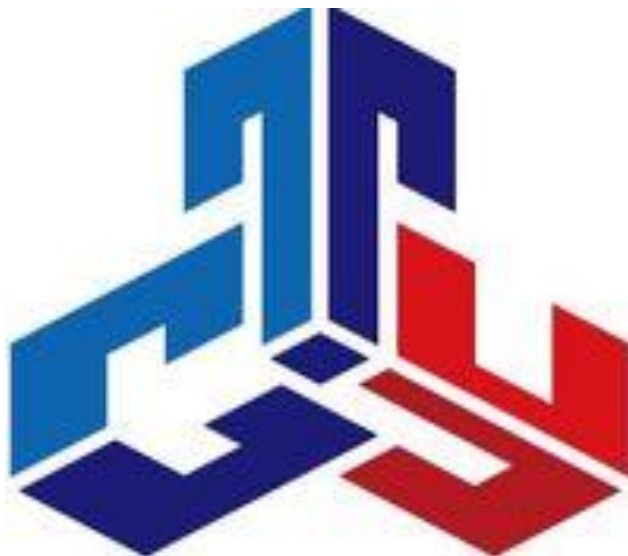


СОВРЕМЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ



СЕТИ ЭВМ И ТЕЛЕКОММУНИКАЦИИ

Методические указания к лабораторным работам

Для студентов направления “Информатика и вычислительная техника”

Рязань 2020 г.

УДК 681.3.06
ББК
С14

Сети ЭВМ и телекоммуникации: методические указания к лабораторным работам / Сост. канд. техн. наук Рыбачек В.П.,
Совр. техн. универ-т. – Рязань, 2020. – 30 с. – 50 экз

Рецензент: доктор технических наук, профессор кафедры САПР Рязанского государственного радиотехнического университета Сускин В.В.

Рассматриваются вопросы использования эмулятора сетевой инфраструктуры GNS3 на базе оборудования фирмы Cisco. Предполагается получение навыков работы с операционной системой Cisco IOS, создания комплексных топологий сетей, настройки протоколов статической и динамической маршрутизации, администрирования маршрутизаторов, анализа внутреннего трафика сетей.

Методические указания предназначены для студентов-бакалавров направления «Информатика и вычислительная техника» при изучении курса «Сети и телекоммуникации» и могут быть полезны специалистам, занимающимся вопросами проектирования локальных вычислительных сетей.

*Печатается по решению Ученого Совета
Современного технического университета.*

УДК 681.3.06
ББК
С14

© В. П. Рыбачек
© Современный технический университет, 2020

ВВЕДЕНИЕ

Компания Cisco Systems является абсолютным лидером на рынке маршрутизаторов (занимает около 70% рынка; на втором месте Juniper с 21%). Cisco предлагает модели от простейших маршрутизаторов для малого офиса (серия 800) до мультигигабитных устройств, размещаемых в ядре Интернета (серия 12000).

Кроме маршрутизаторов Cisco известна коммутаторами ЛВС марки Catalyst, межсетевыми экранами марки PIX, продуктами для IP-телефонии, продуктами марки Aironet для организации беспроводных сетей и др. С учетом всей номенклатуры выпускаемой продукции Cisco Systems является лидером мирового рынка оборудования связи (14%; на втором месте Siemens с 11,7%).

Все маршрутизаторы Cisco работают под управлением операционной системы Cisco IOS. Для каждой модели маршрутизатора предлагаются несколько разновидностей IOS.

Образы IOS различаются по версии. Cisco использует достаточно сложную систему идентификации версий, ознакомиться с которой можно по этой ссылке (<http://www.cisco.com/warp/public/732/abc/releases/>).

Для студентов достаточно будет следующего понимания: номер версии Cisco IOS состоит из трех частей:

- Номер основного релиза (major release; в настоящее время обычно встречаются основные релизы 11.3, 12.0, 12.1, 12.2).
- Номер обновления (maintenance release), начиная с 1. Обновления выпускаются каждые 8 недель, в них включаются исправления ошибок. Набор функциональных возможностей релиза не изменяется.
- Номер выпуска (software rebuild), обозначается буквой, начиная с *a*. Выпуски предназначены для экстренного исправления ошибок, которое не может ждать до следующего обновления.

Таким образом, IOS 12.2(6c) - это основной релиз 12.2, обновление 6, выпуск *c*.

Кроме версий, образы IOS различаются по заложенной в них функциональности. Функциональные возможности группируются в наборы, называемые feature sets. Минимальная функциональность содержится в IP only feature set (или просто "IP"); она включает в себя, в частности, поддержку IP-интерфейсов, статическую и динамическую IP-маршрутизацию, поддержку мониторинга и управления по SNMP. IP Plus feature set включает дополнительные возможности (например, поддержку технологии VoIP для передачи голоса). Также имеются feature sets с функциями межсетевого экрана (FW, Firewall), системы обнаружения атак (IDS), криптозащиты трафика (IPSEC) и др., в том числе имеются и комбинированные образы, например IP Plus FW IPSEC. Бесплатно с маршрутизатором поставляется только IP only feature set, остальные образы необходимо покупать. Для определения того, в каком feature set имеется требуемая вам возможность, следует обратиться к специалисту компании-реселлера.

Таким образом, конкретный образ IOS идентифицируется тремя параметрами:

- аппаратная платформа, для которой он предназначен
- feature set
- версия

Лабораторная работа № 1.

GNS3 - симулятор сетей Cisco. Установка и предварительная настройка симулятора GNS3, создание тестовой топологии.

Цель работы:

Получить базовые навыки работы с эмулятором сетевой инфраструктуры на базе оборудования Cisco.

Задание:

Установить и настроить Graphical Network Simulator, создать тестовую топологию.

Теоретическая часть:

GNS3 это графический инструмент, который позволяет эмулировать комплексные сетевые инфраструктуры на базе оборудования Cisco. В основе проекта, который основан и развивается по принципу ‘open source’ лежат три компонента:

- Dynamips – основная составляющая симулятора, позволяющая эмулировать Cisco IOS
- Dynagen – текстовый интерфейс (front-end) для Dynamips
- Qemu – средство эмуляции и виртуализации различного рода машин (linux хостов, pix/asa, junos устройств)

Чистый Dynamips не очень удобен в плане конфигурирования. Если необходимо создать тестовую среду, состоящую из трех и более маршрутизаторов, то это может оказаться проблемой и отнимет много времени. Для упрощения настройки тестовой среды используется Dynagen. Dynagen – это некая надстройка над Dynamips, с ее помощью создается ini-файлы с описанием запускаемой среды. Достаточно хороший инструмент, но было бы хорошо использовать графическую оболочку для конфигурирования. Для этого как раз есть GNS3 (Graphical Network Simulator) – это графическая надстройка над Dynagen.

GNS3 – это превосходный инструмент, который позволяет всем интересующимся сетевыми технологиями поближе познакомиться с продуктами компании Cisco. Сетевым инженерам и администраторам дает возможность повысить навыки настройки оборудования, а также отличный помощник при подготовке к сдаче сертификационных экзаменов, таких как CCNA, CCNP, CCIP, CCIE RS, CCIE SP, JNCIA, JNCIS, JNCIE.

Отличительной чертой симулятора GNS3 является то, что используются реальные образы Cisco IOS. Поддерживается множество платформ (маршрутизаторы серий: 1700, 2600, 2691, 3600, 3700, 7200, средства обнаружения/предотвращения атак: IPS, IDS, PIX, ASA, коммутаторы: ATM, Frame Relay, Ethernet, а также устройства на базе JunOS, Linux-хосты).

Стоит упомянуть также о том, что эмулированную сеть можно связать с физической(реальной) локальной сетью, тем самым выстраивая единую взаимосвязанную топологию.

Установка и начальная настройка:

Нам понадобится:

- GNS3 – необходим ‘all-in-one’ пакет, в котором содержатся Dynamips, Qemu/Pemu, Putty and WinPCAP 4.1.1.
- WireShark – сниффер (анализатор сетевого трафика)
- Virtual PC Simulator – симулятор конечного хоста
- Cisco IOS – образы ОС.

После того как все перечисленное загружено, можно приступать к установке.

Для начала устанавливаем GNS3. Никаких сложностей возникнуть не должно. В ходе установки будут установлены все необходимые компоненты/драйвера/библиотеки доступа к нижним сетевым уровням.

Далее для своего удобства создайте структуру каталогов (папок), дабы все будущие проекты/лабораторные работы были структурированы:

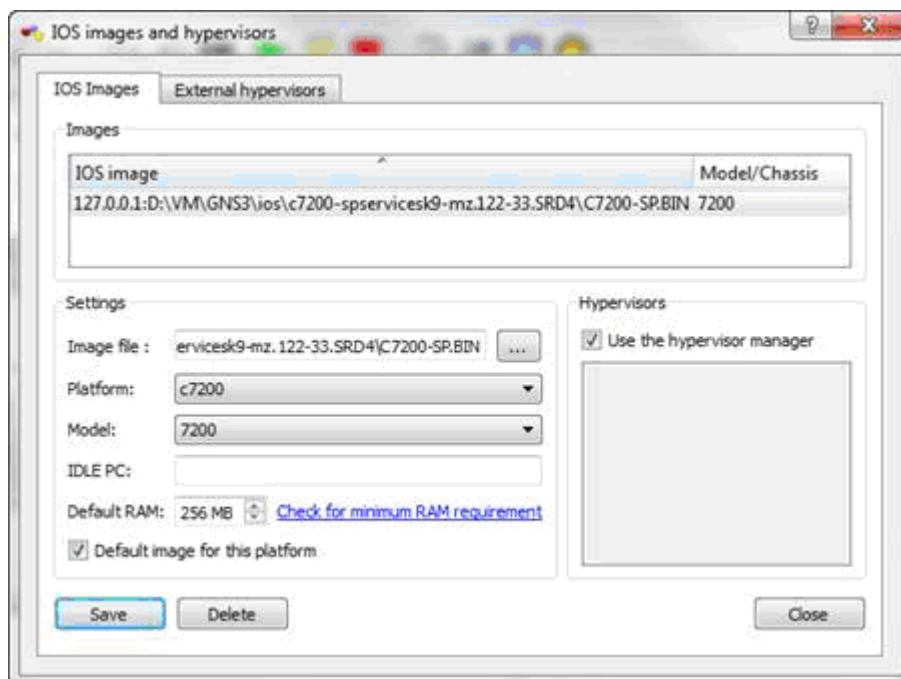
Краткое описание назначения: IOS – хранятся образы Cisco IOS, PROJECT – все проекты создаются здесь, WORK – рабочая директория, PEMU, QEMU, CAPTURE – директории, относящиеся к Qemu. TEMP – временная директория.

