Стадии и этапы разработки . . . . .	11
1.5.1 Стадии разработки . . . . .	11
1.5.2 Этапы разработки . . . . .	11
1.5.3 Содержание работ по этапам . . . . .	11
1.6 Порядок защиты и контроля . . . . .	12
2. Выбор инструментальной системы . . . . .	13
3. Обзор существующих решений . . . . .	15
3.1 Cisco Packet Tracer . . . . .	15
3.2 GNS3 . . . . .	16
3.3 Boson NetSim . . . . .	18
3.4 Cisco IOU . . . . .	19

ЗАКЛЮЧЕНИЕ . . . . .	22
Список использованной литературы . . . . .	22

## ВВЕДЕНИЕ

На заре развития информационной эры (а именно – до 1969 года, который принято считать «днем рождения» компьютерной сети) мало кто мог представить, что группа персональных компьютеров способна стать единой высокотехнологичной системой, ориентированной на решение самого широкого спектра задач. Сегодня, спустя полвека, даже дети знают, что такое компьютерная сеть, и, тем не менее, настройка сетевого оборудования – это достаточно трудоемкий процесс, который не терпит халатного отношения.

При этом, говоря о настройке сетевого оборудования, мы подразумеваем обширный комплекс точных методик, позволяющих обеспечить безупречную оптимизацию системы, которая в свою очередь ведет к расширению коммуникативных и аналитических возможностей всей сети. Важным аспектом данного вопроса является технический прогресс, который постоянно обеспечивает нас все новыми и новыми возможностями, но накладывает на технических специалистов важные требования, в частности – необходимость повышенной адаптивности. Однако для специалистов настройка сетевого оборудования представляет одно из ключевых направлений, где качество является единственно верным приоритетом.

Здесь очень важно понимать суть используемых формулировок. В частности, настройка активного сетевого оборудования представляет собой специфический комплекс мер, который существенно отличается от методик, применяемых в процессе настройки так называемого пассивного оборудования. Подробно рассматривая данный вопрос, необходимо остановиться на следующих нюансах:

- в действительности, на сегодняшний день не существует общепринятого определения для пассивного сетевого оборудования. Чаще всего здесь подразумеваются компоненты, необходимые для формирования СКС (структурированной кабельной системы), которая, образно выражаясь, является «каркасом» любой компьютерной сети. Иными словами, настройка сетевого оборудования такого плана подразумевает работу с кабелями, кабельными лотками и каналами, розетками,

коннекторами, патч-кордами. Фактически пассивное сетевое оборудование – это технические средства, которые обеспечивают передачу сигнала внутри сети, это сами проводники, а также средства их организации и защиты. Логично, что настройка активного сетевого оборудования представляет собой принципиально иной вид деятельности, основанный на конструктивных отличиях оборудования;

- по своей сути активное сетевое оборудование от пассивного отличается одной важной чертой – потребление электрической энергии. Иными словами, «активка» – это интеллектуальные сетевые компоненты, собственно маршрутизаторы, свитчи и так далее, все, что позволяет компьютерам, серверам и рабочим станциям формировать единое информационное пространство. Именно поэтому настройка активного сетевого оборудования подразумевает более тонкий подход. И если, скажем, прокладка сетевого кабеля включает в себя четкую последовательность действий, которая кроме прочего решает также эстетическую задачу, то, к примеру, настройка маршрутизатора может иметь огромное количество нюансов, в зависимости от особенностей будущей сети и включенных в нее технических средств (собственно персональных компьютеров и серверов).

Одним из представителей производителей сетевого оборудования является компания Cisco Systems.

Cisco Systems — компания-производитель сетевого оборудования (основана в 1984) и краткое название для оборудования, которое он производит. Название Cisco изначально писалось с маленькой буквы (cisco); оно произошло от названия города Сан-Франциско, откуда компания родом.

Сначала компания производила маршрутизаторы, но впоследствии значительно расширила ассортимент своей продукции. Сейчас она производит коммутаторы, маршрутизаторы, IP-телефоны, программное обеспечение и многое-многое другое.

Компания известна своей развитой системой сертификации специалистов, которая признаётся и уважается почти во всём мире. Это одна из немногих IT-компаний, которым удалось возвести сертификацию специалистов практически в ранг культа.

В настоящее время существует 5 ступеней сертификации:

- Entry (CCENT), самая слабая ступень;
- Associate (CCNA), минимальная ступень, где требуются хоть какие-то технические знания;
- Professional (CCNP), более-менее нормальный уровень инженера;
- Expert (CCIE), достойный уровень знаний, требующий очень хорошего уровня подготовки; именно это звание было (и остаётся) культовым для Cisco-инженеров. Только 3 процента сертифицированных инженеров Cisco доходят до этого уровня;
- Architect, новый уровень, который должен быть ещё круче чем CCIE. По состоянию на ноябрь 2015 года, в мире есть всего 8 человек, которые могут подтвердить это или опровергнуть.

