

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ**

Факультет _____ компьютерных технологий и управления _____
Кафедра _____ вычислительной техники _____
Направление подготовки (специальность) _____ 230100 _____

**О Т Ч Е Т
о практике**

Тема задания: _____ Проектирование компьютерной сети в cisco packer tracer _____

Студенты _____ Копылов А.А., Ванцян Р.Н., группа 3103 _____

Руководитель практики _____ Соснин В.В. _____

Оценка руководителя _____

Дата _____

Санкт-Петербург
2013г.

Оглавление

1	TeX(LaTeX)	3
1.1	MiKTeX	4
2	Git	4
3	Cisco Packet Tracer	7
3.1	Создание простой топологии сети	10
3.1.1	Добавление маршрутизатора в проект	10
3.1.2	Добавление коммутатора в проект	12
3.1.3	Добавление конечных узлов сети	12
3.1.4	Соединение устройств	13
3.2	Пример простой компьютерной сети	13
3.2.1	Статическая маршрутизация	13
3.2.2	Динамическая маршрутизация	15
3.3	Протоколы	16
3.3.1	DHCP	16
3.3.2	NAT	18
3.4	VLANs, Виртуальные сети	25
3.5	Запароливание маршрутизаторов	30

1 TeX(LaTeX)

TeX - система компьютерной вёрстки текста, разработанная американским профессором информатики Дональдом Кнутом в 1979-м году.

Атрибутом TeX/LaTeX является высочайшее типографское качество результирующего текста. Например, десятки разновидностей пробелов, дефисов и тире, доступные в современных реализациях LaTeX.

Для написания TeX Дональд Кнут создал концепцию «литературного программирования», заключающуюся в генерации документации и исходного кода программы по тексту, язык которого близок к естественному языку.

Недостатком же системы TeX является необходимость работать со значительно более абстрактным, лишенным наглядности представлением текста, что осложняет первоначальное освоение системы и на этапе подготовки текста нередко приводит к ошибкам, выявляющимся лишь при компиляции исходного файла.

TeX принимает исходный .tex-файл, интерпретирует его содержимое, и производит .dvi-файл, который представляет собой постраничное изображение результирующего документа. Рисунки включаются в документ в виде ссылок на внешние файлы. В результирующем .dvi-файле TeX оставляет для них пустые места.

LaTeX - наиболее популярный набор макрорасширений системы компьютерной вёрстки TeX, который облегчает набор сложных документов. Пакет позволяет автоматизировать многие задачи набора текста и подготовки статей, включая набор текста на нескольких языках, нумерацию разделов и формул, перекрёстные ссылки, размещение иллюстраций и таблиц на странице, ведение библиографии и др.

1.1 MiKTeX

MiKTeX - открытый (open source) дистрибутив \TeX для платформы Windows. Одним из существенных достоинств MiKTeX является возможность автоматического обновления установленных компонентов и пакетов.

В состав MiKTeX включены:

- классический \TeX -компилятор;
- различные варианты \TeX : pdf \TeX , e- \TeX , pdf-e- \TeX , Omega, e-Omega, NTS;
- конверторы \TeX в PDF;
- MetaPost;
- полный набор общеиспользуемых макропакетов: LaTeX, ConTeXt и др.;
- средство просмотра Yip;
- инструменты и утилиты;

2 Git

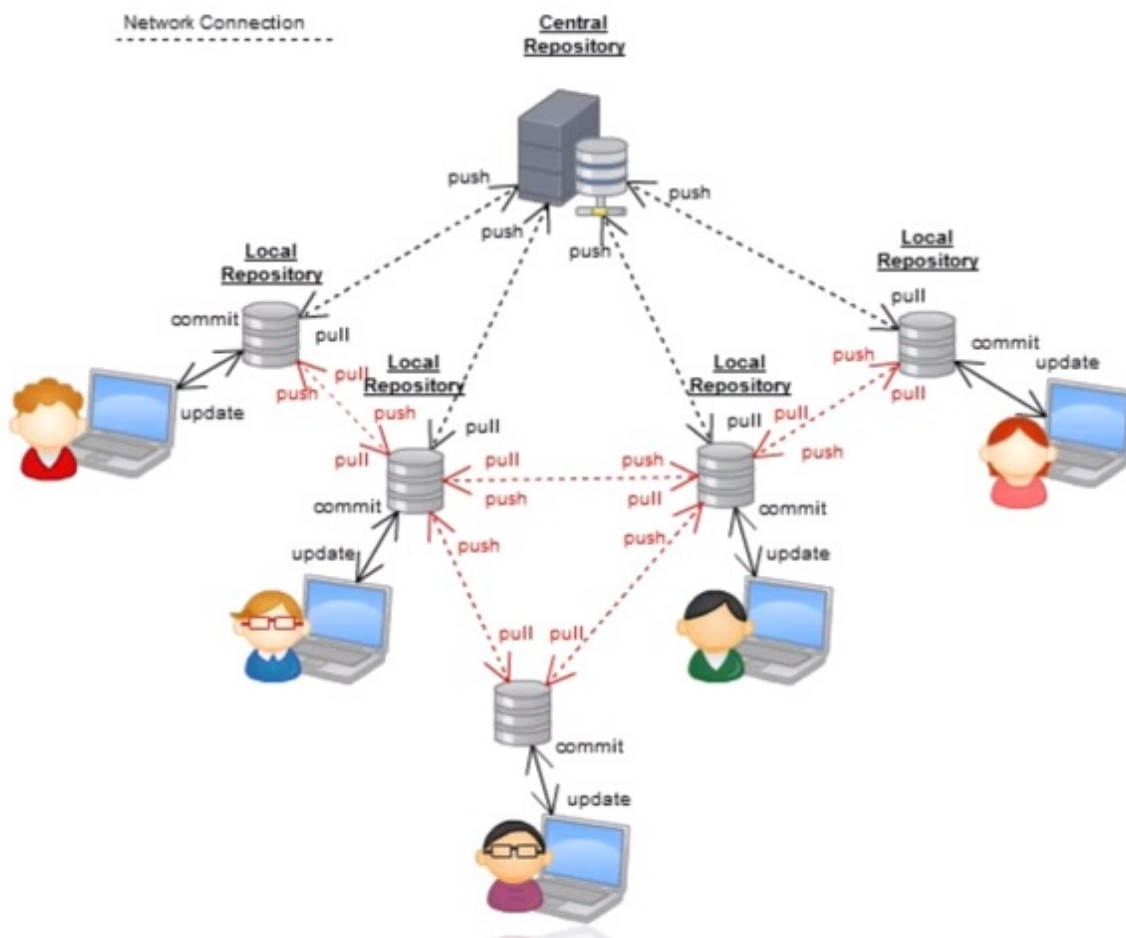
Git - система контроля версий (т.е. такая система, которая хранит все изменения в файле). В любой момент можно откатиться до любой заданной заранее позиции.

После изменений в файле, чтобы сохранить изменения, нужно сделать commit (фиксация). В commit может входить как один, так сразу и несколько файлов.

При работе в команде другим членам необходимо видеть то, что сделали Вы и использовать Ваши изменения. А Вам необходимо использовать изменения других участников команды. Для этого будет необходима система контроля версий.

Системы контроля версий хранят код в репозитории. Репозиторий является именно тем местом, которое синхронизирует различия, загруженные разными участниками процесса в коде.

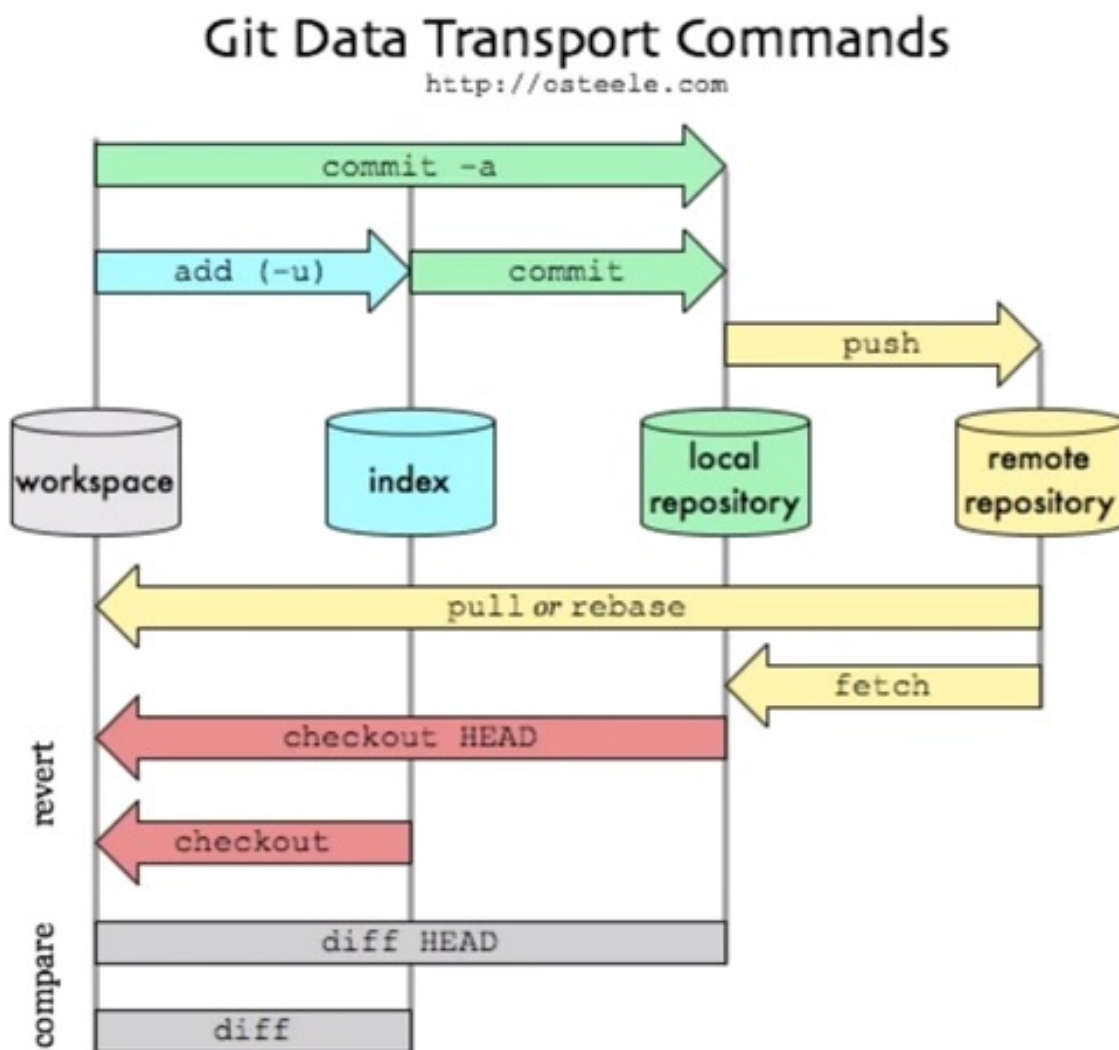
В Git между главным репозиторием и пользователем существует промежуточный репозиторий. Именно в этом заключается отличие Git от других систем. Локальный репозиторий обеспечивает работу без интернета.