После этого запускаем GNS3. При первом запуске появляется диалоговое окно, в котором предлагается указать начальные настройки – (1 пункт) расположение Dynamips, (2 пункт) расположение образов Cisco IOS.

Для начала выбираем пункт #1, появляется диалоговое окно Preference. На вкладке Dynamips, в Executable path указываем путь к Dynamips. Если при установке использовался полный пакет (GNS3-x. x-win32-all-in-one), то ничего менять не надо, просто нажимаем кнопку Test. Должна появиться зеленая надпись: Dynamips successfully started. Нажимаем кнопку Apply, затем ОК.

Примечание: тоже самое можно сделать позже, используя главное меню Edit > Preferences

Далее необходимо указать путь к образам Cisco IOS. Для этого жмем на пункт (2). Если вы закрыли диалоговое окно, выбираем пункт меню Edit > IOS images and hypervisors.



Появится диалоговое окно, в котором необходимо указать образы Cisco IOS для каждой платформы, которые будут использоваться для последующего запуска. Для этого нажимаем кнопку [...] рядом с Image File и выбираем образ Cisco IOS. В поле Platform и Model указываем платформу и модель, в многих случаях по имени образа платформа и модель определяется автоматически. IDLE PC временно оставляем пустым. Нажимаем Save.

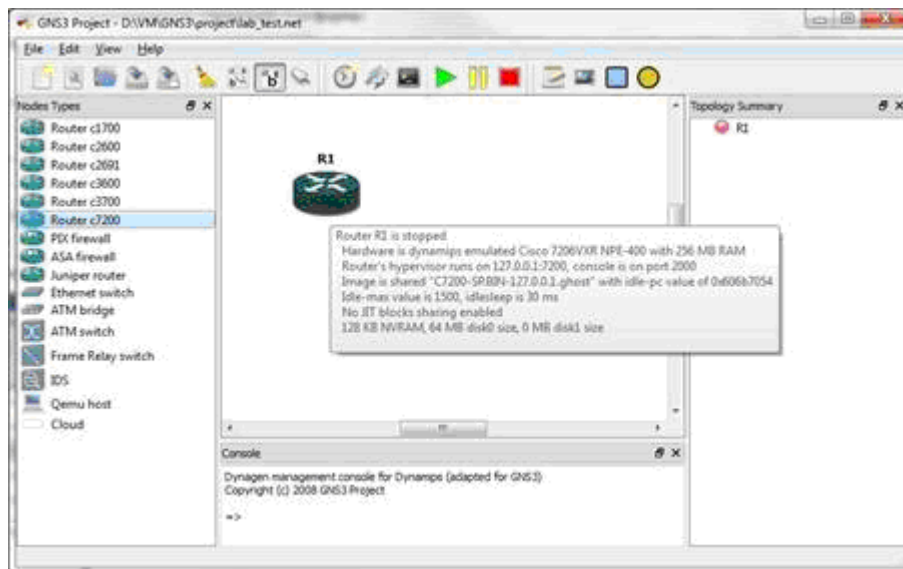
Необходимо выполнить эти действия для всех образов, которые вы планируете использовать.

Создание пробной топологии:

Теперь можно приступить к созданию тестовой среды. Запускаем GNS3, при старте выводится диалоговое окно создания/открытия проекта:

Назовем наш тестовый проект LAB_TEST (путь к файлу проекта заполняется автоматически, если все директории на этапе предварительной настройки были указаны). Также рекомендуется указать галочки 'сохранять файлы nvram', 'экспортировать конфигурации маршрутизаторов'. Это очень упростит работу с проектом в дальнейшем.

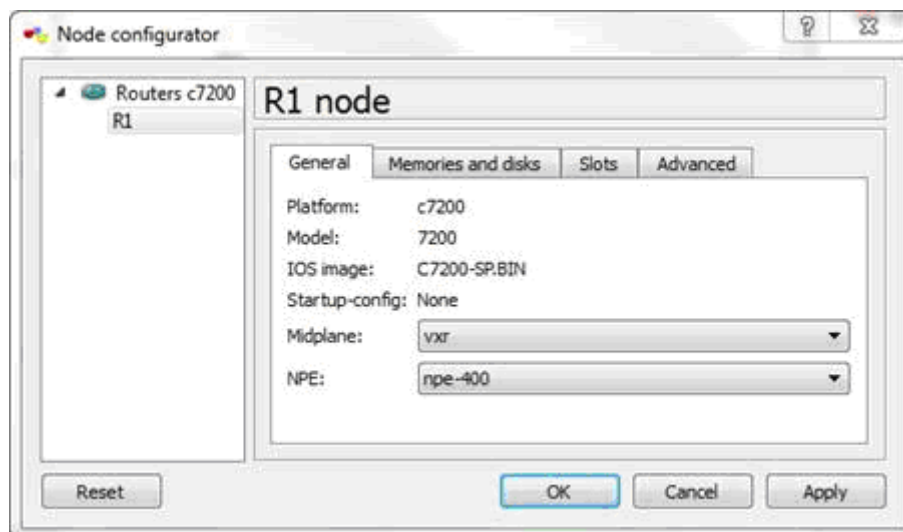
Добавим маршрутизатор 7200 в рабочую область проекта. Имя добавленного элемента – R1, при добавлении устройство выключено (о чем символизирует красный индикатор в правой панели). При наведении на иконку курсора мыши отобразится краткая информация об устройстве.



Выделив элемент правой кнопкой мыши, получим контекстное меню устройства:

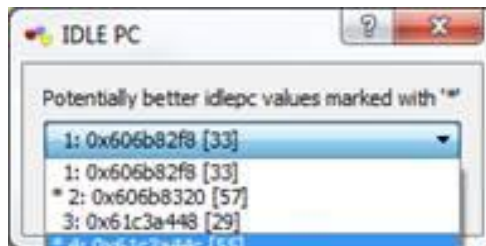


Пункт 'Configure' содержит основные настройки маршрутизатора, такие как параметры памяти RAM, NVRAM, дисков PCMCIA, компоновку слотов (интерфейсных модулей), MAC адрес шасси, регистр(confreg)

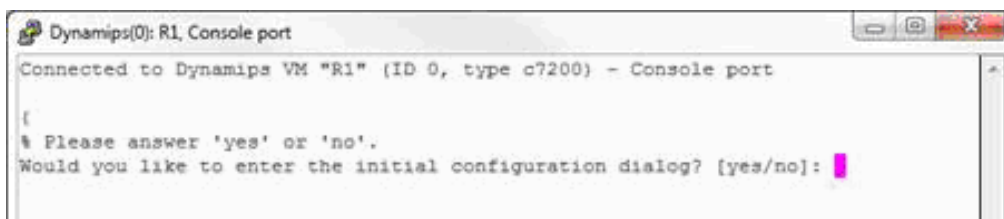


Запускаем маршрутизатор кнопкой Start контекстного меню. Сразу же после запуска рекомендуется выполнить подбор параметра IdlePC (находится там же в контекстном меню). Значительно снижает нагрузку на центральный процессор.

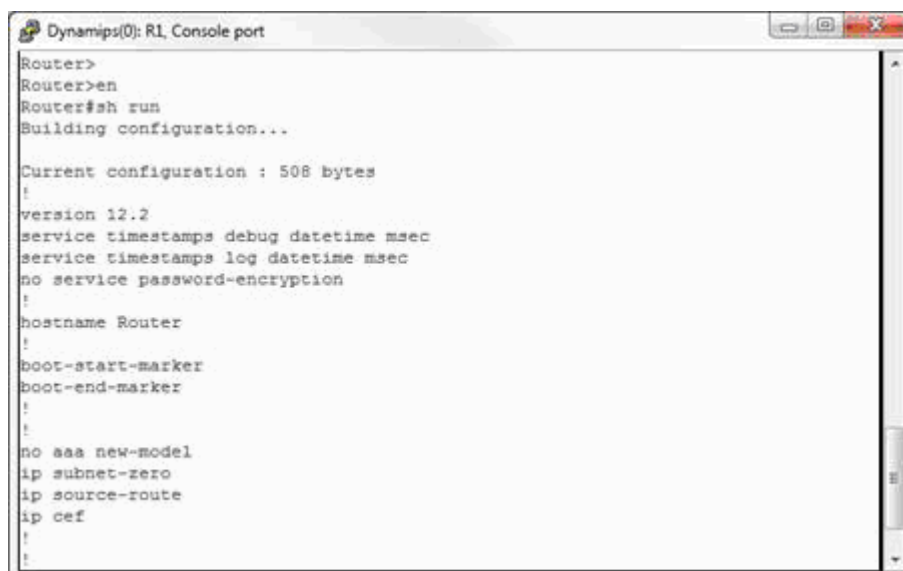
Пример подбора, наиболее подходящий параметр выделяется (*) звездочкой:



Подключение к маршрутизатору консолью (терминалом, в нашем случае Putty). В контекстном меню маршрутизатора выбираем Console, после чего откроется клиент Putty с подключением к маршрутизатору R1:



Отображается диалог начальной конфигурации устройства. Далее уже действуем по своему усмотрению настраивая устройство с самого начала в ручную.