Начинающим сетевым инженерам может быть полезно начать изучение сетей именно с Cisco, потому что:

- 1) Это один из самых распространенных производителей сетевого оборудования.
- 2) В Интернете есть большое количество как учебных материалов, так и просто советов по настройке и другой документации.
- 3) Есть разные эмуляторы и симуляторы оборудования Cisco, которые позволяют собрать лабораторный стенд для практических задач.
- 4) Многие передовые технологии в первую очередь приходят в мейнстрим именно у Cisco.
- 5) Некоторые производители оборудования заимствуют синтаксис командного интерфейса Cisco, или стараются делать командный интерфейс похожим на интерфейс Cisco. Этот факт облегчает переход на другое оборудование с оборудования Cisco, в том случае если он потребуется.

Существует достаточно большое количество IT специалистов, сетевых администраторов, которым необходимо просто быстро запустить небольшой проект на оборудовании Cisco собственными силами, не привлекая к этому дорогостоящих специалистов Cisco.

Они не собираются посвящать остаток жизни изучению всех тонкостей сетевых технологий, маршрутизации, информационной безопасности, беспроводной связи и т.д. Им просто требуется работающее, достаточно надежное, понятное и контролируемое решение, например, доступ в Ин-

тернет для небольшого офиса. Или защищенный доступ к офисной сети из дома через Интернет (VPN). Или же, к примеру, бюджетно оборудовать офис беспроводной связью Wi-Fi. Примеров может быть много, просто это наиболее типовые запросы на сегодняшний день.

Соответственно, необходимо решение, которое быстро и легко поможет настроить оборудование Cisco и решить данные проблемы.

# 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## 1.1. Общие положения

### 1.1.1. Наименование программы

Программное обеспечение для автоматизации конфигурирования сетевого оборудования Cisco. Условное обозначение – АКСО[7].

### 1.1.2. Краткая характеристика области применения

АКСО предназначено для частичной или полной автоматизации процесса изменения исходного текста с целью получения отличного от первоначального стиля изложения.

### 1.1.3. Основание для проведения разработки

Перечень документов, на основании которых ведется разработка надстройки:

- 1) Приказ ректора УрФУ № \_\_\_\_\_ от "\_\_\_" \_\_\_\_\_ г.

Порядок оформления и предъявления результатов проектирования устанавливается согласно документам:

- 1) Методические указания по выполнению выпускной квалификационной работы бакалавра техники и технологий по направлению "Информатика и вычислительная техника"/ сост. А.Б.Николаев. – Москва: МАДИ, 2010. – 17 с.
- 2) Методические указания к выполнению выпускных бакалаврских работ по направлению 552800 «Информатика и вычислительная техника» по специальности 230101 65 «Вычислительные машины, комплексы, системы, сети»/Под общей ред.В.А.Бархоткина. - Москва:МИЭТ, 2006. - 15с.
- 3) Соколов С.С. Рекомендации по оформлению курсовых, выпускных и дипломных проектов (работ). Электронные методические указания - Екатеринбург: Изд. УрФУ, 2010. - 38 с.



## 1.2. Назначение разработки

### 1.2.1. Функциональное назначение

Функциональным предназначением АКСО является предоставлению пользователю удобных инструментов для облегчения конфигурирования сетевого оборудования от компании Cisco Systems, таких как маршрутизаторы и/или коммутаторы.

### 1.2.2. Эксплуатационное назначение

АКСО предназначено для автоматизации процесса конфигурирования сетевого оборудования с целью получения исходного текста конфигурации либо автоматического применения внесенных с помощью АКСО изменений на сетевом оборудовании. Пользователями АКСО могут являться системные администраторы, сетевые инженеры, студенты.

## 1.3. Требования к программному средству

### 1.3.1. Требования к функциональным характеристикам

- 1) Разработать структуру классов, хранящую информацию о сетевых устройствах.
- 2) Ввести основные модели маршрутизаторов и коммутаторов.
- 3) Разработать редактор, позволяющий добавлять новые модели (в том числе, на основе уже существующих моделей), изменять и удалять существующие. Редактор должен обеспечивать возможность поиска в списке имеющихся моделей. Так же при помощи данного редактора должны производиться операции визуального изменения конфигурации для выбранного модели сетевого оборудования.
- 4) Разработать систему для экспорта/импорта выбранной конфигурации в виде XML-файла, содержащего все необходимые сведения.
- 5) Разработать систему для сохранения выбранной конфигурации в виде текстового файла, содержащего команды конфигурирования
- 6) Разработать справочную подсистему. Справка должна содержать краткую информацию о системе и ее возможностях, описание действий пользователя и получаемых результатов при работе с программным обеспечением.

7) Реализовать перечисленные выше функции в рамках приложения для настольного компьютера с графическим интерфейсом.