Схематически Git работает следующим образом:

Вы работаете в своем рабочем пространстве (workspace), система параллельно с вашими изменениями вносит изменения в главный индекс(index), следя за теми файлами, которые Вы добавите (с помощью команды add). После этого можно зафиксировать текущее состояние с помощью команды commit. Затем файлы добавляются в локальный репозиторий. Наконец, когда у Вас есть доступ к интернету, Вы переносите (проталкиваете) фай-

лы (push) в удаленный репозиторий (у нас он находится на github). После чего из удаленного репозитория можно загрузить файлы с текущими изменениями(pull) либо посмотреть изменения с текущей версией (diff) либо другие пользователи (или Вы, но в другую директорию) могут загрузить изменения(checkout).



Основные git-команды:

git init - создание репозитория.

git status - состояние проекта; выводит информацию обо всех изменениях, внесенных в дерево директорий проекта по сравнению с последним коммитом рабочей ветки; отдельно выводятся внесенные в индекс и неиндексированные файлы.

git add - индексация изменений; позволяет внести в индекс - временное хранилище - изменения, которые затем войдут в коммит.

git commit - совершение коммита.

Ключи:

git commit -a - совершит коммит, автоматически индексируя изменения в файлах проекта. Новые файлы при этом индексироваться не будут! Удаление же файлов будет учтено.

git commit -m «commit comment» - комментируем коммит прямо из командной строки вместо текстового редактора.

git commit "filename" - внесет в индекс и создаст коммит на основе изменений единственного файла.

git log - разнообразная информация о коммитах в целом, по отдельным файлам и различной глубины погружения в историю.

git diff - изменения, не внесенные в индекс.

git push - вносим изменения в удаленный репозиторий.

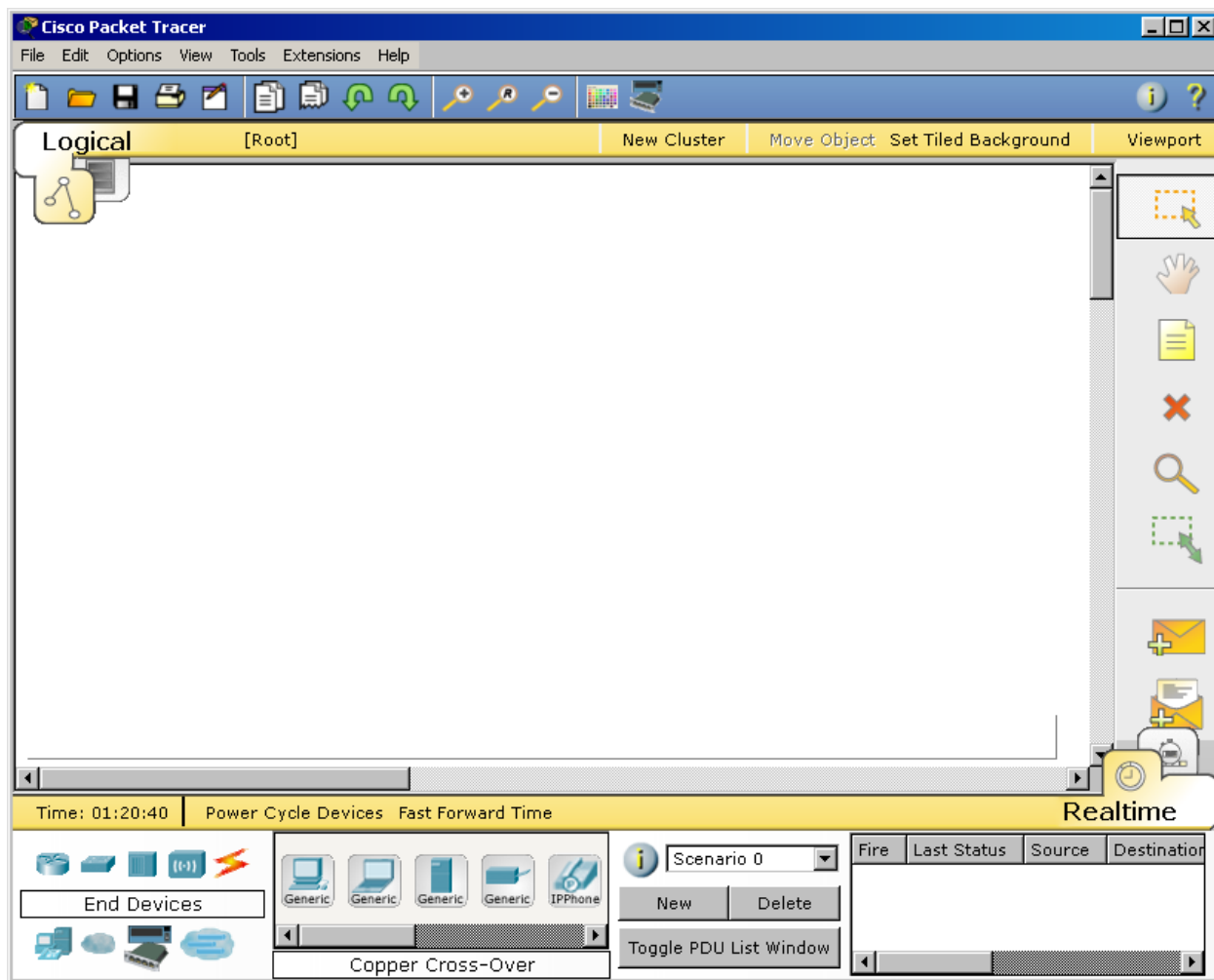
git pull - забираем изменения из удаленного репозитория.

Использованные материалы: <http://git-scm.com/book/ru/>,
<http://githowto.com/ru/>

3 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.

После запуска симулятора Packet Tracer, перед нами главное окно программы.



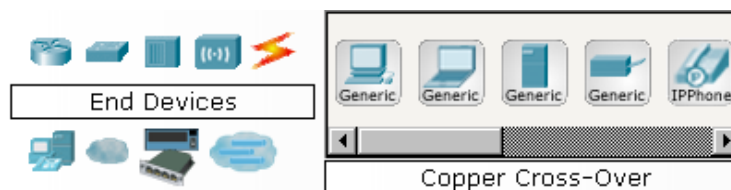
Большую часть данного окна занимает рабочая область, в которой можно размещать различные сетевые устройства, соединять их различными способами и как следствие получать самые разные сетевые топологии.

Сверху, над рабочей областью, расположена главная панель программы и ее меню. Меню позволяет выполнять сохранение, загрузку сетевых топологий, настройку симуляции, а также много других интересных функций. Главная панель содержит на себе наиболее часто используемые функции меню.



Справа от рабочей области, расположена боковая панель, содержащая ряд кнопок отвечающих за перемещение полотна рабочей области, удаление объектов и т.д.

Снизу, под рабочей областью, расположена панель оборудования.



Данная панель содержит в своей левой части типы доступных устройств, а в правой части доступные модели.

При наведении на каждое из устройств, в прямоугольнике, находящемся в центре между ними будет отображаться его тип. Типы устройств, наиболее часто используемые в Packet Tracer, представлены на рисунке.



Типы соединений (рассмотрение типов подключений идет слева направо, в соответствии с приведенным снизу рисунком).



- Автоматический тип - при данном типе соединения PacketTracer автоматически выбирает наиболее предпочтительные тип соединения для выбранных устройств
- Консоль - консольные соединения
- Медь Прямое - соединение медным кабелем типа витая пара, оба конца

кабеля обжаты в одинаковой раскладке. Подойдет для следующих соединений: коммутатор - коммутатор, коммутатор - маршрутизатор, коммутатор - компьютер и др.