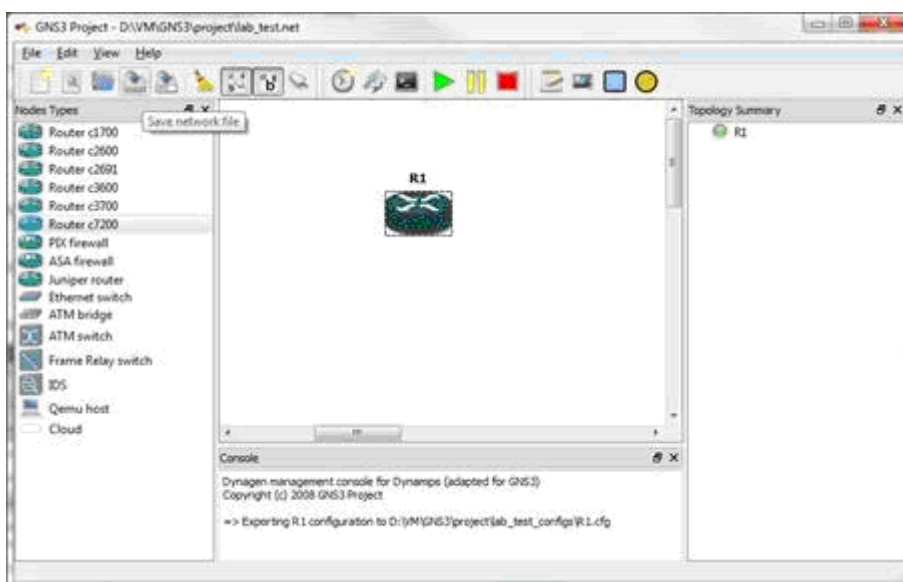
Сменим для примера имя маршрутизатора с дефолтного Router на R1, в соответствии с нашей топологией. Сохраним текущую конфигурацию в «startup» командой #wr (аналог команды #copy run start):


```
Dynamips(0): R1, Console port
!
!
!
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#host R1
R1(config)#
R1(config)#end
R1#w
*Jul 20 17:06:47.071: %SYS-5-CONFIG_I: Configured from console by console
R1#wr
Building configuration...
[OK]
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
```

Следует заметить, что ограничений по поддержке протоколов /команд/ функционала в эмуляторе нет. Все это зависит только от типа используемого образа IOS.

Будем считать, что в нашем случае настройка маршрутизатора выполнена. Для того, чтобы при следующей загрузке маршрутизатор подхватил нашу сохраненную конфигурацию выполним следующее: на панели задач найдем кнопку ‘Save Network File’, при нажатии которой произойдет сохранение проекта как такового, плюс экспорт конфигурационных файлов всех устройств. В нашем случае об этом будет свидетельствовать событие в центральной рабочей области :

‘=> Exporting R1 configuration to D:\VM\GNS3\project\lab_test_configs\R1.cfg’



Теперь, в папке проектов ‘D:\VM\GNS3\project\’ имеем следующую структуру:

```

lab_test.net
1  autostart = False
2  [127.0.0.1:7200]
3      workingdir = lab_test_working
4      udp = 10000
5      [[7200]]
6          image = D:\VM\GNS3\ios\c7200-sp\servicesk9-mz.122-33.SRD4\C7200-SP.BIN
7          idlepc = 0x61c3a44c
8          sparsemem = True
9          ghostios = True
10         [[ROUTER R1]]
11             console = 2000
12             cnfg = lab_test_configs\R1.cfg
13             x = -466.0
14             y = -441.0
15     [GNS3-DATA]
16         configs = lab_test_configs
17         workdir = lab_test_working
18

```

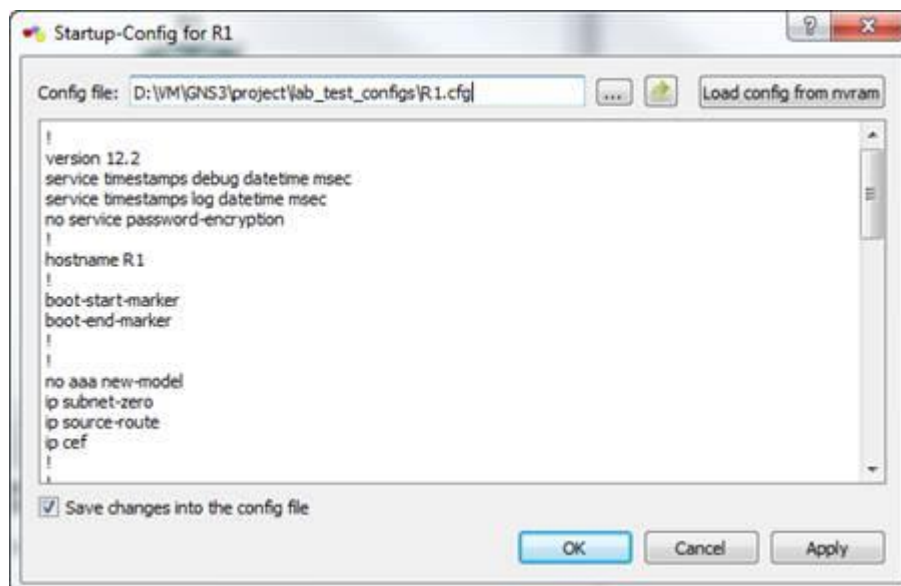
Папка ‘<имяпроекта>_configs’ содержит экспортированные конфигурации устройств:

Имя	Дата изменения	Тип	Размер
R1.cfg	20.07.2010 17:13	Файл "CFG"	1 КБ

Папка ‘<имяпроекта>_working’ содержит рабочие файлы проекта(файлы памяти RAM, NVRAM, файлы дисков и т. д.). Это виртуальные аналоги физических устройств маршрутизатора:

Имя	Дата изменения	Тип	Размер
c7200_ghost-C7200-SP.BIN-127.0.0.1.ghost_bootflash	20.07.2010 16:55	Файл "GHOST_BO...	8 192 КБ
c7200_ghost-C7200-SP.BIN-127.0.0.1.ghost_disk0	20.07.2010 16:55	Файл "GHOST_DL...	65 536 КБ
c7200_ghost-C7200-SP.BIN-127.0.0.1.ghost_log.txt	20.07.2010 16:55	Текстовый докум...	3 КБ
c7200_ghost-C7200-SP.BIN-127.0.0.1.ghost_nvram	20.07.2010 16:55	Файл "GHOST_NV...	128 КБ
c7200_R1_bootflash	20.07.2010 16:55	Файл	8 192 КБ
c7200_R1_disk0	20.07.2010 16:55	Файл	65 536 КБ
c7200_R1_lock	20.07.2010 16:34	Файл	0 КБ
c7200_R1_log.txt	20.07.2010 17:26	Текстовый докум...	26 КБ
c7200_R1_nvram	20.07.2010 16:55	Файл	128 КБ
c7200_R1_rommon_vars	20.07.2010 17:07	Файл	1 КБ
C7200-SP.BIN-127.0.0.1.ghost	20.07.2010 16:55	Файл "GHOST"	262 144 КБ

Инструмент версии GNS3 0.7.2 – редактор стартовой конфигурации. Выделив устройство правой кнопкой, выбираем пункт ‘Startup-config’, появится редактор конфигурации (открывается на редактирование файл ‘R1.cfg’, который мы экспортировали в предыдущем пункте):



Если есть понимание структуры конфигурационного файла, вносить изменения можно прямо в этом редакторе.

После редактирования (и нажатия кнопки ОК), редактор предложит загрузить этот стартовый конфигурационный файл (startup-config) в текущую конфигурацию (running-config), выполнит команду на маршрутизаторе **R1#copy running-config startup-config**

Содержание отчёта:

1. Титульный лист.
2. Цель работы и задание.
3. Скриншоты основных этапов работы.
4. Анализ полученных результатов и выводы по работе.

Контрольные вопросы:

1. Перечислить программное обеспечение, необходимое в лабораторной работе.
2. Дать определение и назначение каждому из приложений.
3. Кратко описать создание пробной топологии.

Лабораторная работа № 2.

Подключение и администрирование маршрутизаторов Cisco, основы операционной системы Cisco IOS, статическая маршрутизация, настройка протокола RIP.

Цель работы:

Получить базовые навыки администрирования маршрутизаторов Cisco.

Задание:

Создать топологию для собственной квартиры. Для каждой комнаты - своя подсеть, каждый член семьи имеет свой компьютер-хост. Выполнить 2 варианта маршрутизации для своей топологии: статическая и RIP.

Теоретическая часть:

В каждом маршрутизаторе имеется некоторое число физических интерфейсов. Наиболее распространенными типами интерфейсов являются: Ethernet/FastEthernet и последовательные интерфейсы (Serial).

Интерфейс командной строки: Маршрутизатор конфигурируется в командной строке операционной системы Cisco IOS. Подсоединение к маршрутизатору осуществляется через Telnet на IP-адрес любого из его интерфейсов или с помощью любой терминальной программы через последовательный порт компьютера, связанный с консольным портом маршрутизатора. Последний способ предпочтительнее, потому что процесс конфигурирования маршрутизатора может изменять параметры IP-интерфейсов, что приведет к потере соединения, установленного через Telnet. Кроме того, по соображениям безопасности доступ к маршрутизатору через Telnet следует запретить.