#### 1.3.2. Требования к надежности функционирования и безопасности

Надёжность системы должна обеспечивать работоспособность в течение всего срока эксплуатации при бесперебойном питании ЭВМ. Нарботка на отказ при эксплуатации программного средства должна составлять не менее 8 часов. Программное обеспечение не должно содержать явных логических ошибок и функционировать без сбоев.

В течение срока эксплуатации необходимо выполнение требований «ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов».

#### 1.3.3. Требования к составу и параметрам технических средств

В состав технических средств должна входить ЭВМ, включающая в себя:

- процессор с тактовой частотой не менее 1 ГГц;
- ОЗУ не менее 512 МБ;
- экран с разрешением не менее 1024x768 точек;
- клавиатура;
- манипулятор «мышь».

#### 1.3.4. Требования к информационной и программной совместимости

АКСО должна иметь возможность функционировать под управлением различных операционных систем (Windows, Linux и т.д.).

#### 1.3.5. Требования к исходным кодам и языкам программирования

В ходе прохождения преддипломной практики необходимо выбрать на каком языке программирования будет написан исходный код программного средства, а также интегрированную среду разработки.

#### 1.3.6. Специальные требования

- Программа должна обеспечивать взаимодействие с пользователем (оператором) посредством графического пользовательского интерфейса.

## 1.4. Требования к программной докуменации

### 1.4.1. Предварительный состав программной документации

Предварительный состав программной документации должен включать в себя[5, 6]:

- 1) техническое задание;
- 2) текст программы;
- 3) описание программы;
- 4) программу и методики испытаний;
- 5) пояснительную записку[1–3];
- 6) описание применения;
- 7) руководство пользователя.

## 1.5. Стадии и этапы разработки

### 1.5.1. Стадии разработки

Разработка должна быть произведена в три стадии[4]:

- 1) Разработка технического задания;
- 2) Рабочее проектирование;
- 3) Внедрение;

### 1.5.2. Этапы разработки

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

- 1) разработка АКСО;
- 2) разработка программной документации;
- 3) испытания АКСО.

На стадии внедрения должен быть выполнен этап разработки - подготовка АКСО.

### 1.5.3. Содержание работ по этапам

На этапе разработки АКСО должна быть выполнена работа по программированию (кодированию) и отладке программного обеспечения (АКСО).

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованием

п. 1.4.1 настоящего технического задания.

На этапе испытаний АКСО должны быть выполнены перечисленные ниже виды работ:

- 1) проверка выполнения заданных функций АКСО;
- 2) выявления и устранения недостатков в АКСО и программной документации;
- 3) корректировка АКСО и программной документации по результатам тестирований[8].

На этапе подготовки АКСО должна быть выполнена работа по подготовке программного средства и программной документации для эксплуатации.

#### 1.6. Порядок защиты и контроля

Защита осуществляется перед Государственной аттестационной комиссией (ГАК), утвержденной приказом ректора.

## 2. ВЫБОР ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ

Приложения для настольных компьютеров в значительной степени зависят от платформы, на которой они работают. Приложение, скомпилированное для ОС Linux, не может работать в другой операционной системе без использования эмуляторов по причине различий в системных вызовах и библиотеках в этих операционных системах. Необходимо было найти возможность разработки платформонезависимых приложений.

Одним из методов является разработка версий библиотек для всех необходимых платформ, не изменяя код приложения, а просто перекомпилируя его для каждой платформы. Этот подход облегчал жизнь разработчикам до того момента, как компания Sun Microsystems представила революционную концепцию виртуальных машин, подарив миру платформонезависимый язык программирования Java. Приложения Java исполняются при помощи виртуальной машины Java (Java Virtual Machine). Код, написанный однажды, может быть развернут на любой платформе, для которой доступна виртуальная машина Java.

Но за все приходится платить. В случае с языком Java приходится расплачиваться производительностью. Производительность приложений на Java не настолько высока, как у приложений на C/C++, скомпилированных в машинный код. Другим недостатком является большое потребление памяти этими приложениями. Без сомнения, язык Java занимает лидирующие позиции, но на сегодняшний день существуют приложения, обрабатывающие большие объемы данных (например, в области биотехнологии) с особо жесткими требованиями к производительности. И снова мы сталкиваемся с необходимостью разработки приложений, компилируемых в машинный код. Разработка приложений на языке C занимает много времени, в то время как C++ является лучшим решением. Имеется ряд фреймворков, предназначенных для использования при разработке приложений на C++. Фреймворк Qt является одним из них.

Qt позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путём простой компиляции программы для каждой ОС без изменения исходного кода. Включает в себя все ос-

новые классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Отличительная особенность Qt от других библиотек — использование Meta Object Compiler (МОС) — предварительной системы обработки исходного кода. МОС позволяет во много раз увеличить мощь библиотек, вводя такие понятия, как слоты и сигналы. Кроме того, это позволяет сделать код более лаконичным. Утилита МОС ищет в заголовочных файлах на C++ описания классов, содержащие макрос Q\_OBJECT, и создаёт дополнительный исходный файл на C++, содержащий метаобъектный код.