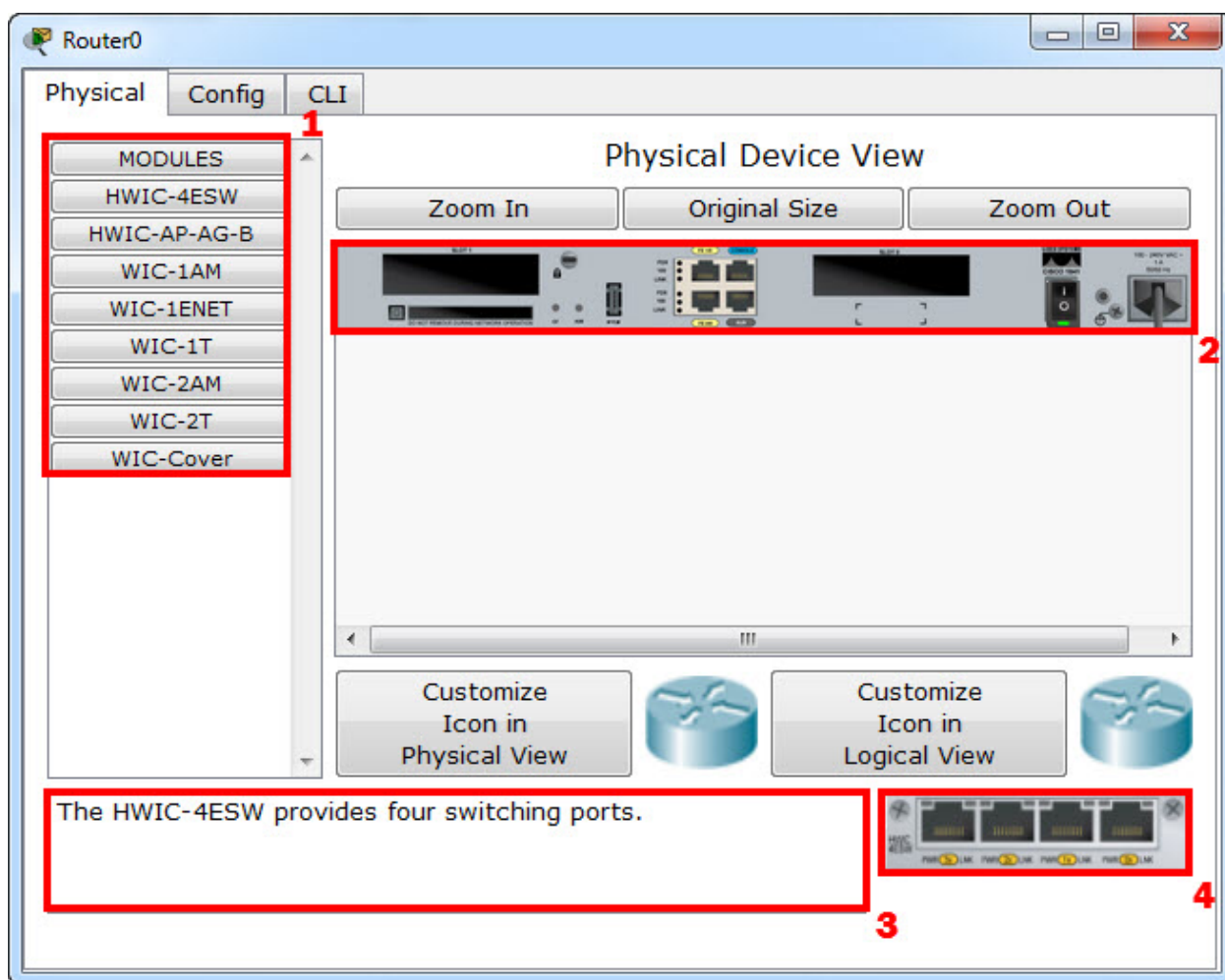
- Медь кроссовер - соединение медным кабелем типа витая пара, концы кабеля обжаты как кроссовер. Подойдет для соединения двух компьютеров, компьютер-роутер, роутер-роутер.
- Оптика - соединение при помощи оптического кабеля, необходимо для соединения устройств имеющих оптические интерфейсы.
- Телефонный кабель - обыкновенный телефонный кабель, может понадобиться для подключения телефонных аппаратов.
- Коаксиальный кабель - соединение устройств с помощью коаксиального кабеля.

3.1 Создание простой топологии сети

3.1.1 Добавление маршрутизатора в проект

Для того чтобы добавить маршрутизатор в проект сети, необходимо выбрать левым кликом мыши данный тип оборудования, также выбрать модель и добавить в проект, кликнув на рабочем поле программы.

После того как оборудование добавлено в проект, можно открыть окно параметров данного устройства которое предоставляет возможность доступа к аппаратной конфигурации данного юнита, а также возможность конфигурации средствами IOS CLI или меню.



Активная по умолчанию вкладка "Physical" позволяет управлять аппаратной конфигурацией маршрутизатора и отображает следующие элементы:

- 1- список доступных для установки модулей;
- 2- внешний вид оборудования;
- 3- описание выбранного модуля;
- 4- внешний вид выбранного модуля.

Любые операции по добавлению или удалению модулей производятся только на выключенном оборудовании. В связи с этим, виртуальное оборудование должно быть отключено от сети кнопкой питания перед установкой или удалением сетевого модуля.

Вкладка CLI предоставляет доступ к консоли Console 0 маршрутизатора. По умолчанию на доступ к консоли пароль не установлен.

3.1.2 Добавление коммутатора в проект

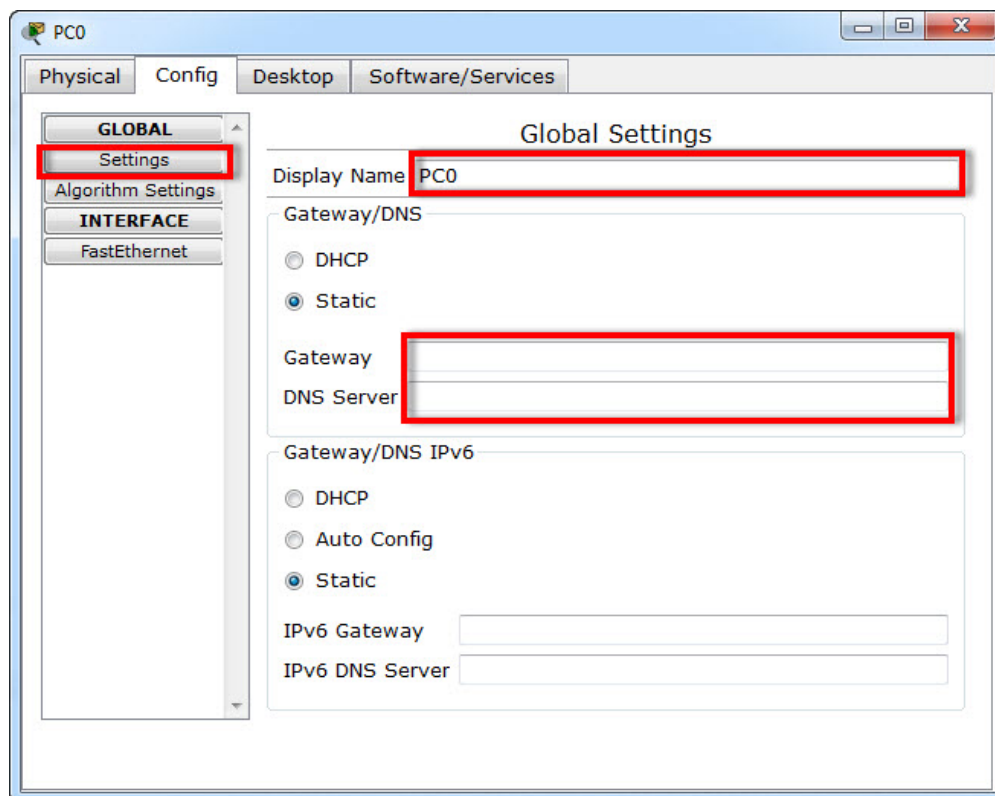
Добавление коммутатора в проект практически идентично добавлению маршрутизатора. Отличия в начальной стадии при выборе панели и в самом окне параметров добавленного устройства.

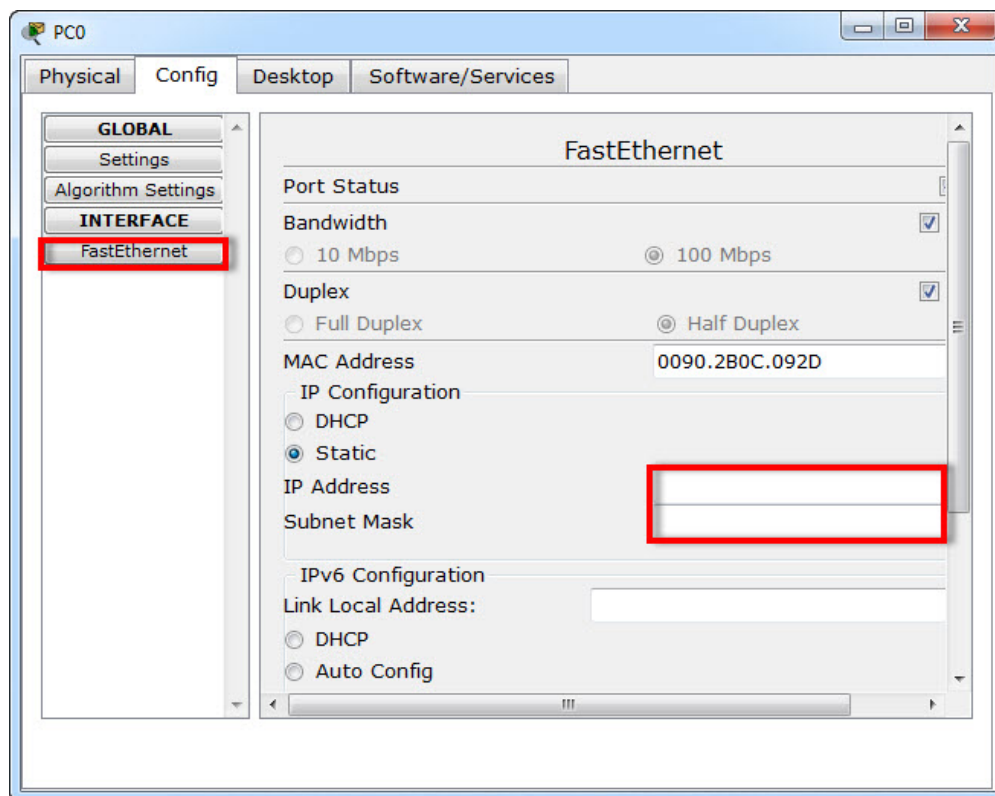
3.1.3 Добавление конечных узлов сети

Конечные узлы сети, такие как сервера, рабочие станции и ноутбуки, добавляются в топологию идентично другим устройства проекта.