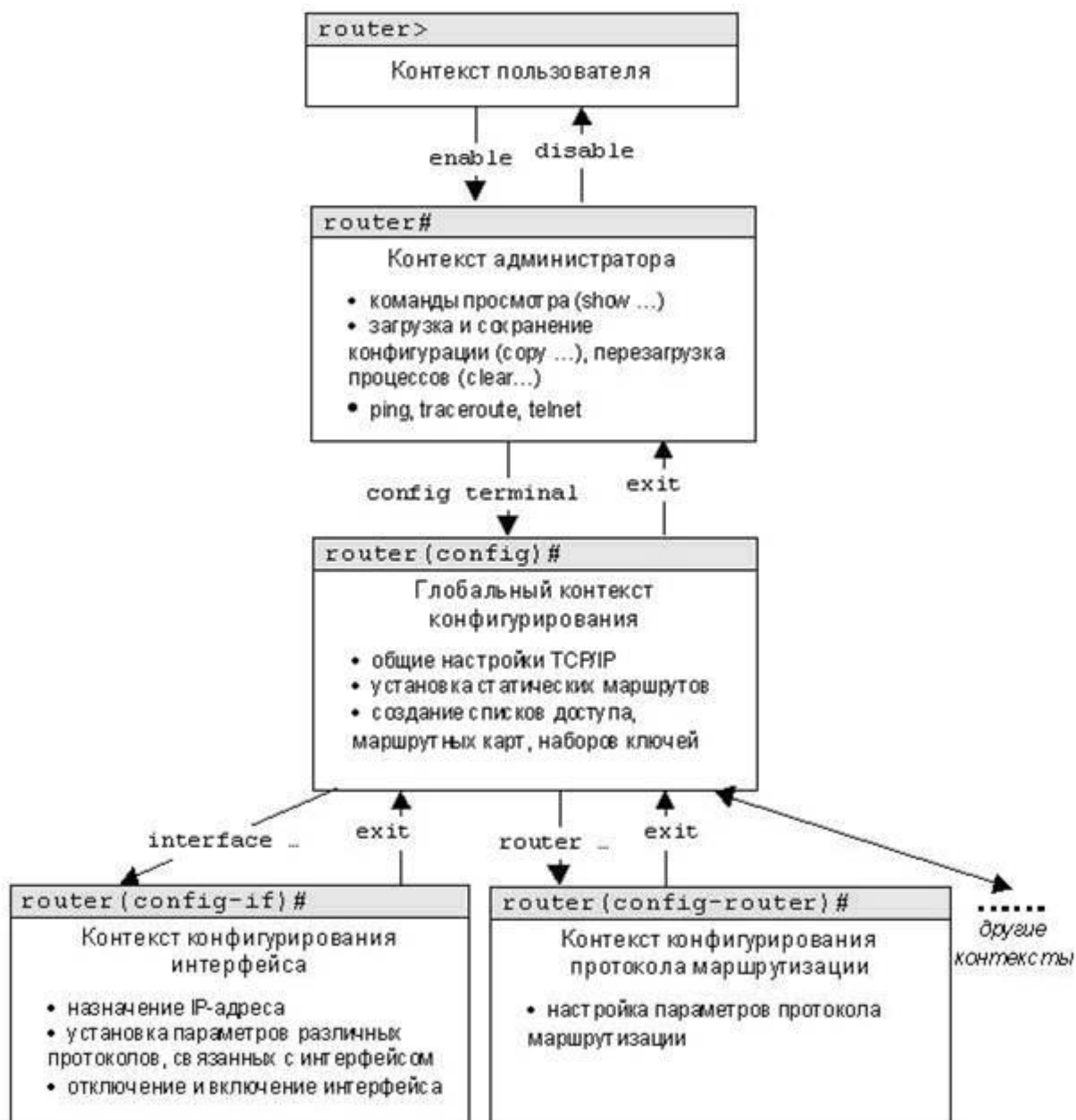
Обратите внимание, что аварийное отключение оператора от консоли не регистрируется маршрутизатором и сеанс остается в том же состоянии. При повторном подключении оператор окажется в том же контексте (если только не сработал автоматический выход в контекст пользователя по таймеру неактивности). Напротив, при разрыве Telnet-соединения маршрутизатор закрывает сеанс работы оператора.

При работе в командной строке Cisco IOS существует несколько контекстов (режимов ввода команд).

Контекст пользователя: Открывается при подсоединении к маршрутизатору; обычно при подключении через сеть требуется пароль, а при подключении через консольный порт пароль не нужен. В этот же контекст командная строка автоматически переходит при продолжительном отсутствии ввода в контексте администратора. В контексте пользователя доступны только простые команды, не влияющие на конфигурацию маршрутизатора. Вид приглашения командной строки:

router>

Вместо слова **router** выводится имя маршрутизатора, если оно установлено.



Контекст администратора (контекст "exec"): Открывается командой **enable**, поданной в контексте пользователя; при этом обычно требуется пароль администратора. В контексте администратора доступны команды, позволяющие получить полную информацию о конфигурации маршрутизатора и его состоянии, команды перехода в режим конфигурирования, команды сохранения и загрузки конфигурации. Вид приглашения командной строки:

router#

Обратный переход в контекст пользователя производится по команде **disable** или по истечении установленного времени неактивности. Завершение сеанса работы - команда **exit**.

Любая команда конфигурации вступает в действие немедленно после ввода (а не после возврата в контекст администратора).

Все команды и параметры могут быть сокращены (например, "enable" - "en", "configure terminal" - "conf t"); если сокращение окажется неоднозначным, маршрутизатор сообщит об этом, а по нажатию табуляции выдаст варианты, соответствующие введенному фрагменту.

В любом месте командной строки для получения помощи может быть использован вопросительный знак:

router#? /список всех команд данного контекста с комментариями/

router#co? /список всех слов в этом контексте ввода, начинающихся на "co" - нет пробела перед "?"/

router#conf? /список всех параметров, которые могут следовать за командой config - перед "?" есть пробел/

Имена сетевых интерфейсов также могут быть сокращены, например, вместо "ethernet0/1" достаточно написать "e0/1".

Отмена любой команды (отключение опции или режима, включаемых командой, снятие или удаление параметров, назначаемых командой) производится подачей этой же команды с префиксом "no", например:

router(config-if)#shutdown /отключить интерфейс/

router(config-if)#no shutdown /включить интерфейс/

Любая команда X, введенная в контекстах пользователя или администратора и не опознанная маршрутизатором, трактуется как telnet X. Это не очень удобно, поскольку любой ошибочный ввод приводит к тому, что маршрутизатор начинает опрашивать сервер DNS для преобразования введенной строки в IP-адрес, что вызывает неприятные задержки в работе оператора. Избежать этого можно, подав команду в режиме конфигурации терминальной линии:

router(config-line)#transport preferred none

Перезагрузка маршрутизатора, если таковая необходима, выполняется по команде reload, поданной в контексте администратора.

Просмотр, сохранение и загрузка конфигурации: При загрузке маршрутизатор считывает команды конфигурации из NVRAM, где они хранятся в виде текстового файла. Таким образом, создается рабочая конфигурация (running config). Конфигурация, сохраненная в NVRAM, называется начальной (startup config). В процессе работы маршрутизатора администратор может вводить дополнительные конфигурационные команды, в результате чего рабочая конфигурация становится отличной от начальной.

Просмотр начальной и рабочей конфигураций маршрутизатора производится в контексте администратора:

router#show startup-config

router#show running-config

Настройка интерфейсов: По умолчанию все интерфейсы выключены. Интерфейс включается командой

router(config-if)#no shutdown

Работоспособность настроек уровней 1 и 2 можно проверить командой в контексте администратора:

router#show interface интерфейс

Глобальный контекст конфигурирования открывается командой **config terminal** ("конфигурировать через терминал"), поданной в контексте администратора. Глобальный контекст конфигурирования содержит как непосредственно команды конфигурирования маршрутизатора, так и команды перехода в контексты конфигурирования подсистем маршрутизатора.

Контекст конфигурирования интерфейса открывается командой **interface имя_интерфейса** (например **interface serial0**), поданной в глобальном контексте конфигурирования

Настройка IP-адреса интерфейса производится командой:

router(config-if)#ip address адрес маска

Подробная информация о параметрах протокола IP доступна в контексте администратора по команде:

router#show ip interface имя_интерфейса

Краткая сводная таблица состояний IP-интерфейсов:

router#show ip interface brief

Назначение статических маршрутов: Для назначения дополнительных статических маршрутов в контексте глобальной конфигурации вводится команда:

router(config)#ip route префикс маска след_маршрутизатор адм. приоритет]

Статический маршрут активен только тогда, когда следующий маршрутизатор достижим - то есть, существует маршрут в сеть, где находится следующий маршрутизатор (и соответствующий интерфейс активен).

Настройка протокола RIP: Конфигурирование протокола RIP выполняется в контексте, попасть в который можно командой

router(config)#router rip

router(config-router)#

Единственной обязательной командой конфигурации RIP является команда (или несколько команд) **network**:

router(config-router)#network префикс

Получив указанную команду, маршрутизатор выполняет следующие действия:

1. Заменяет введенный префикс на адрес классовой сети, которой этот префикс принадлежит. Например, если вы ввели `network 134.56.12.0`, то в конфигурации будет: `network 134.56.0.0`. Соответственно, если вы подали две команды с аргументами `134.56.12.0` и `134.56.25.0`, в конфигурации всё равно будет одна строка `network 134.56.0.0`. Причина такого поведения IOS - тяжелое наследие классовой адресации в сочетании с необходимостью одновременной поддержки обеих версий RIP. Следует подчеркнуть, что такое поведение имеет место, независимо от используемой версии RIP.
2. Включает в RIP-систему все непосредственно подсоединенные сети, которые попадают в адресное пространство классовой сети, определенной на шаге 1. Все эти сети будут включаться маршрутизатором в вектор расстояний, рассылаемый соседям.
3. Векторы расстояний рассылаются через все интерфейсы, подключенные к сетям, определенным на шаге 2.

Последствием непродуманного поведения маршрутизатора на шаге 1, может быть следующая ситуация. Маршрутизатор подключен к сетям `1.1.1.0/24`, `1.2.2.0/24`, `1.3.3.0/24`, при этом в RIP-систему входят только первые две сети. Однако "округление" до классовой сети `1.0.0.0/8`, выполняемое на шаге 1, приводит к тому, что в RIP-систему включается также и сеть `1.3.3.0/24`.