Qt позволяет создавать собственные плагины и размещать их непосредственно в панели визуального редактора. Также существует возможность расширения привычной функциональности виджетов, связанной с размещением их на экране, отображением, перерисовкой при изменении размеров окна.

Qt комплектуется визуальной средой разработки графического интерфейса «Qt Designer», позволяющей создавать диалоги и формы в режиме WYSIWYG. В поставке Qt есть «Qt Linguist» — графическая утилита, позволяющая упростить локализацию и перевод программы на многие языки; и «Qt Assistant» — справочная система Qt, упрощающая работу с документацией по библиотеке, а также позволяющая создавать кросс-платформенную справку для разрабатываемого на основе Qt ПО. Начиная с версии 4.5.0 в комплект Qt включена среда разработки «Qt Creator», которая включает в себя редактор кода, справку, графические средства «Qt Designer» и возможность отладки приложений. «Qt Creator» может использовать GCC или Microsoft VC++ в качестве компилятора и GDB в качестве отладчика. Для Windows версий библиотека комплектуется компилятором, заголовочными и объектными файлами MinGW.

### 3. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Существует достаточно большое количество симуляторов и эмуляторов для оборудования Cisco Systems. В ходе показать все существующие инструменты, которые решают эту задачу.

Вначале немного терминологии.

Симуляторы — имитируют некий набор команд, он вшит и стоит только выйти за рамки, сразу получим сообщение об ошибке. Классический пример — Cisco Packet Tracer.

Эмуляторы же напротив — позволяют проигрывать (выполняя байт трансляцию) образы (прошивки) реальных устройств, зачастую без видимых ограничений. В качестве примера — GNS3/Dynamips.

#### 3.1. Cisco Packet Tracer

Packet Tracer — симулятор сети передачи данных, выпускаемый фирмой Cisco Systems. Позволяет делать работоспособные модели сети, настраивать (командами Cisco IOS) маршрутизаторы и коммутаторы, взаимодействовать между несколькими пользователями (через облако). В симуляторе реализованы серии маршрутизаторов Cisco 800, 1800, 1900, 2600, 2800, 2900 и коммутаторов Cisco Catalyst 2950, 2960, 3560, а так же межсетевой экран ASA 5505. Беспроводные устройства представлены маршрутизатором Linksys WRT300N, точками доступа и сотовыми вышками. Кроме того есть серверы DHCP, HTTP, TFTP, FTP, DNS, AAA, SYSLOG, NTP и EMAIL, рабочие станции, различные модули к компьютерам и маршрутизаторам, IP-фоны, смартфоны, хабы, а так же облако, эмулирующее WAN. Объединять сетевые устройства можно с помощью различных типов кабелей, таких как прямые и обратные патч-корды, оптические и коаксиальные кабели, последовательные кабели и телефонные пары.

Этот симулятор доступен как под Windows, так и для Linux, бесплатно для учащихся Сетевой Академии Cisco.

Его плюсы — дружелюбность и логичность интерфейса. Кроме этого в нем удобно проверять работу разных сетевых сервисов, вроде DHCP/DNS/HTTP/SMTP/POP3 и NTP. И одна из самых интересных фиш — это воз-

возможность перейти в режим simulation и увидеть перемещения пакетов с замедлением времени.

Однако, стоит заметить, что реализованная функциональность устройств ограничена и не предоставляет всех возможностей реального оборудования. Практически всё, что выходит за рамки CCNA, на нем собрать не получится. К примеру, ЕЕМ отсутствует напрочь.