Окно параметров конечных устройств варьируется в зависимости от самого устройства. Некоторые различия сразу заметны.

Параметры сетевых интерфейсов для данного вида устройств устанавливаются через меню на вкладке Config. Данная вкладка практически идентична для перечисленных выше устройств. Сетевые параметры указываются в меню Settings и меню свойств сетевого интерфейса (FastEthernet) вкладки Config.





3.1.4 Соединение устройств

После того как все необходимые для работы устройства добавлены в проект, необходимо все единицы оборудования соединить между собой. Для этого используется меню Connections.

Выбор кабеля зависит от подключаемого оборудования и технологии подключения. Каждый раз при соединении оборудования будет предлагаться выбор интерфейса, если таковые есть в наличии и не участвуют в другом подключении.

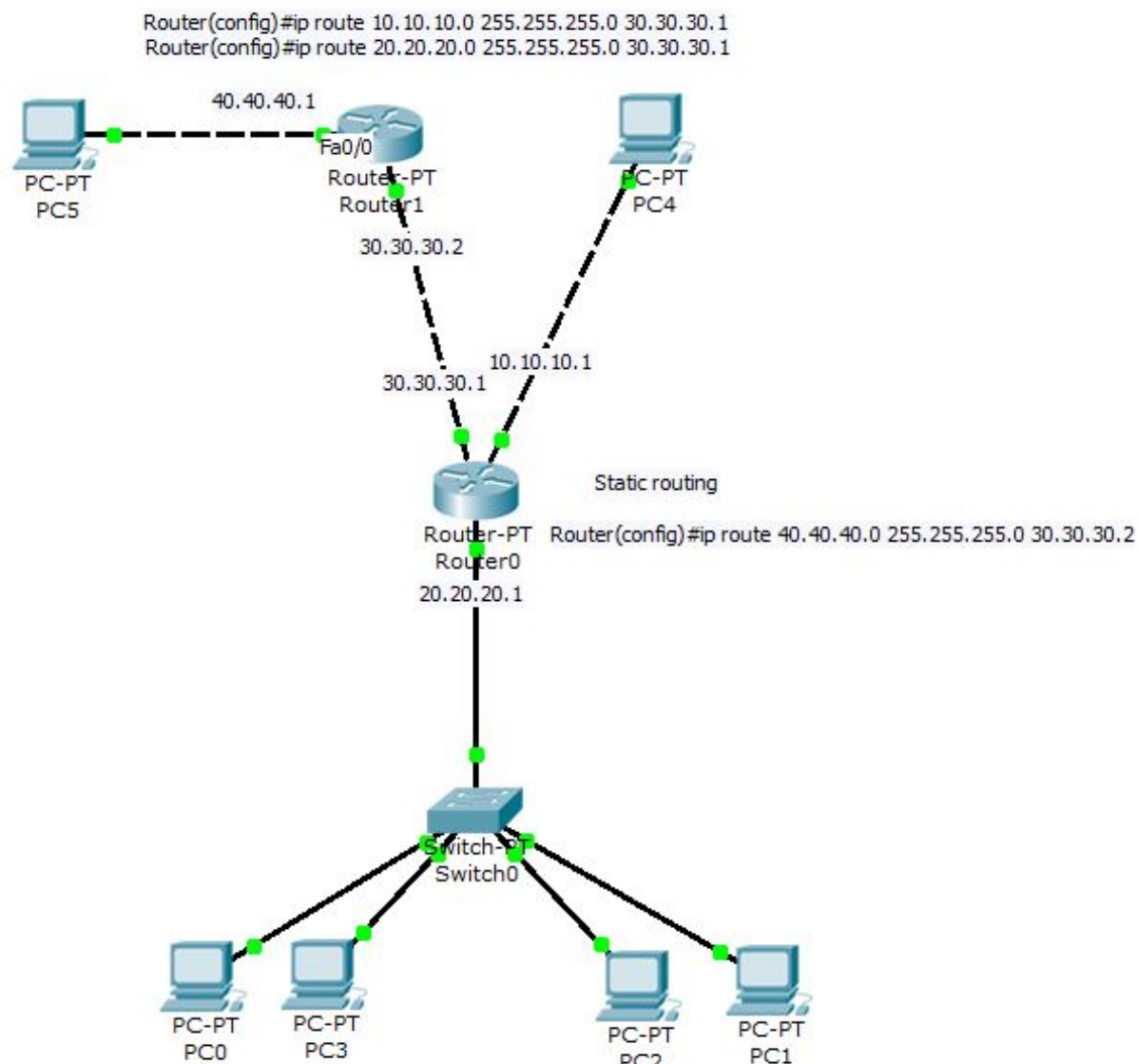
3.2 Пример простой компьютерной сети

Находится в файле easy_compset.pkt

3.2.1 Статическая маршрутизация

Статическая маршрутизация - вид маршрутизации, при котором маршруты указываются в явном виде при конфигурации маршрутизатора.

#ip route “Network” “Mask” “Next Hop”



В данном примере маршрутизатор Router1 «не знает» о сетях 10.10.10.0 и 20.20.20.0, т.к. он находится в сетях 40.40.40.0 и 30.30.30.0, поэтому на нем нужно настроить маршрутизацию, для этого пишем:

```
#ip route 10.10.10.0 255.255.255.0 30.30.30.1
```

```
#ip route 20.20.20.0 255.255.255.0 30.30.30.1
```

В свою очередь маршрутизатор Router0 «не знает» о сети 40.40.40.0 и поэтому мы должны ей сами «рассказать» о данной сети:

```
#ip route 40.40.40.0 255.255.255.0 30.30.30.2
```

3.2.2 Динамическая маршрутизация

Динамическая маршрутизация - вид маршрутизации, при котором таблица маршрутизации редактируется программно.

Динамическая маршрутизация используется для общения маршрутизаторов друг с другом. Маршрутизаторы передают друг другу информацию о том, какие сети в настоящее время подключены к каждому из них. Маршрутизаторы общаются, используя протоколы маршрутизации. В данном случае мы используем протокол RIP. Настройка состоит лишь в том, чтобы занести в таблицу каждого маршрутизатора имя сети, к которой он подключён. Для этого:

```
#router rip
```

```
#network 44.0.0.0
```

```
#network 22.0.0.0
```

```
#network 33.0.0.0
```

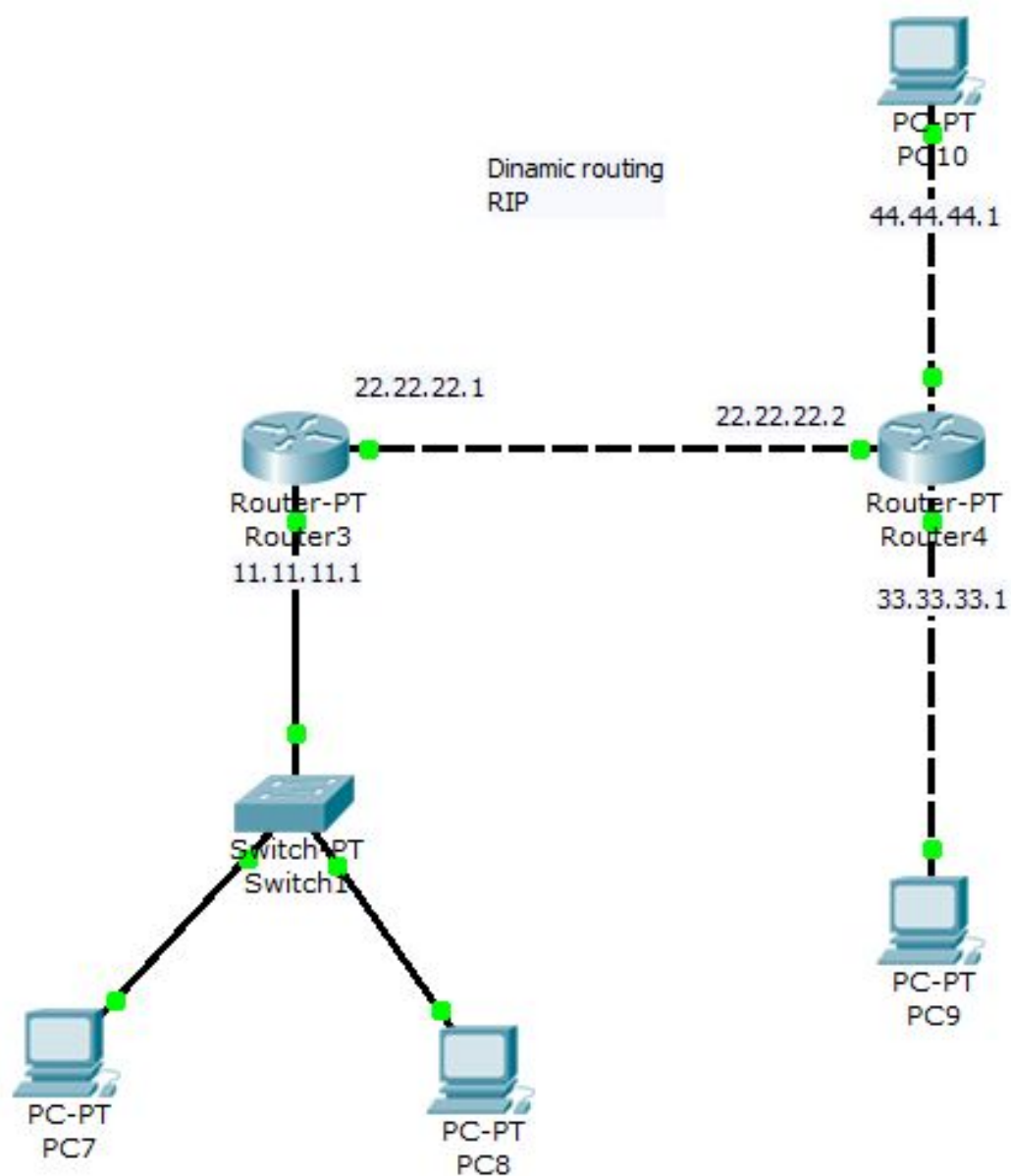
Для Router4 и

```
#router rip
```

```
#network 11.0.0.0
```

```
#network 22.0.0.0
```

Для Router3



3.3 Протоколы

3.3.1 DHCP

DHCP - сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети.