По умолчанию маршрутизатор рассылает сообщения RIP версии 1, а принимает сообщения обеих версий. Поскольку в современных сетях протокол RIP v1 является полностью неадекватным (за редкими исключениями, где требуется обеспечение совместимости с устаревшим оборудованием/ПО), необходимо активизировать версию 2:

`router(config-router)#version 2`

Настройка метрик: В качестве метрики RIP использует число промежуточных узлов (hops). Иногда бывает необходимо принудительно ухудшить метрику какого-либо маршрута, чтобы заставить протокол использовать другой маршрут:

`router(config-router)#offset-list ACL {in|out} N [интерфейс]`

Аргументы команды:

- ACL - номер стандартного списка доступа (1-99), используемого для отбора маршрутов, метрика которых будет увеличена. Отбираются маршруты к сетям, адреса которых подпадают под действие оператора `permit` списка доступа.
- `in` или `out` - параметр указывает, метрика каких маршрутов увеличивается: входящих (маршрутизатор получает маршрут, увеличивает метрику и использует в расчетах увеличенное значение) или исходящих (маршрутизатор использует обычное значение, но отправляет увеличенное).
- N - число, прибавляемое к метрике.

- интерфейс - если указан интерфейс, то действие команды распространяется только на маршруты, получаемые (in) или отправляемые (out) с этого интерфейса.

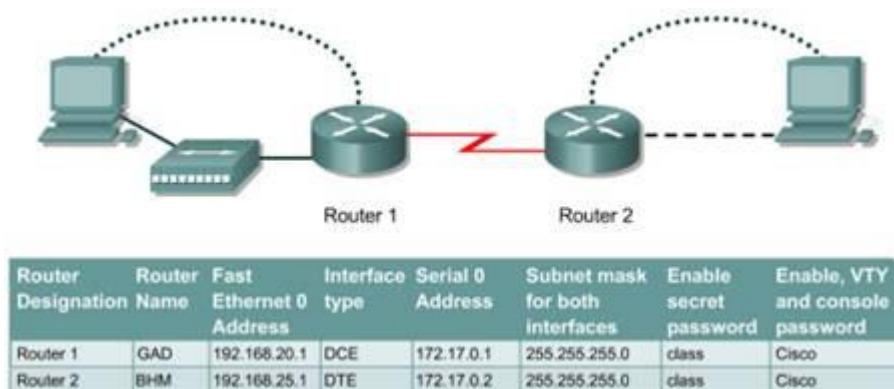
Примечание - Здесь и далее по ходу лабораторного практикума необходимо четко различать направление движения маршрутной информации и направление движения пользовательских дейтаграмм. Если говорится, что маршрутизатор получает маршрут в сеть А через интерфейс I, это означает, что через интерфейс I поступает информация о доступности сети А. Если эта информация будет установлена в таблицу маршрутов, то дейтаграммы, адресованные в А, будут покидать интерфейс I.

Кроме того, необходимо четко различать, что за адреса в том или ином случае отбираются списками доступа. Например, в команде `offset-list` список доступа отбирает адреса сетей из поступившего (или отправляемого) вектора расстояний, а не адреса отправителя (или получателя) RIP-сообщений. Списки доступа являются универсальным инструментом для отбора некоторой адресной информации, и интерпретация действия списков целиком зависит от контекста, в котором они применяются.

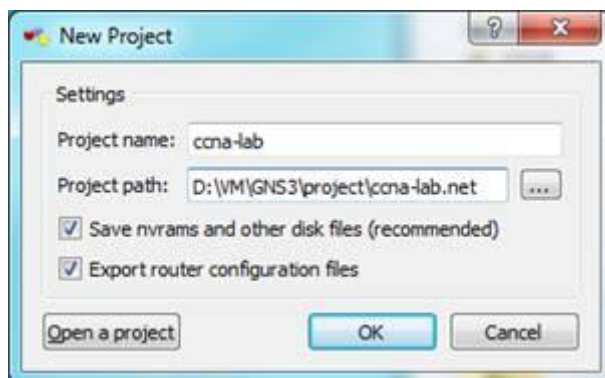
Show & debug: В выводе команды `show ip route` маршруты, полученные через протокол RIP, отмечены буквой R. Кроме обычной информации (адрес назначения, следующий маршрутизатор и интерфейс), выводятся административный приоритет маршрута (обычно 120), метрика и время, прошедшее с момента последнего обновления сведений о маршруте.

Пример работы с пакетом программ:

В качестве топологии возьмем лабораторную работу из учебного руководства Cisco. Схема достаточно тривиальная, но позволит понять, как в GNS3 строить топологию. Будем использовать два маршрутизатора, которые соединяют две разные сети. Подробнее на рисунке:

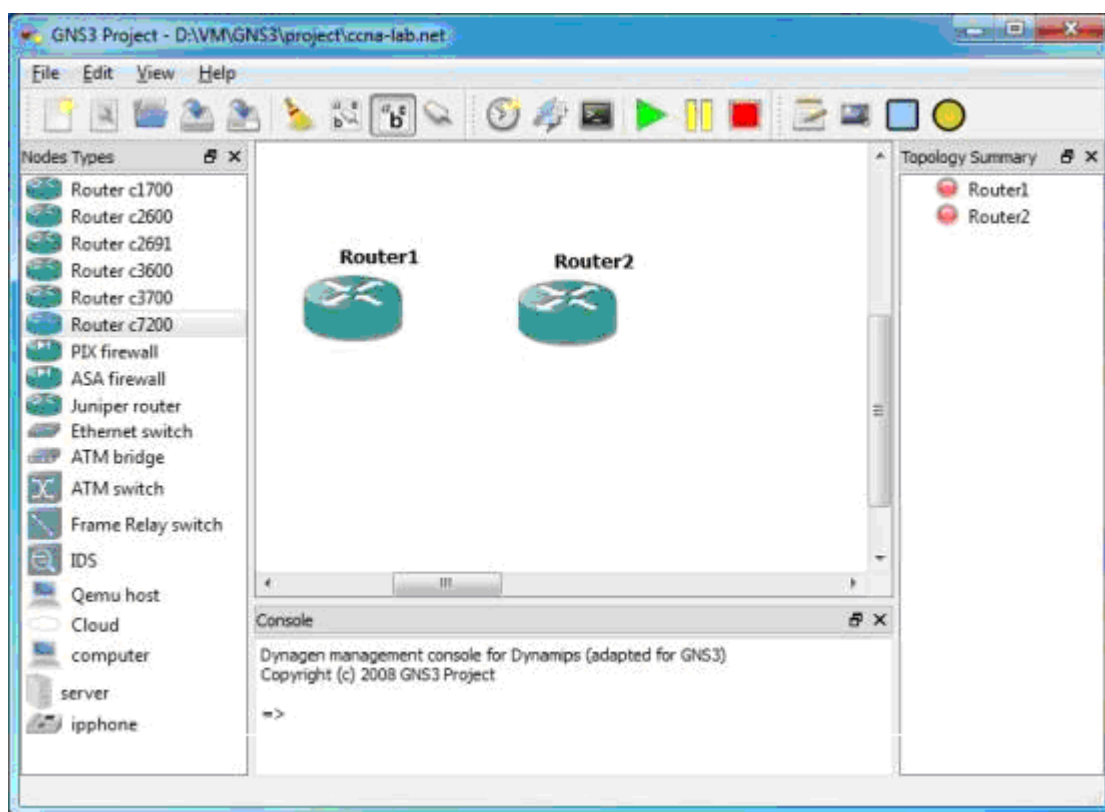


Приступим к созданию проекта, для начала запустим GNS и назовем наш проект (или открываем созданный до этого проект кликнув «open a project» и выбрав файл проекта с расширением. net):

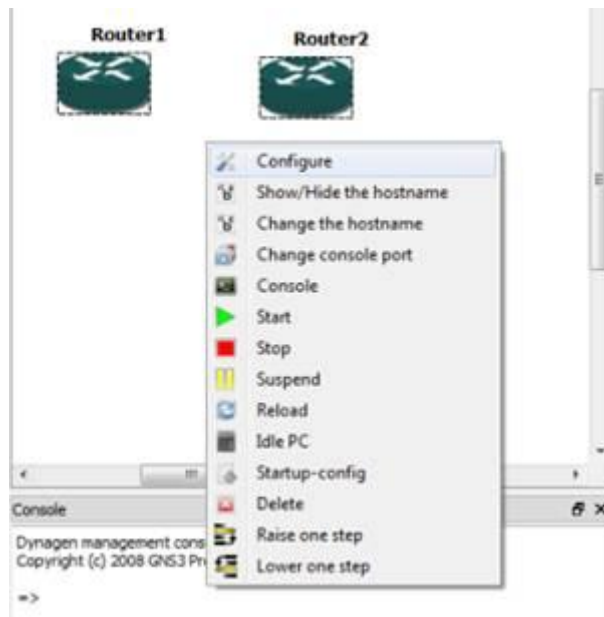


Для удобства работы с проектом, отметим галочками пункты «Сохранять nvrams и другие диски» и «Экспортировать конфигурационные файлы маршрутизаторов».