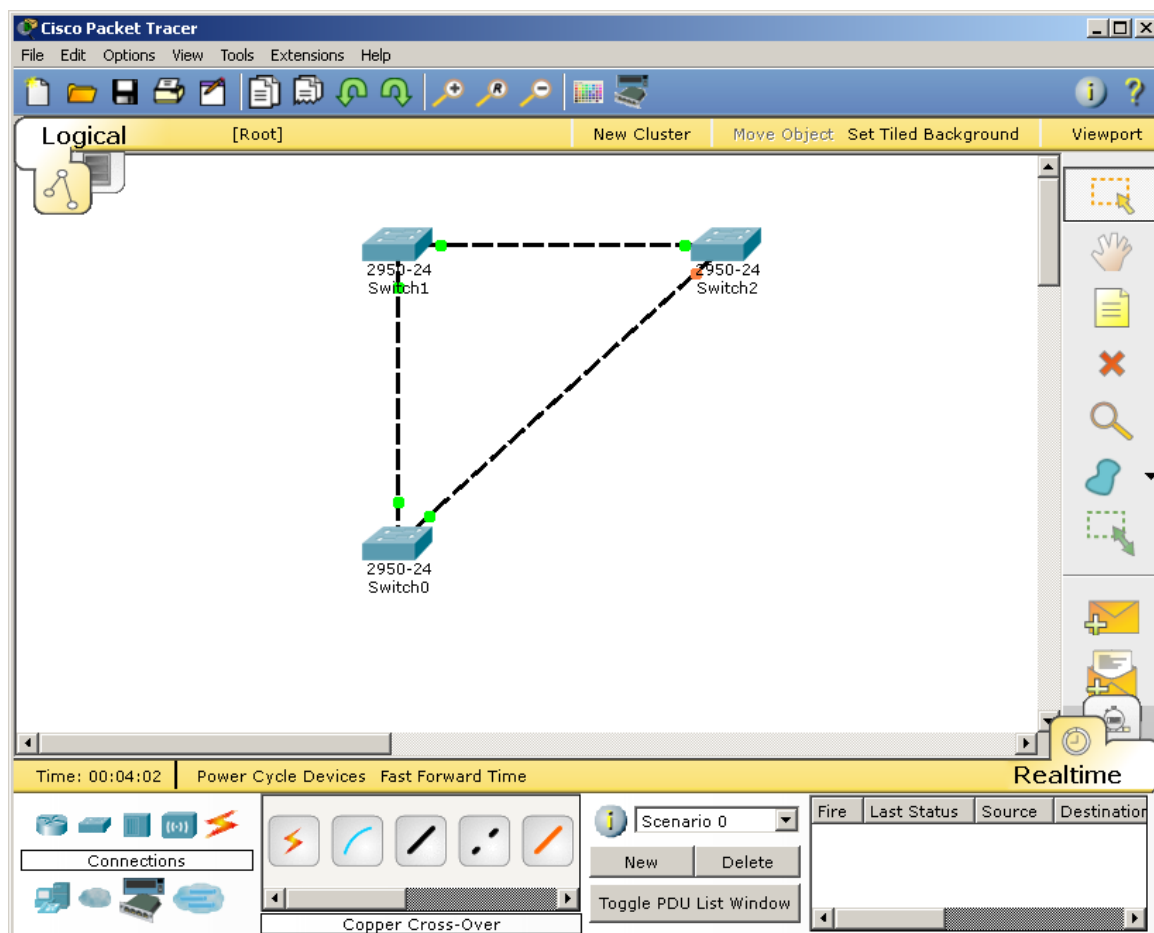


Рисунок 1. Cisco Packet Tracer.

### 3.2. GNS3

Следующий — GNS3, который представляет собой графический интерфейс (на Qt) для эмулятора dynamips.

Свободный проект, доступен под Linux, Windows и Mac OS X.

Сайт проекта GNS — [www.gns3.net/](http://www.gns3.net/)

Но большинство его функций, призванных улучшить производительность, работают только под Linux (ghost IOS, который срабатывает в случае использования множества одинаковых прошивок), 64 битная версия