Добавлена новая сеть:

192.168.10.0 255.255.255.0

На Router3 настроили DHCP присвоение IP адресов из диапазона:

192.168.10.10 - 192.168.10.254 Для этого на Router3:

```
ip dhcp pool net1
```

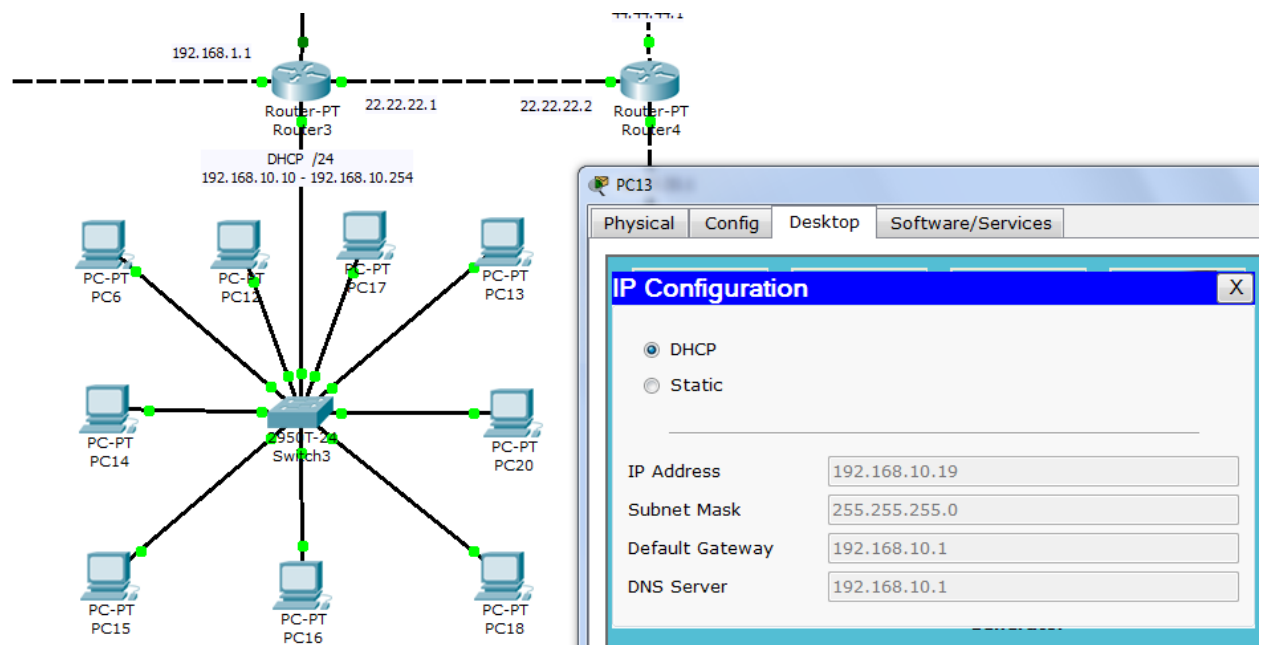
```
network 192.168.10.0 255.255.255.0
```

```
default-router 192.168.10.1
```

```
ip dhcp excluded-address 192.168.10.1 192.168.10.10
```

После чего, на всех ПК в IP Configurations выставляем поле DHCP.

На Router3 с помощью команды `show ip dhcp binding` видим результат работы DHCP сервера.



```
Router#show ip dhcp binding
```

IP address	Client-ID/ Hardware address	Lease expiration	Type
192.168.10.16	0090.2B53.68C1	--	Automatic
192.168.10.12	00D0.BABA.08B9	--	Automatic
192.168.10.13	0060.70CD.D639	--	Automatic
192.168.10.14	0090.216E.CA96	--	Automatic
192.168.10.15	0001.6320.A440	--	Automatic
192.168.10.17	000A.4170.46D7	--	Automatic
192.168.10.18	00E0.8F67.9E45	--	Automatic
192.168.10.19	00D0.5811.B83D	--	Automatic
192.168.10.20	000C.85BB.7559	--	Automatic
192.168.10.11	0001.6431.9620	--	Automatic

Router#

3.3.2 NAT

Network Address Translation — механизм, который позволяет транслировать в один "внешний" ip-адрес много внутренних адресов.

Дело в том, что теоретически существует $255*255*255*255=4\ 228\ 250\ 625$. 4 миллиарда адресов. С распространением всемирной паутины адреса IPv4 исчерпываются. И люди придумали технологию, которая временно поможет справиться с этой проблемой. Так, было предложено разделить пространство адресов на публичные (белые, внешние) и приватные (частные, серые).

К последним относятся три диапазона:

10.0.0.0/8

172.16.0.0/12

192.168.0.0/16

Их можно свободно использовать в своей частной сети, и поэтому, разумеется, они не должны быть уникальными и будут повторяться. Как же быть с уникальностью во внешнем пространстве?

Тут нам поможет NAT. Рассмотрим на примере сети 40.40.40.0, в котором есть один хост с ip-адресом 40.40.40.2, внутренний порт маршрутизатора: 40.40.40.1, внешний: 30.30.30.2.

Чтобы настроить NAT необходимо:

1) Указать "Внешний" и "Внутренний" интерфейсы маршрутизатора на котором настраиваем NAT.

```
interface fastethernet 0/0
```

```
ip nat inside
```

```
exit
```

```
interface fastethernet 1/0
```

```
ip nat outside
```

2) Далее создаём разрешающее правило с именем "5":

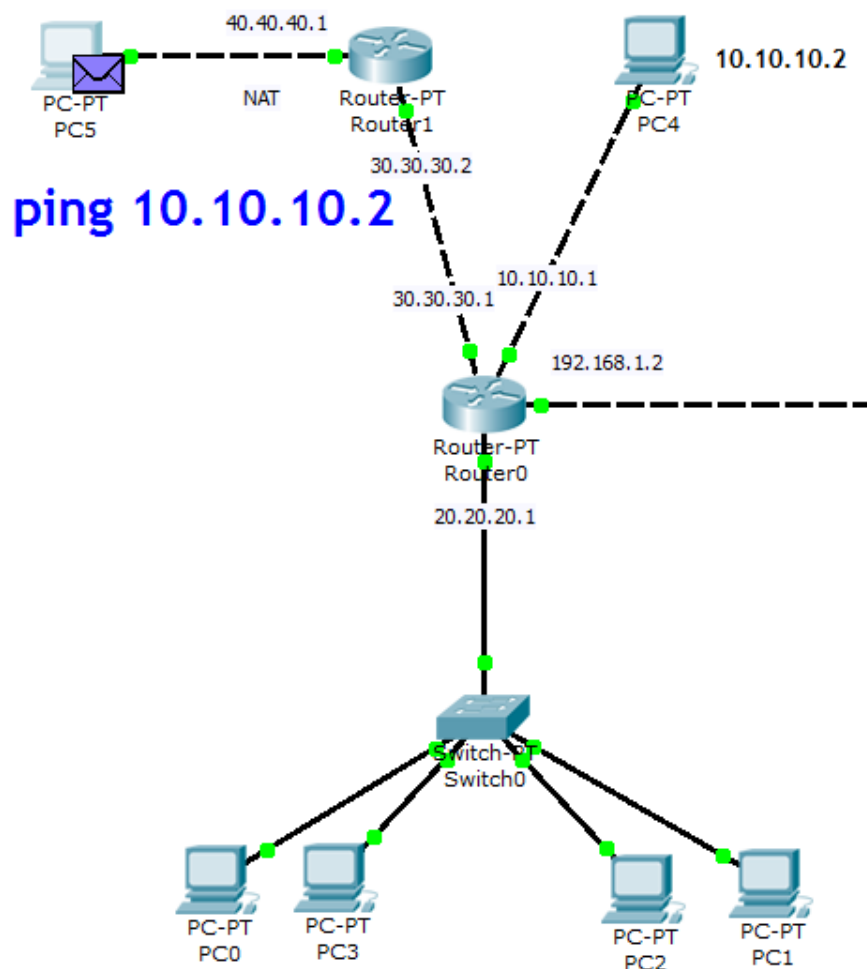
```
access-list 5 permit any
```

3) Далее привязываем созданное правило к внешнему интерфейсу:

```
ip nat inside source list 5 interface fastethernet 1/0 overload
```

Что же будет происходить, когда мы будем пинговать с нашего 40.40.40.1 какой-нибудь внешний адрес, например: 10.10.10.2?

Формируем эхо запрос. В командной строке PC5 набираем ping 10.10.10.2.



Далее переключаемся на режим "симуляции" в правом нижнем углу и наблюдаем процесс шаг за шагом.