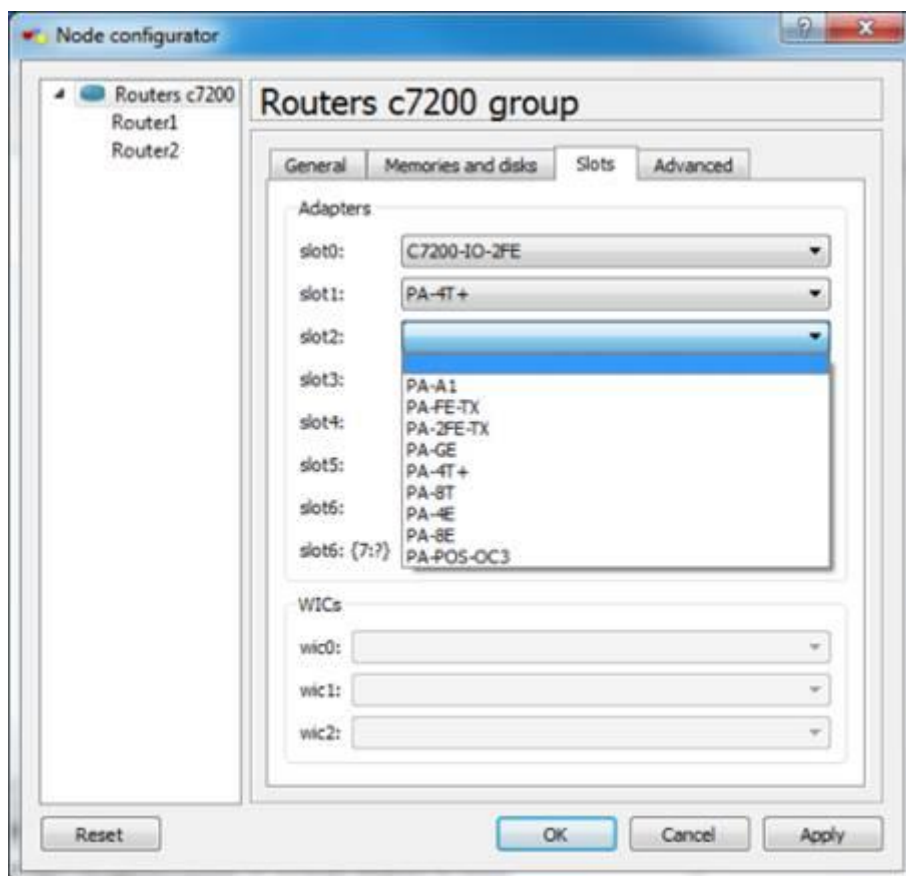
Получим чистое рабочее пространство нашего проекта, первое, что необходимо сделать, добавить пару маршрутизаторов. Переименуем их в Router1 и Router2 для соответствия рисунку лабораторной работы.



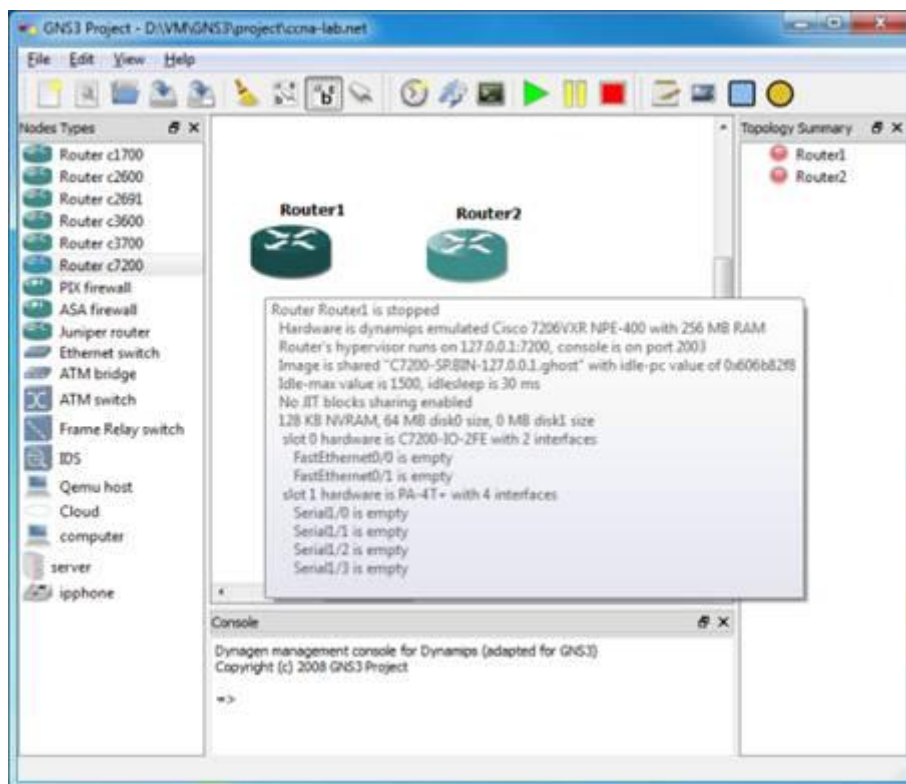
В качестве платформы по умолчанию при начальной конфигурации GNS мы выбрали маршрутизатор серии 7200. Потому как это наиболее интересная платформа с точки зрения всевозможных настроек и стабильности IOS(для эмуляции) в том числе (7200 модели: 7204VXR, 7206VXR Применение: Средний провайдер; ядро небольшой городской сети или сети крупного предприятия; граница крупной сети Цена: от 20000). Для того, чтобы соединить роутеры, нам необходимо добавить в них модули интерфейсных карт (по аналогии с аппаратными маршрутизаторами):



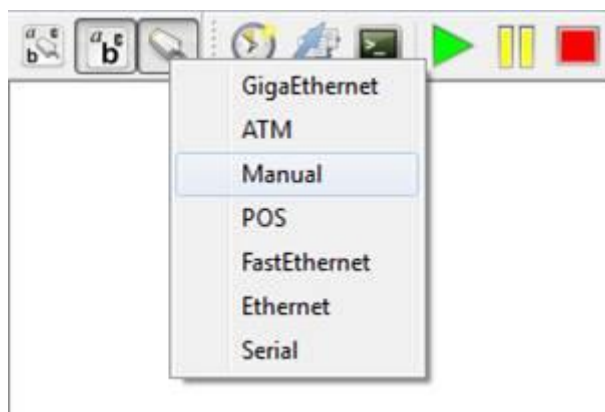
Выбираем контекстное меню «Configure» -> «Slots» и «вставляем» в нужный слот необходимый модуль. В нулевой слот ставим модуль «2FE» с двумя fastethernet интерфейсами, а в первый «4Т+» с четырьмя последовательными (serial) интерфейсами.



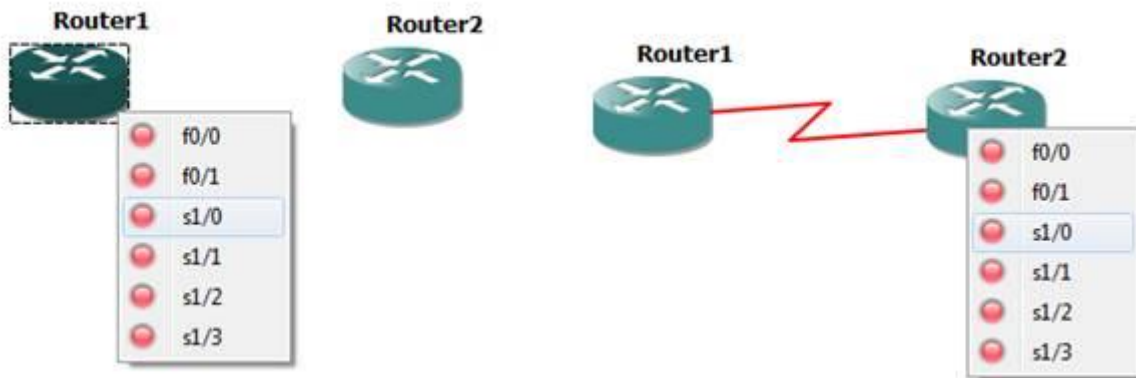
Наведя на роутер курсор, мы видим, что доступны следующие интерфейсы fa0/0, fa0/1, ser1/0, ser1/1, ser1/2 и ser1/3.



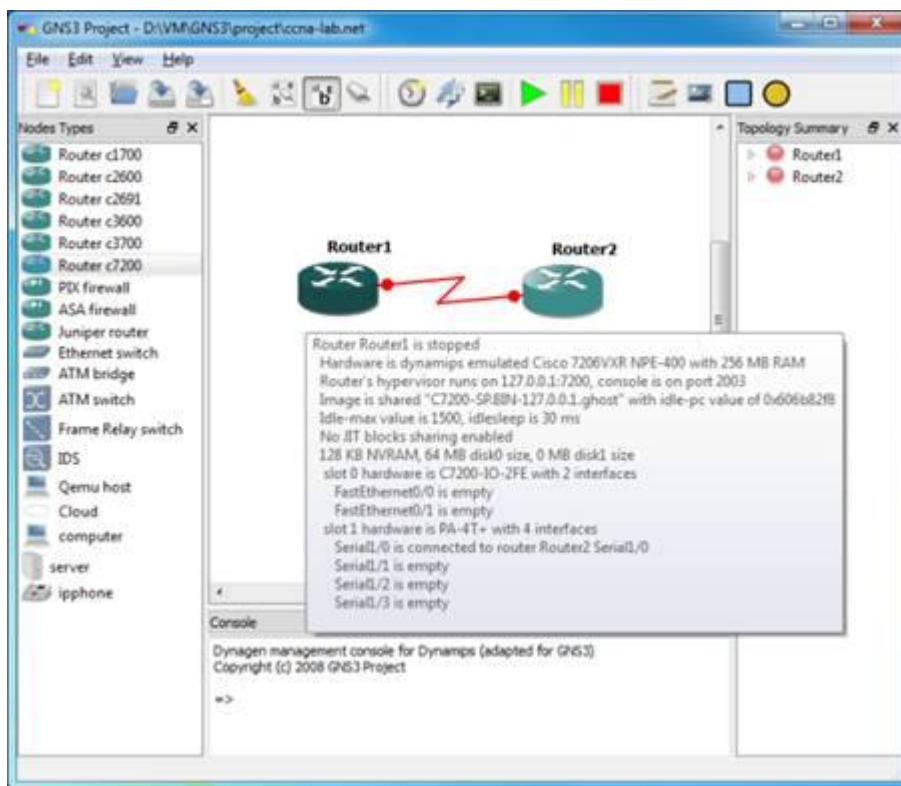
Для нашей топологии этого более чем достаточно. Для более сложных лабораторных работ можно использовать и другие модули, описание и функционал которых не трудно найти в Интернете. Для соединения роутеров мы будем использовать последовательные интерфейсы, для этого на верхней панели нажмем кнопку с изображением коннектора («add a link»), режим выберем ручной(manual):



Далее, при выборе первого роутера откроется список интерфейсов, выберем, например, s1/0 и на втором роутере, к примеру, выберем тоже интерфейс s1/0.



