

**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: \_\_\_\_\_ тема \_\_\_\_\_

Студенты \_\_\_\_\_ Копылов А.А., Ванцян Р.Н., группа 3103 \_\_\_\_\_

Руководитель практики \_\_\_\_\_ Соснин В.В. \_\_\_\_\_

Оценка руководителя \_\_\_\_\_

Дата \_\_\_\_\_

Санкт-Петербург  
2013г.

# 1 TeX(LaTeX)

TeX - система компьютерной вёрстки текста, разработанная американским профессором информатики Дональдом Кнудом 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  $\text{\TeX}$  для платформы Windows. Одним из существенных достоинств MiKTeX является возможность автоматического обновления установленных компонентов и пакетов.

В состав MiKTeX включены:

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

## 2 Git

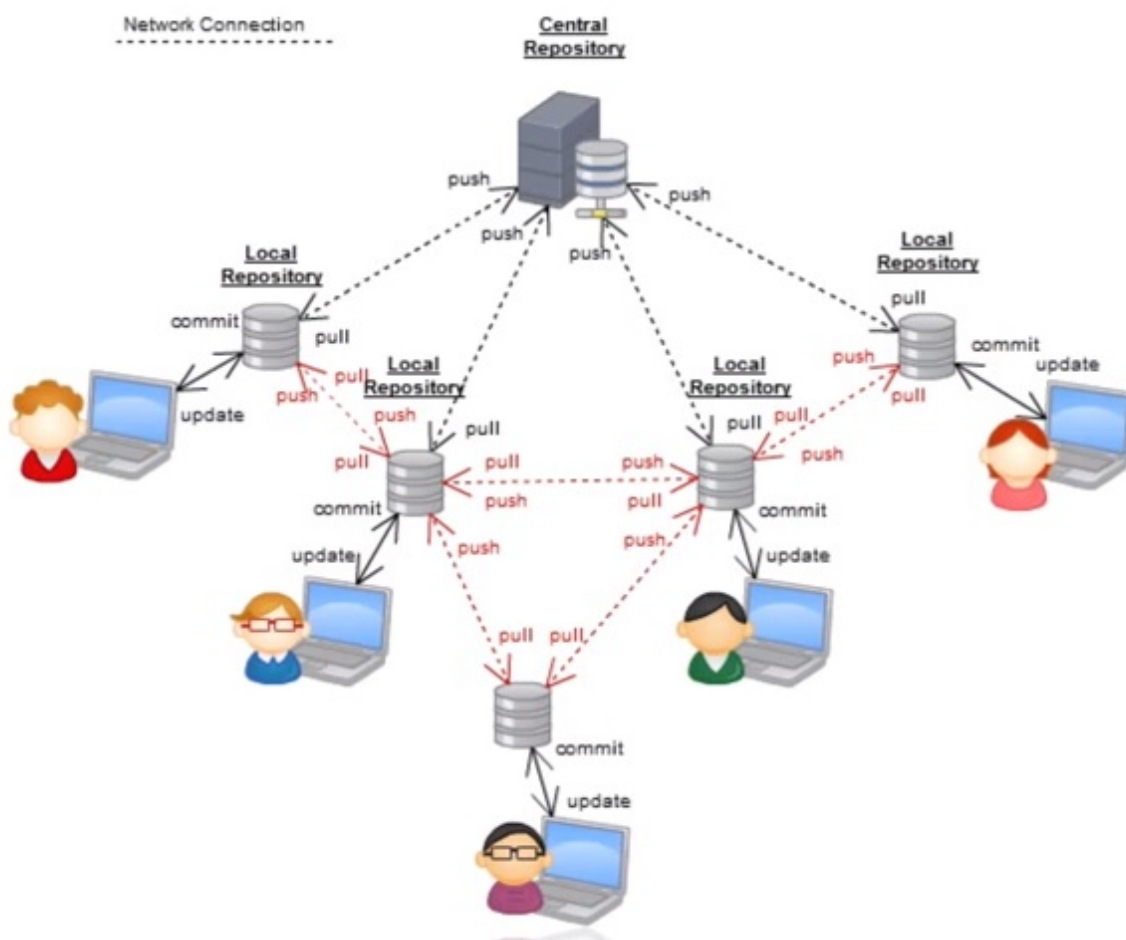
Git - система контроля версий (т.е. такая система, которая хранит все изменения в файле). В любой момент можно откатиться до любой заданной заранее позиции.