Mar1:

PDU Information at Device: PC5

OSI Model

Outbound PDU Details

At Device: PC5
Source: PC5
Destination: PC4

In Layers

Layer7

Layer6

Layer5

Layer4

Layer3

Layer2

Layer1

Out Layers

Layer7

Layer6

Layer5

Layer4

Layer 3: IP Header Src. IP: 40.40.40.2,
Dest. IP: 10.10.10.2 ICMP Message
Type: 8

Layer 2: Ethernet II Header
0010.11CE.0427 >> 0002.4A0B.9ECD

Layer 1: Port(s): FastEthernet

1. The Ping process starts the next ping request.

2. The Ping process creates an ICMP Echo Request message and sends it to the lower process.

3. The source IP address is not specified. The device sets it to the port's IP address.

4. The device sets TTL in the packet header.

5. The destination IP address is not in the same subnet and is not the broadcast address.

6. The default gateway is set. The device sets the next-hop to default gateway.

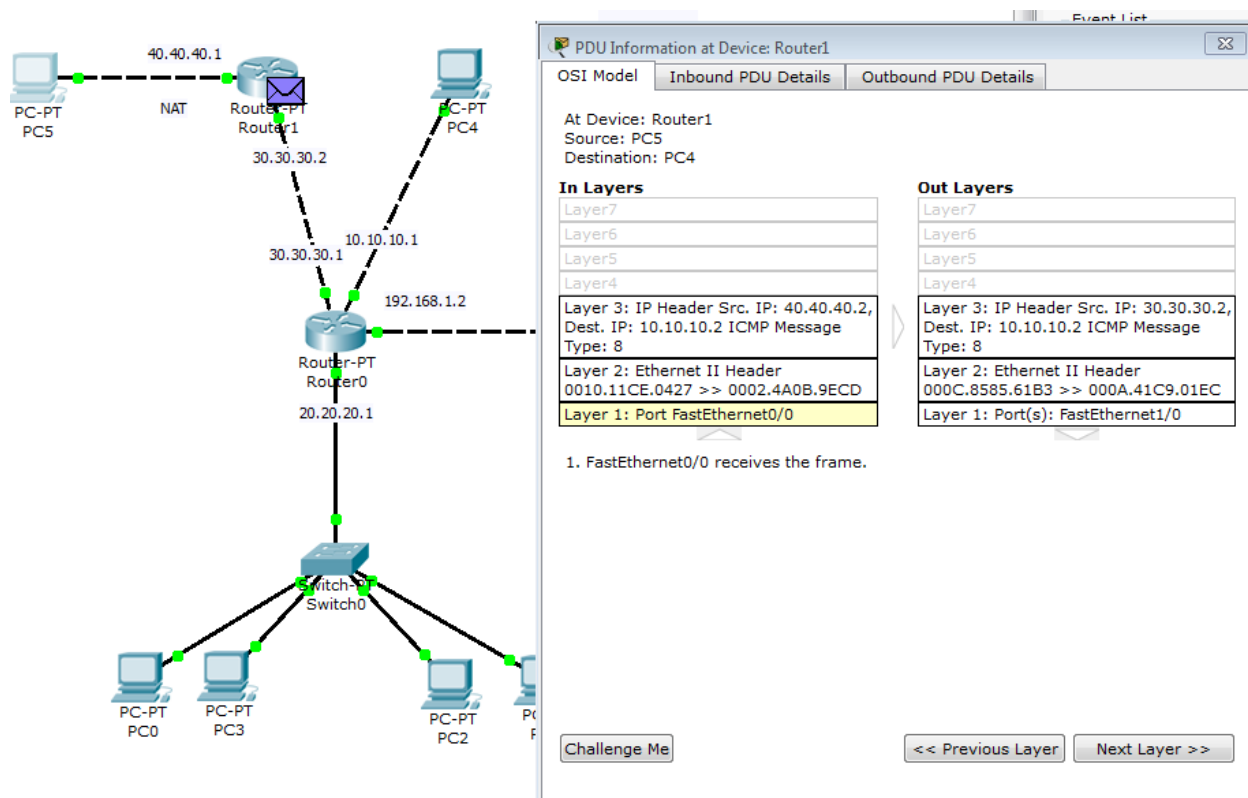
Challenge Me

<< Previous Layer

Next Layer >>

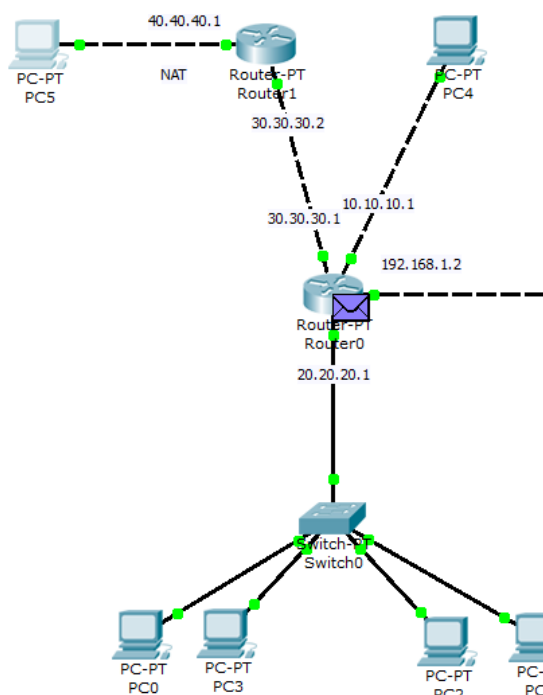
20

Шаг2:



Как видим, ip.src на данном шаге поменялся с 40.40.40.2 на 30.30.30.2. Т.е. натирующее устройство "Router1" заменяет в заголовке IP пакета адрес источника и вставляет свой адрес.

Шаг3:



PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

At Device: Router0
Source: PC5
Destination: PC4

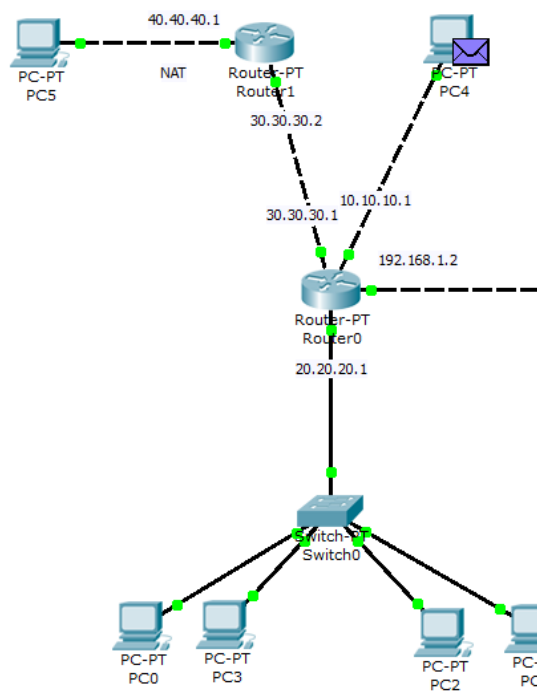
In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 30.30.30.2, Dest. IP: 10.10.10.2 ICMP Message Type: 8	Layer 3: IP Header Src. IP: 30.30.30.2, Dest. IP: 10.10.10.2 ICMP Message Type: 8
Layer 2: Ethernet II Header 000C.8585.61B3 >> 000A.41C9.01EC	Layer 2: Ethernet II Header 00D0.D36A.04E8 >> 0060.70A3.0C03
Layer 1: Port FastEthernet1/0	Layer 1: Port(s): FastEthernet0/0

1. FastEthernet1/0 receives the frame.

Challenge Me << Previous Layer Next Layer >>

Шаг4:

Получив запрос от 30.30.30.2, PC4 отправляет ICMP reply на адрес 30.30.30.2 (направляющее устройство)



PDU Information at Device: PC4

OSI Model | Inbound PDU Details | Outbound PDU Details

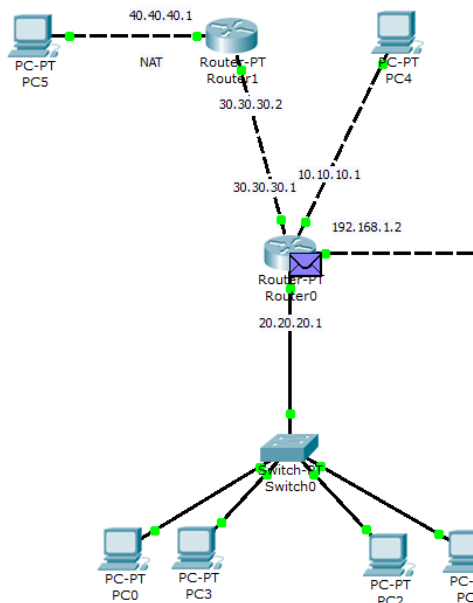
At Device: PC4
Source: PC5
Destination: PC4

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 30.30.30.2, Dest. IP: 10.10.10.2 ICMP Message Type: 8	Layer 3: IP Header Src. IP: 10.10.10.2, Dest. IP: 30.30.30.2 ICMP Message Type: 0
Layer 2: Ethernet II Header 00D0.D36A.04E8 >> 0060.70A3.0C03	Layer 2: Ethernet II Header 0060.70A3.0C03 >> 00D0.D36A.04E8
Layer 1: Port FastEthernet	Layer 1: Port(s): FastEthernet

1. FastEthernet receives the frame.

Challenge Me << Previous Layer Next Layer >>

Mar5:



PDU Information at Device: Router0

OSI Model | Inbound PDU Details | Outbound PDU Details

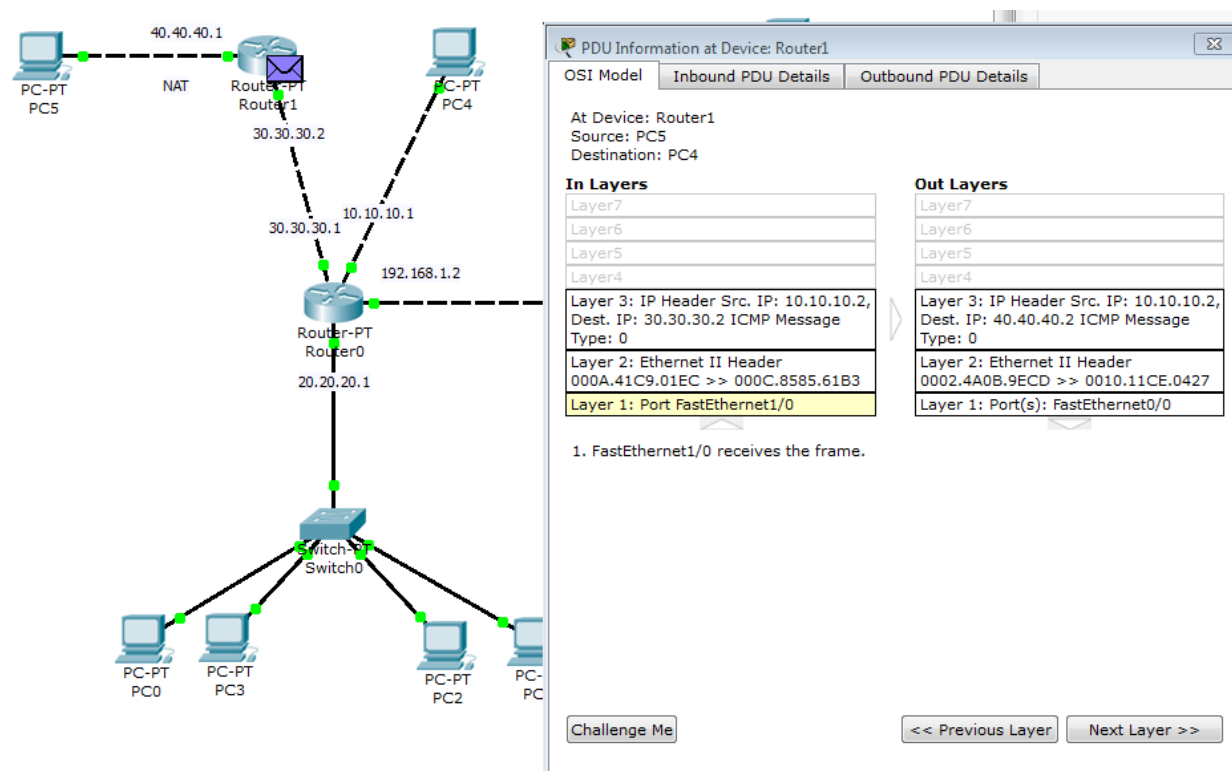
At Device: Router0
Source: PC5
Destination: PC4

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 10.10.10.2, Dest. IP: 30.30.30.2 ICMP Message Type: 0	Layer 3: IP Header Src. IP: 10.10.10.2, Dest. IP: 30.30.30.2 ICMP Message Type: 0
Layer 2: Ethernet II Header 0060.70A3.0C03 >> 00D0.D36A.04E8	Layer 2: Ethernet II Header 000A.41C9.01EC >> 000C.8585.61B3
Layer 1: Port FastEthernet0/0	Layer 1: Port(s): FastEthernet1/0

1. FastEthernet0/0 receives the frame.

Challenge Me << Previous Layer Next Layer >>

Шаг6:

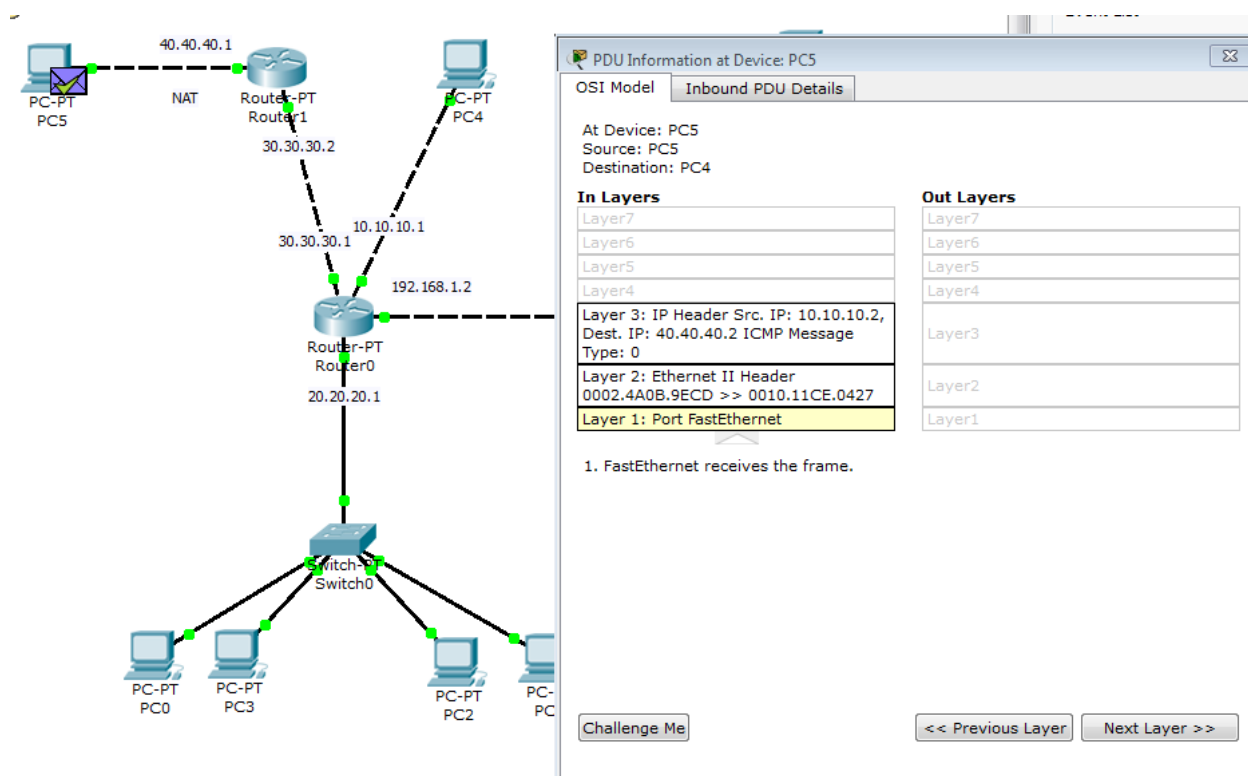


На этом же шаге происходит обратная замена, IP получателя меняется с 30.30.30.2 на 40.40.40.2 (PC5).

Шаг7:

PC5 благополучно получил ответ от PC4 и даже не подозревает о "магических" преобразованиях, которые происходили с его пакетами в сети.

Стоит отметить, что PC5 скрывается с помощью технологии NAT от внешних запросов.



3.4 VLANs, Виртуальные сети

Технология VLAN является одной из главных технологий коммутации в сетях Ethernet. Создание виртуальных локальных сетей позволило «виртуально» разделять пользователей, которые подсоединены к одному коммутатору на несколько групп, которые не общаются между собой по средствам коммутации.

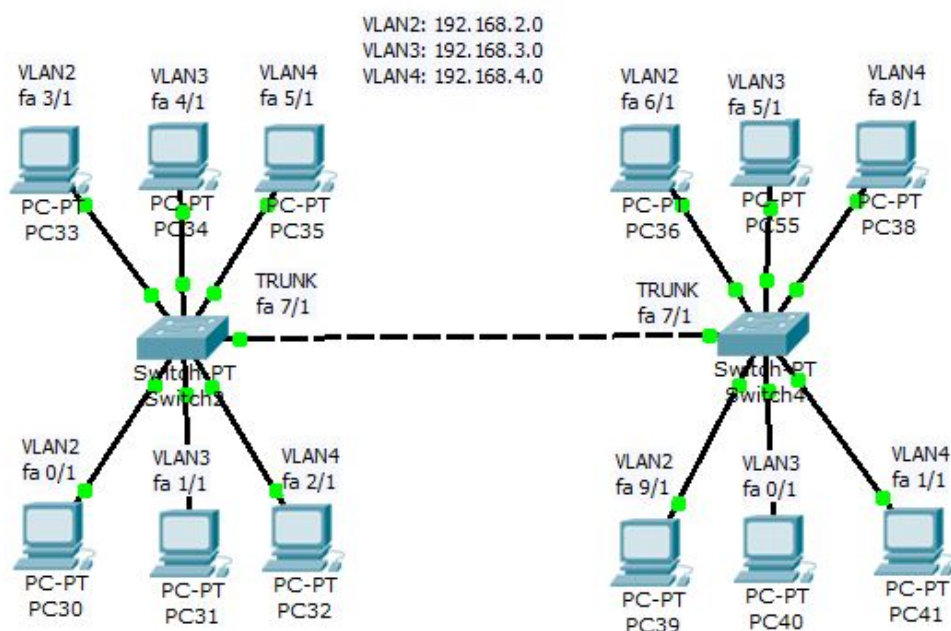
VLANs существуют на втором уровне модели OSI. То есть, «виланы» можно настроить на коммутаторе второго уровня. По сути VLAN – это просто метка в кадре, который передается по сети. Метка содержит номер «виллана» (VLAN ID или VID). На портах коммутаторов указывается в каком VLAN они находятся. В зависимости от этого весь трафик, который выходит через порт помечается меткой, то есть VLAN. Таким образом каждый порт имеет PVID (port vlan identifier). Этот трафик может в дальнейшем проходить через другие порты коммутатора(ов), которые находятся в этом VLAN и не пройдут через все остальные порты. В итоге, создается

изолированная среда (подсеть), которая без дополнительного устройства (маршрутизатора) не может взаимодействовать с другими подсетями.

Зачем нужны такие виртуальные сети?

- Возможность построения сети, логическая структура которой не зависит от физической. То есть, топология сети на канальном уровне строится независимо от географического расположения составляющих компонентов сети.
- Возможность обезопасить сеть от несанкционированного доступа. То есть, на канальном уровне кадры с других виланов будут отсекаются портом коммутатора независимо от того, с каким исходным IP-адресом инкапсулирован пакет в данный кадр.
- Возможность применять политики на группу устройств, которые находятся в одном вилане.
- Возможность использовать виртуальные интерфейсы для маршрутизации.

Создадим сеть, используя данную технологию. На рисунке приведена физическая топология сети. Сверху подписаны номера портов и имена VLAN.