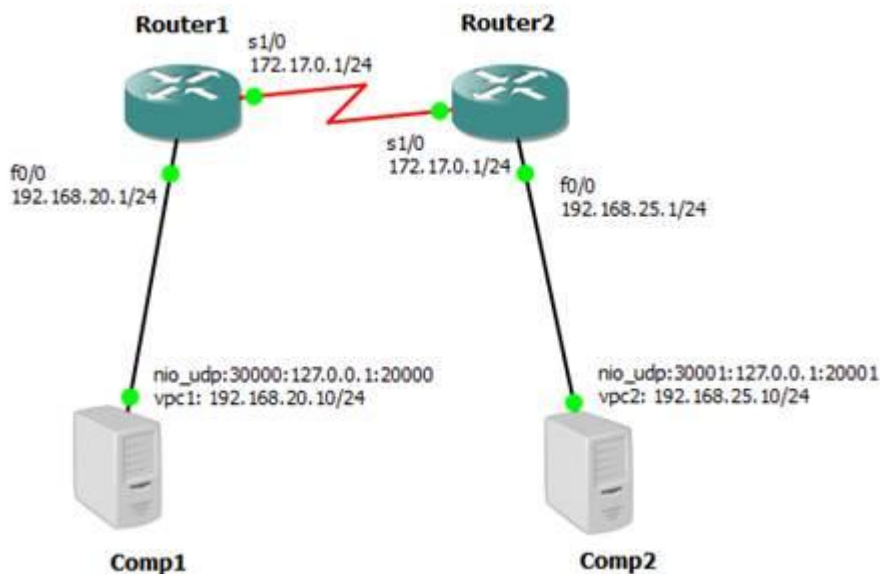
«Зависнув» курсором над первым роутером, мы увидим «S1/0 is connected to Router2 S1/0».



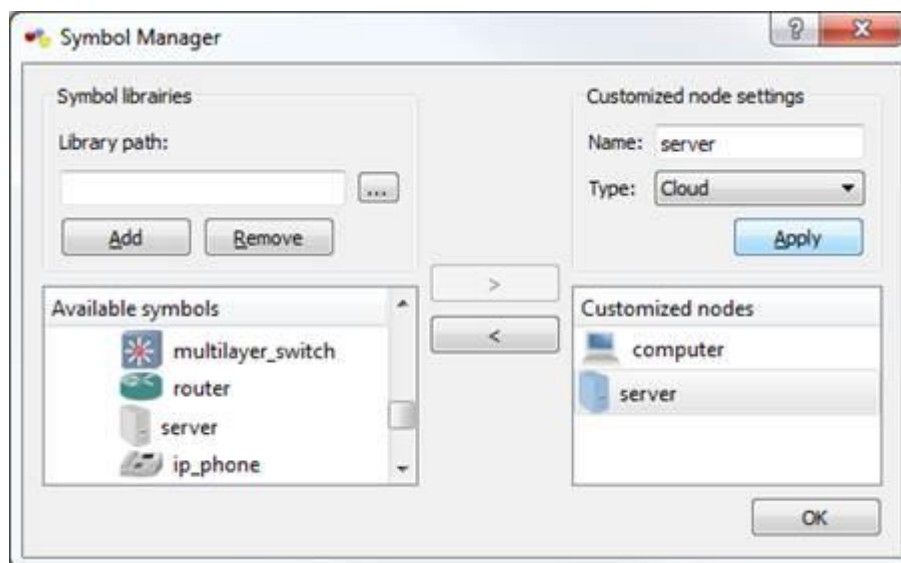
Подпишем интерфейсы с указанием адресов, для того чтоб имена интерфейсов отображались, нужно нажать кнопку на панели:



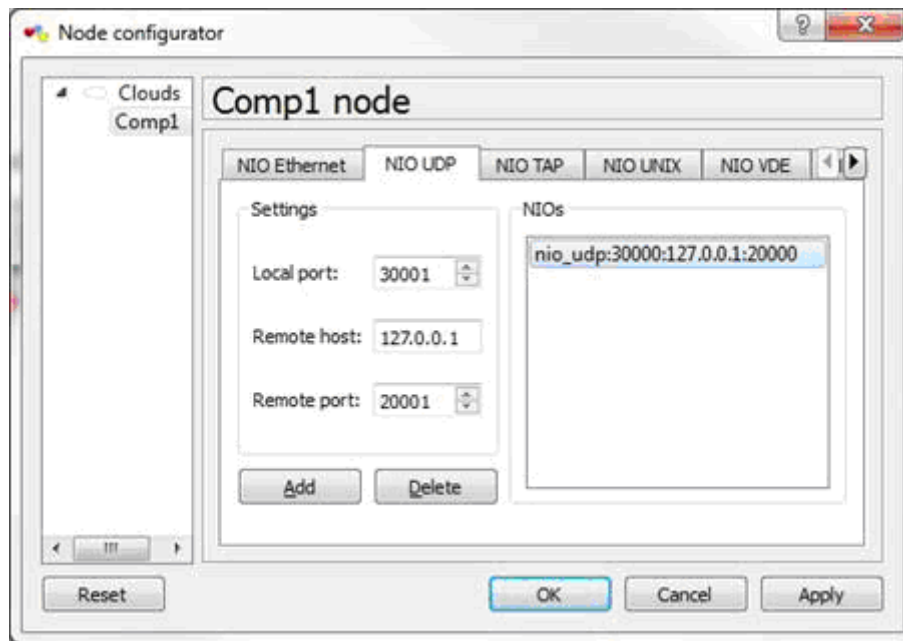
Настроим объект в GNS, тип – облако, иконку присвоим в виде PC для наглядности:



Остановимся подробнее на эмуляции конечных узлов (хостов). Для начала добавим новый символ в оболочку GNS, для этого в главном меню окна GNS идем в «Edit» -> «Symbol Manager», добавляем к примеру символ SERVER из поля доступных символов(слева) в используемые(справа). Выбираем тип – «Cloud» (облако) и обязательно(!) жмем «Apply».



После этого, в списке доступных устройств появляется объект Server. Добавим его в рабочую область и нажмем «Configure». На вкладке «NIO UDP» указываем «local port 30000, host 127.0.0.1, remote port 20000», это потребуется в дальнейшем для связки объекта SERVER с эмулятором узла VPCS.



Будьте внимательны! Виртуальному узлу (host) #1 в консоли VPC («VPCS1») соответствует связка портов – «nio_udp:30000:127.0.0.1:20000», второму виртуальному узлу («VPCS2») соответствует - «nio_udp:30001:127.0.0.1:20001» и так далее, всего утилита VPC может эмулировать до 9 узлов, причем все они могут быть в разных сетях и подключены к любым устройствам в топологии. В корневой папке с утилитой есть файл «startup.vpc», в нем можно настроить начальные параметры симулятора при запуске. В нашем случае мы зададим параметры двух виртуальных узлов. Первому узлу назначим адрес 192.168.20.10 шлюз 192.168.20.1 и префикс маски /24, второму узлу назначим адрес 192.168.25.10 шлюз 192.168.25.1 и префикс маски /24. Эта адресация соответствует нашей тестовой топологии.

```

1 # The startup file of VPC
2 # pc1, pc2 configured!
3 1
4 ip 192.168.20.10 192.168.20.1 24
5 2
6 ip 192.168.25.10 192.168.25.1 24
7

```

После того, как отредактировали startup-файл симулятора VPC, запустим его исполняемый файл. Увидим результат выполнения startup-файла: консоль

сообщит, что применены параметры для PC1, PC2. Проверяйте их при работе, дабы не ломать голову, что что-то не работает. Переключаться в консоли между узлами очень просто – введите номер узла, к которому хотите переключиться и нажмите enter. Из под виртуальных узлов можно выполнять простейшие команды для проверки сетевой связи устройств (например ping). Мы видим, что PC1 успешно пингует интерфейс маршрутизатора fa0/0, а также другой интерфейс – serial1/0.



```

Welcome to Virtual PC Simulator for dynamips, v0.16c
Dedicated to Daling.
Build time: Mar 12 2009 11:32:18
All rights reserved.

NOTICE: MAY NOT use this software for commercial purposes unless
you get an appropriate commercial license for it.

Please contact me at mirnshi@gmail.com or http://mirnshi.cublog.cn
if you have any questions.

Press '?' to get help.

Executing the startup file
PC1 : 192.168.20.10 255.255.255.0 gateway 192.168.20.1
PC2 : 192.168.25.10 255.255.255.0 gateway 192.168.25.1

VPCS 3 >1
VPCS 1 >
VPCS 1 >ping 192.168.20.1
192.168.20.1 icmp_seq=1 time=60.000 ms
192.168.20.1 icmp_seq=2 time=40.000 ms
192.168.20.1 icmp_seq=3 time=30.000 ms
192.168.20.1 icmp_seq=4 time=30.000 ms
192.168.20.1 icmp_seq=5 time=30.000 ms

VPCS 1 >ping 172.17.0.1
172.17.0.1 icmp_seq=1 time=60.000 ms
172.17.0.1 icmp_seq=2 time=30.000 ms
172.17.0.1 icmp_seq=3 time=30.000 ms
172.17.0.1 icmp_seq=4 time=30.000 ms
172.17.0.1 icmp_seq=5 time=50.000 ms

```

Запускаем маршрутизатор. Нажимаем правой кнопкой на маршрутизаторе и выбираем в контекстном меню Start. Открыв диспетчер задач можно увидеть, что загрузка процессора 100%.

Для того чтобы уменьшить загрузку ЦП, необходимо сконфигурировать параметр idlers. Для этого нажимаем правой кнопкой на пиктограмме маршрутизатора и выбираем пункт меню Idle pc. После этого начинается подсчет значений idlers. После этого появится диалоговое окно IDLE PC, в котором предлагается выбрать одно из подсчитанных значений. Наиболее лучшее значение помечается звездочкой (“*”). Если не появились записи помеченные звездочкой или загрузка ЦП не уменьшилась, следует выбрать другие значения, для этого повторите описанную выше процедуру. Загрузка процессора происходит по причине того что Dynamips не знает когда Cisco IOS находится в режиме бездействия, или когда производит какие-то операции.