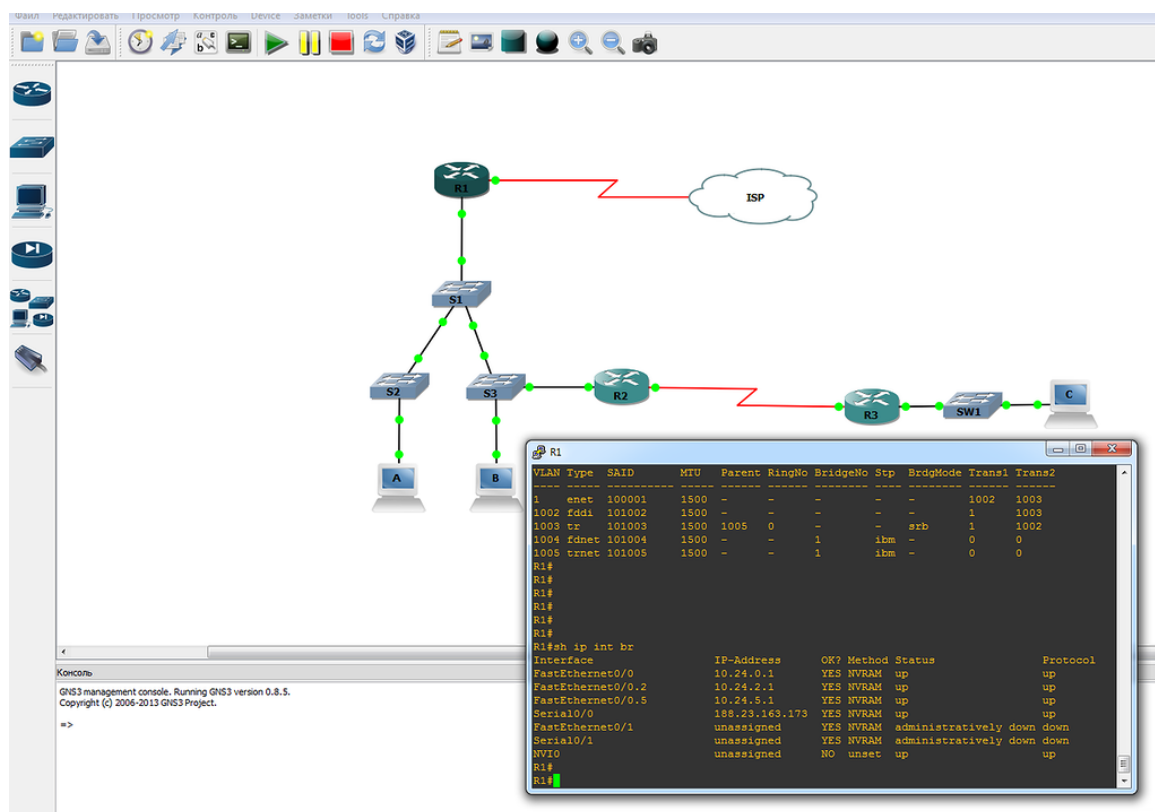


Рисунок 2. GNS3.

так же только для Linux.

Это эмулятор, который работает с настоящими прошивками IOS. Для того чтобы им пользоваться, у вас должны быть прошивки. Скажем, вы купили маршрутизатор Cisco, с него можно их и вытащить. К нему можно подключать виртуальные машины VirtualBox или VMware Workstation и создавать достаточно сложные схемы, при желании можно пойти дальше и выпустить его в реальную сеть. Кроме того, Dynamips умеет эмулировать как старые Cisco PIX, так и небезызвестную Cisco ASA, причем даже версии 8.4.

Но при всем этом есть масса недостатков:

- количество платформ строго ограничено: запустить можно только те шасси, которые предусмотрены разработчиками dynamips;
- запустить ios 15 версии возможно только на платформе 7200;
- невозможно полноценно использовать коммутаторы Catalyst, это связано с тем что на них используется большое количество специфических интегральных схем, которые соответственно крайне сложно эмулировать. Остается использовать сетевые модули (NM) для мар-

шутизаторов;

- при использовании большого количества устройств гарантированно будет наблюдаться проседание производительности.

Что имеем в сухом остатке? — Инструмент, в котором можно создавать достаточно сложные топологии, готовиться к экзаменам уровня CCNP, с некоторыми оговорками.

### 3.3. Boson NetSim

Пару слов о симуляторе Boson NetSim, который недавно обновился до 9й версии.

Выпускается только под Windows, цена колеблется от 179\$ за CCNA и до 349\$ за CCNP.

Представляет собой некий сборник лабораторных работ, сгруппированный по темам экзамена. Как можно наблюдать по скриншотам, интерфейс состоит из нескольких секций: описание задачи, карта сети, в левой части находится список всех лабораторных.

Закончив работу, можно проверить результат и узнать все ли было сделано.

Есть возможность создания собственных топологий, с некоторыми ограничениями.

Основные возможности Boson NetSim:

- Поддерживает 42 маршрутизатора, 6 коммутаторов и 3 других устройства
- Симулирует сетевой трафик с помощью технологии виртуальных пакетов
- Предоставляет два различных стиля просмотра: режим Telnet'а или режим подключения по консоли
- Поддерживает до 200 устройств на одной топологии
- Позволяет создавать свои собственные лаборатории
- Включает в себя лаборатории, которые поддерживают симуляцию SDM
- Включает в себя не-Cisco устройства, такие как TFTP Server, TACACS + и генератор пакетов (это, вероятно, те самые 3 других устройства)

Недостатки у него те же, что и в Packet Tracer.