Имеем два коммутатора Switch2 и Switch4. В конфигурации каждого коммутатора создаём новые «виланы» с именами vlan2, vlan3, vlan4. Для этого заходим в консоль switch2 и пишем:

```
Enable
```

```
Configure terminal
```

```
vlan 2 (где 2 – это vlan ID)
```

```
name vlan2
```

```
exit
```

```
vlan 3
```

```
name vlan3
```

```
exit
```

```
vlan 4
```

```
name vlan4
```

```
exit
```

Тоже самое проделываем в конфигурации switch4. Теперь у нас есть новые «виланы», осталось только «рассказать» портам коммутаторов в какие «виланы» они входят. Стоит заметить, что порты, которые соединяют 2 коммутатора будут являться TRUNK портами, т.е. будут пропускать трафик от всех VLAN'ов. Остальные же порты, настраиваем как ACCESS, которые предназначены для передачи трафика одного VLAN'а.

У switch2 порты 7/1 настраиваем как TRUNK:

```
interface fastEthernet 7/1
```

```
switchport mode trunk
```

Портам 0/1 и 3/1 присваиваем vlan2:

```
interface fastEthernet 0/1
```

```
switchport access vlan 2
```

```
exit
```

```
interface fastEthernet 3/1
```

```
switchport access vlan 2
```

Портам 1/1 и 4/1 присваиваем vlan3:

```
interface fastEthernet 1/1
```

```
switchport access vlan 3
```

```
exit
```

```
interface fastEthernet 4/1
```

```
switchport access vlan 3
```

Портам 2/1 и 5/1 присваиваем vlan4:

```
interface fastEthernet 2/1
```

```
switchport access vlan 4
```

```
exit
```

```
interface fastEthernet 5/1
```

```
switchport access vlan 5
```

Тоже самое для switch4.

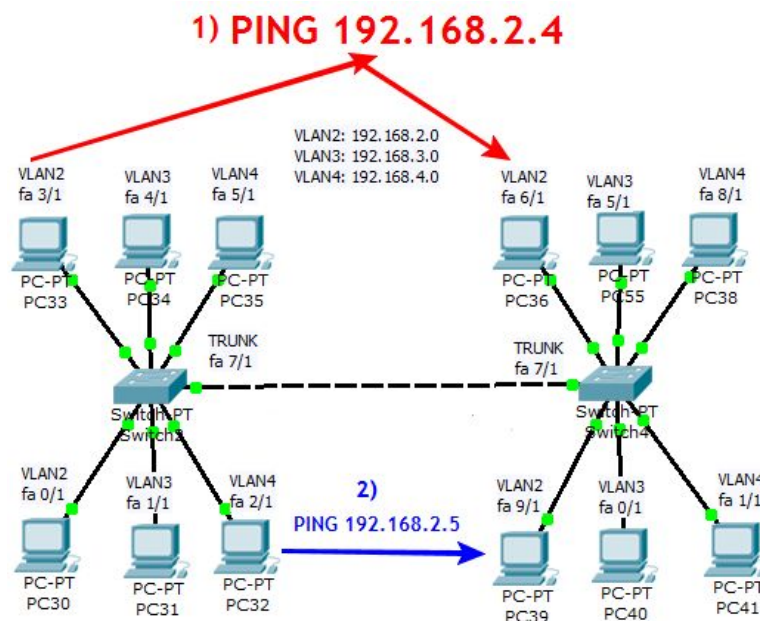
С помощью команды **show vlan** проверяем что получилось на switch2:

VLAN	Name	Status	Ports
1	default	active	Fa6/1, Fa8/1, Fa9/1
2	vlan2	active	Fa0/1, Fa3/1
3	vlan3	active	Fa1/1, Fa4/1
4	vlan4	active	Fa2/1, Fa5/1

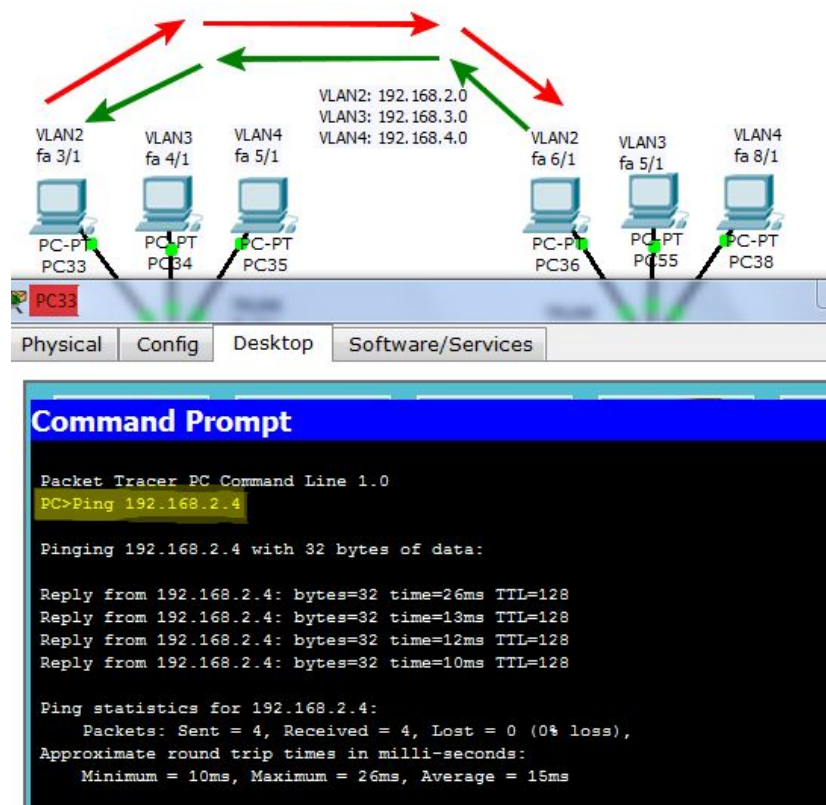
И на switch4:

VLAN	Name	Status	Ports
1	default	active	Fa2/1, Fa3/1, Fa4/1
2	vlan2	active	Fa6/1, Fa9/1
3	vlan3	active	Fa0/1, Fa5/1
4	vlan4	active	Fa1/1, Fa8/1

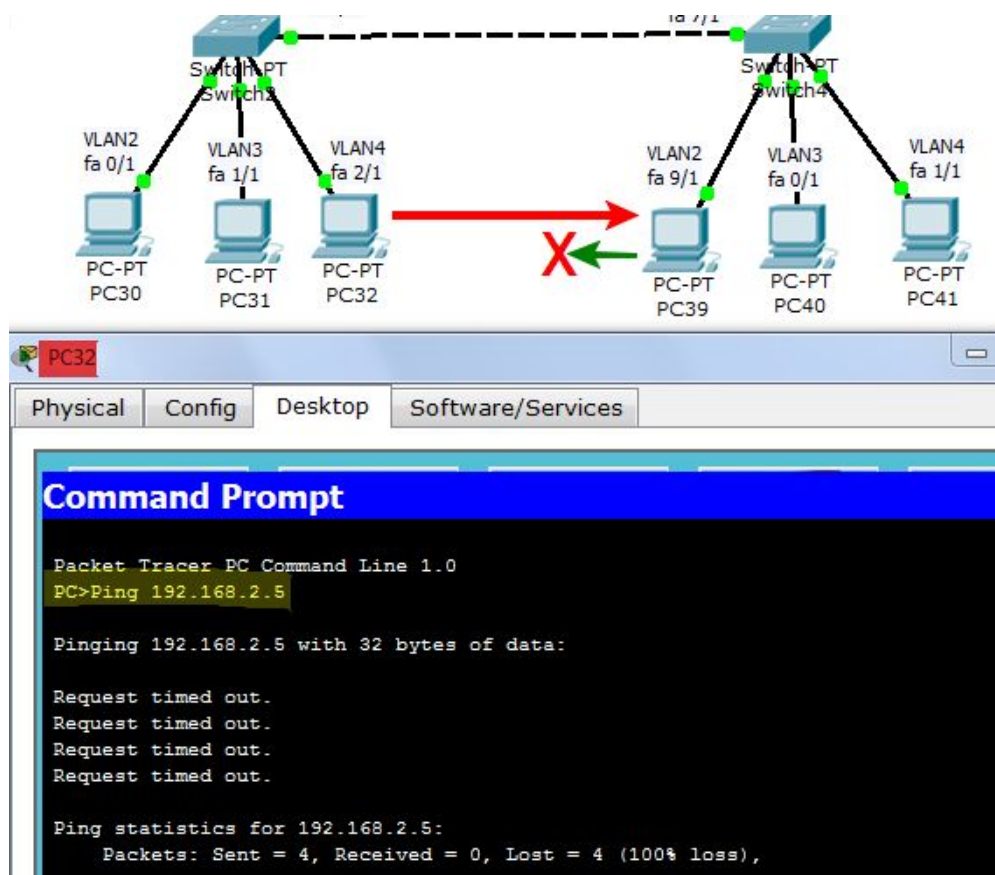
Теперь попробуем пинговать компьютеры из одного VLANa, а затем из разных.



1) В командной строке PC33 набираем ping 192.168.2.4. Всё в порядке.



2) В командной строке PC32 набираем ping 192.168.2.5. Time out. Но опять всё в порядке, ведь PC32 и PC39 находятся в разных виртуальных сетях.



3.5 Запароливание маршрутизаторов

Пароль на вход в привилегированный режим (privileged mode). Напомним, что вход в него осуществляется командой Enable.

Для этого:

Enable

Configure terminal

Enable password [пароль]

Пароль на вход по терминальным соединениям (гипертерминал и консольный кабель):

Enable

Configure terminal

Line console 0

Password [пароль]

Login [логин] - можно оставить пустым.

Пароль на виртуальные соединения - Telnet, SSH.

Enable

Configure terminal

Line vty 0 4

Password [пароль]

Login [логин] - можно оставить пустым.

Чтобы зашифровать пароль в конфигурационном файле:

Enable

Configure terminal

service password-encryption