Для того, чтобы обратиться к другому маршрутизатору («Router 2») необходимо выполнить настройку маршрутизации.


```
Dynamips(1): Router1, Console port
Router>
Router>
Router>
Router>
Router>
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname GAD
GAD(config)#int s1/0
GAD(config-if)#ip address 172.17.0.1 255.255.255.0
GAD(config-if)#no shut
GAD(config-if)#int fa0/0
GAD(config-if)#ip address 192.168.20.1 255.255.255.0
GAD(config-if)#no shut
GAD(config-if)#end
GAD#wr
*Sep 22 18:09:09.915: %SYS-5-CONFIG_I: Configured from console by console
Building configuration...
[OK]
GAD#
```

```
Dynamips(0): Router2, Console port
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname BHM
BHM(config)#int ser0/0
% Invalid input detected at '^' marker.
BHM(config)#int ser1/0
BHM(config-if)#ip address 172.17.0.2 255.255.255.0
BHM(config-if)#no shut
BHM(config-if)#int fast0/0
BHM(config-if)#ip address 192.168.25.1 255.255.255.0
BHM(config-if)#no shut
BHM(config-if)#end
BHM#
*Sep 22 18:11:39.039: %SYS-5-CONFIG_I: Configured from console by console
BHM#
BHM#wr
Building configuration...
[OK]
```

После того, как на маршрутизаторах настроили интерфейсы и маршрутизацию с первого виртуального узла PC1 доступен второй узел PC2.

```
vpc
welcome to Virtual PC Simulator for dynamips, v0.16c
Dedicated to Daling.
Build time: Mar 12 2009 11:32:18
All rights reserved.

NOTICE: MAY NOT use this software for commercial purposes unless
you get an appropriate commercial license for it.

Please contact me at mirnshi@gmail.com or http://mirnshi.cublog.cn
if you have any questions.

Press '?' to get help.
Executing the startup file
PC1 : 192.168.20.10 255.255.255.0 gateway 192.168.20.1
PC2 : 192.168.25.10 255.255.255.0 gateway 192.168.25.1

VPCS 3 >1
VPCS 1 >
VPCS 1 >ping 192.168.25.10
192.168.25.10 icmp_seq=1 time=70.000 ms
192.168.25.10 icmp_seq=2 time=70.000 ms
192.168.25.10 icmp_seq=3 time=100.000 ms
192.168.25.10 icmp_seq=4 time=70.000 ms
192.168.25.10 icmp_seq=5 time=70.000 ms
VPCS 1 > VERY GOOD ;)
```

Содержание отчёта:

1. Титульный лист.
2. Цель работы и задание.
3. Скриншоты основных этапов работы.
4. Анализ полученных результатов и выводы по работе .

Контрольные вопросы:

1. Перечислить основные команды, использованные в лабораторной работе.
2. Кратко описать процесс настройки интерфейсов маршрутизатора.
3. Протокол динамической маршрутизации RIP (ver. 1,2). Сравнительный анализ версий протокола.
4. Прокомментировать настройку статических маршрутов, протокола RIP и метрик.

Лабораторная работа № 3.

Настройка маршрутизации по протоколу OSPF.

Цель работы:

Получить базовые навыки администрирования маршрутизаторов Cisco.

Задание:

Разработать топологию сети одноэтажного офисного здания. На этаже 4 офиса, каждый из которых находится в отдельной подсети. Необходимо реализовать доступ между офиса таким образом, чтобы с любого ПК в офисе «А» можно было получить доступ к любому ПК в офисе «Б», предусмотрев временную потерю связи между офисами. Количество узлов выбрать в соответствии с ограничениями VPCS.

Теоретическая часть:

Алгоритмы состояния связей. Протокол OSPF (алгоритм Дейкстры):

Протокол OSPF (Open Shortest Path First) является реализацией алгоритма состояния связей (он принят в 1991 году) и обладает многими особенностями, ориентированными на применение в больших гетерогенных сетях.

Протокол OSPF вычисляет маршруты в IP-сетях, сохраняя при этом другие протоколы обмена маршрутной информацией.

Непосредственно связанные маршрутизаторы называются "соседями". Каждый маршрутизатор хранит информацию о том, в каком состоянии по его мнению находится сосед. Маршрутизатор полагается на соседние маршрутизаторы и передает им пакеты данных только в том случае, если он

уверен, что они полностью работоспособны. Для выяснения состояния связей маршрутизаторы-соседи достаточно часто обмениваются короткими сообщениями HELLO.

Для распространения по сети данных о состоянии связей маршрутизаторы обмениваются сообщениями другого типа. Эти сообщения называются router links advertisement - объявление о связях маршрутизатора (точнее, о состоянии связей). OSPF-маршрутизаторы обмениваются не только своими, но и чужими объявлениями о связях, получая в конце-концов информацию о состоянии всех связей сети. Эта информация и образует граф связей сети, который, естественно, один и тот же для всех маршрутизаторов сети.

Кроме информации о соседях, маршрутизатор в своем объявлении перечисляет IP-подсети, с которыми он связан непосредственно, поэтому после получения информации о графе связей сети, вычисление маршрута до каждой сети производится непосредственно по этому графу по алгоритму Дэйкстры. Более точно, маршрутизатор вычисляет путь не до конкретной сети, а до маршрутизатора, к которому эта сеть подключена. Каждый маршрутизатор имеет уникальный идентификатор, который передается в объявлении о состоянии связей. Маршрутизатор вычисляет оптимальный маршрут до каждой адресуемой сети, но запоминает только первый промежуточный маршрутизатор из каждого маршрута. Таким образом, результатом вычислений оптимальных маршрутов является список строк, в которых указывается номер сети и идентификатор маршрутизатора, которому нужно переслать пакет для этой сети. Указанный список маршрутов и является маршрутной таблицей.

Содержание отчёта:

1. Титульный лист.
2. Цель работы и задание.
3. Скриншоты основных этапов работы.
4. Анализ полученных результатов и выводы по работе .

Контрольные вопросы:

1. Обоснование реализации разработанной топологии сети.
2. Протокол динамической маршрутизации OSPF. Описание работы протокола.
3. Сравнительный анализ версий протокола OSPF.

Лабораторная работа № 4.

Захват и анализ трафика, проходящий по связям эмулируемой внутри GNS3 топологии.

Цель работы:

Анализ работы алгоритмов протоколов и трафика, проходящего внутри сети.

Задание:

Используя топологии из лабораторной работы №2 (RIP маршрутизация) и работы №3 (OSPF маршрутизация), проанализировать внутренний трафик сетей (сравнить теоретическую работу протоколов и состав сообщений с реальным моделированием топологий сетей). Составить сравнительный анализ протоколов маршрутизации.

Теоретическая часть:

Wireshark - это сетевой анализатор с графическим интерфейсом. Позволяет в интерактивном режиме просматривать пакеты, передаваемые по сети или анализировать ранее захваченные пакеты, загрузив их из сохраненного файла.

После того как был установлен WireShark, для перехвата пакетов в любой предварительно созданной топологии, нужно на линии соединяющей оборудование вызвать контекстное меню, в котором выбрать Capture (Захват). После чего еще раз вызвать это же меню, в котором выбрать Start WireShark. Сразу после этого будет запущен WireShark и, все проходящие внутри данной линии данные будут отображаться в реальном времени.

Содержание отчёта:

1. Титульный лист
2. Цель работы и задание
3. Скриншоты основных этапов работы
4. Анализ полученных результатов и выводы по работе

Контрольные вопросы:

1. Дать определение анализатору трафика.
2. Перечислить цели, для которых применяется сетевой анализатор.
3. Форматы сообщения протоколов RIP и OSPF и их версий.
4. Сравнительный анализ протоколов.

Библиографический список

1. Олифер В. Г., Олифер Н. А Компьютерные сети. Принципы, технологии, - протоколы. - СПб: Питер, 2013. -944с.
2. Савин Е.З. Волоконно оптические кабели и пассивные компоненты ВОЛП: учеб. Пособие. – М.: Изд. УМЦ ЖДТ, 2012. -223 с.
3. А. Б. Семёнов, Волоконно-оптические подсистемы современных структурированных кабельных систем. – АйТи Пресс, 2007. – 630с.
4. А. Б. Семёнов, Администрирование структурированных кабельных систем. – АйТи Пресс, 2013. – 192с.

5. Н. В. Максимов, И. И. Петров – Компьютерные сети: Учебное пособие – М.:ФОРУМ: ИНФА-М, 2005. – 336с.: ил.
6. Гук М. Аппаратные средства локальных сетей. Энциклопедия – СПб: Издательство «Питер», 2000. -439 с.
7. Cisco Systems. Руководство по технологиям объединенных сетей.2002 г.

СОДЕРЖАНИЕ

Введение.....	3
<i>Лабораторная работа №1.</i>	
GNS3 Симулятор сетей Cisco. Установка и предварительная настройка симулятора GNS3, создание тестовой топологии.....	4
<i>Лабораторная работа №2.</i>	
Подключение и администрирование маршрутизаторов Cisco, основы операционной системы Cisco IOS, статическая маршрутизация, настройка протокола RIP.....	11
<i>Лабораторная работа №3.</i>	
Настройка маршрутизации по протоколу OSPF.....	26
<i>Лабораторная работа №4.</i>	
Захват и анализ трафика, проходящий по связям эмулируемой внутри GNS3 топологии.....	27

Подписано в печать 14.03.20. Формат 84x108/32
Гарнитура Таймс. Печать офсетная.
Бумага мелованная. Усл. Печ. л. – 1,58.
Тираж 50 экз.

Издательство Современного технического
университета 390048, г. Рязань, ул.
Новоселов, 35А.

(4912) 300630, 30 08 30