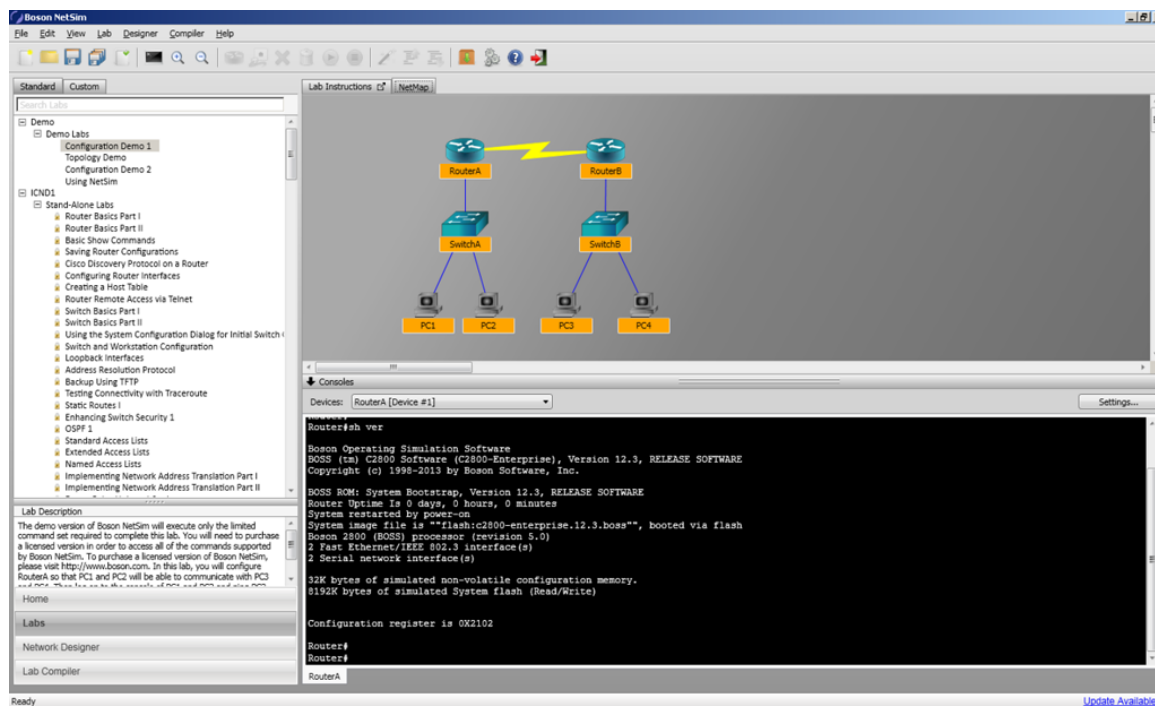


Рисунок 3. Boson NetSim.

Итог? — Тем, кому не жалко определенной суммы, и при этом не хочется разбираться и создавать свои топологии, а хочется просто попрактиковаться перед экзаменом, будет очень кстати.

### 3.4. Cisco IOU

Ну и наконец знаменитый Cisco IOU (Cisco IOS on UNIX) — это проприетарный софт, который официально не распространяется вообще никак.

Существует мнение, что Cisco может отследить и идентифицировать того, кто использует IOU. При запуске происходит попытка HTTP POST запроса на сервер `xml.cisco.com`. Данные, которые при этом отправляются, включают в себя `hostname`, `логин`, версию IOU и т.д.

Известно, что Cisco TAC использует именно IOU. Эмулятор пользуется большой популярностью у тех, кто готовится к сдаче CCIE. Изначально работал только под Solaris, но со временем был портирован и на Linux. Состоит из двух частей — `l2iou` и `l3iou`, по названию можно догадаться, что первый эмулирует канальный уровень и коммутаторы, а второй — сетевой и маршрутизаторы.

Конфигурирование проводится путем редактирования текстовых конфигурационных файлов, но некоторое время назад для него был разработан

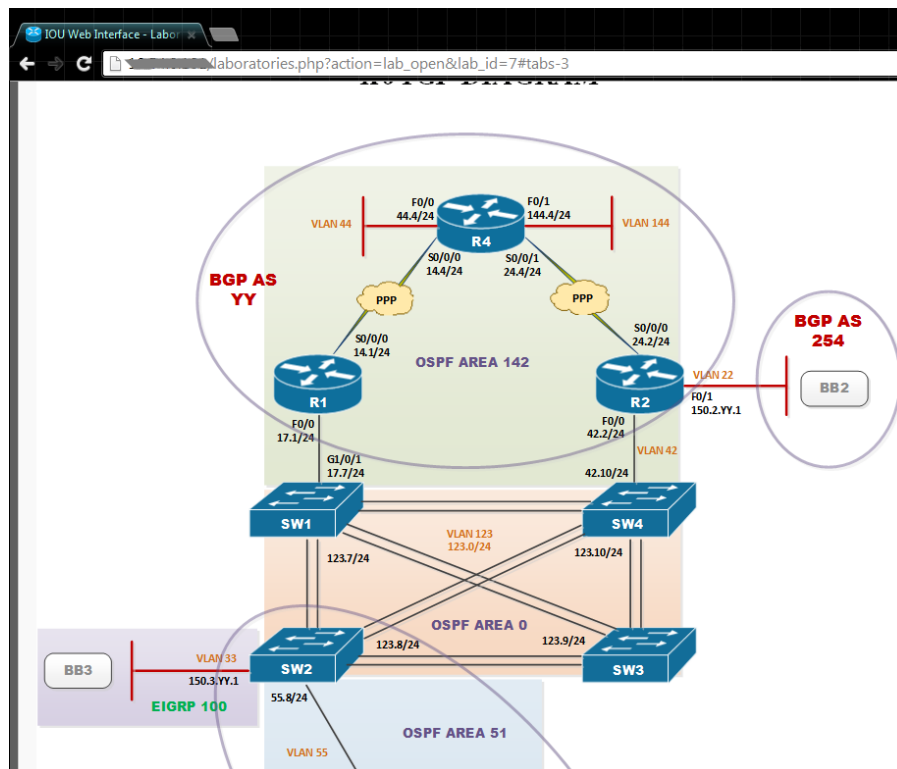


Рисунок 4. Cisco IOU.

и графический интерфейс, веб фронтенд.

Интерфейс достаточно интуитивен, с его помощью можно производить практически все действия.

Запуск вот такой топологии (см. рис. 5) приводит всего лишь к 20% загрузке CPU.

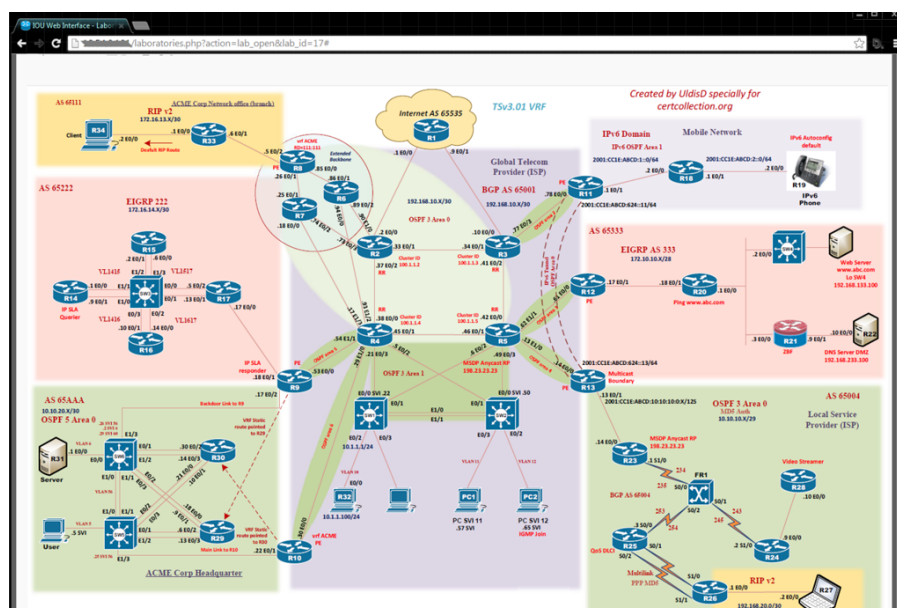


Рисунок 5

К слову, это топология для подготовки к сдаче CCIE.

Для того чтобы подключиться к любому устройству на схеме, достаточно просто кликнуть на нем и сразу же откроется putty.

Возможности IOU действительно очень большие. Хотя и не без недостатков, некоторые проблемы на канальном уровне все же имеются. В некоторых, к примеру, невозможно жестко выставить дуплекс, но это всё мелочи — весь основной функционал работает, и работает отлично.

## ЗАКЛЮЧЕНИЕ

Актуальность темы дипломной работы определяется быстрым ростом количества компаний, имеющих сложную информационную структуру. Наряду с этим возникает необходимость настройки огромного количества активного сетевого оборудования. Все эти факторы приводят к тому, что администраторам информационной инфраструктуры компании приходится тратить много времени на порой рутинные работы.

Цель дипломной работы состоит в разработке системы для автоматизированного создания конфигурационных файлов оборудования.

Поставленная цель обусловила следующие задачи дипломной работы:

- определить сущности проектируемой системы и архитектуру их взаимодействия;
- реализовать систему в виде кроссплатформенного приложения с графическим интерфейсом;
- провести тестированием в лабораторной среде.

Объектом исследования выступает сетевое оборудование Cisco Systems.

Предметом исследования в дипломной работе являются современные средства коммуникации и их интеграция в методы управления информационной инфраструктурой для повышения таких аспектов информационной безопасности предприятия, как целостность и доступность.

Теоретическая значимость дипломного исследования состоит в развитии и совершенствовании методологии управления оборудованием информационной инфраструктуры предприятия.

Практическая значимость работы определяется тем, что ее результаты позволяют повысить степень эффективности управления информационной инфраструктурой предприятия и снизить связанные с этим операционные расходы при использовании разработанной системы, направленной на повышение уровня таких аспектов администрирование, как быстрота и легкость обслуживания.

Новизна дипломной работы заключается в разработке и реализации кроссплатформенной модели автоматизации конфигурирования оборудования.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- [1] В.А. Бархоткин. *Методические указания к выполнению выпускных бакалаврских работ по направлению 552800 «Информатика и вычислительная техника» по специальности 230101 65 «Вычислительные машины, комплексы, системы, сети».* МИЭТ, Москва, 2006.
- [2] С.С. Соколов. *Рекомендации по оформлению курсовых, выпускных и дипломных проектов (работ).* УрФУ, Екатеринбург, 2010.
- [3] А.Б. Николаев. *Методические указания по выполнению выпускной квалификационной работы бакалавра техники и технологий по направлению «Информатика и вычислительная техника».* МАДИ, Москва, 2010.
- [4] ГОСТ 19.102-77 ЕСПД. *Стадии разработки.*
- [5] ГОСТ 19.105-78 ЕСПД. *Общие требования к программным документам.*
- [6] ГОСТ 19.106-78 ЕСПД. *Требования к программным документам, выполненным печатным способом.*
- [7] ГОСТ 19.201-78. *Техническое задание, требования к содержанию и оформлению.*
- [8] ГОСТ 28195-89. *Оценка качества программных средств. Общие положения.*