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.

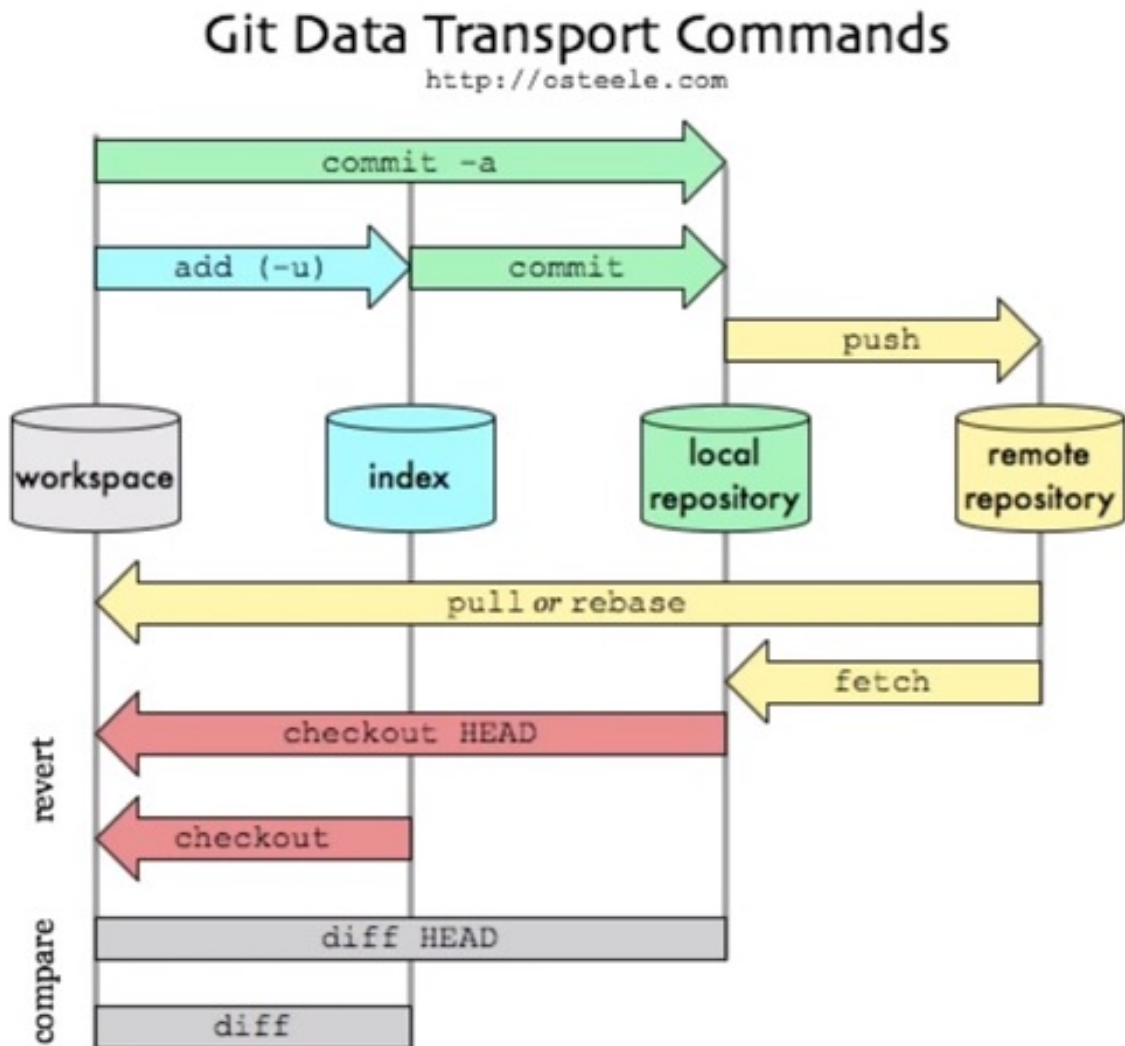
В Git между главным репозитием и пользователем существует промежуточный репозиторий. Именно в этом заключается отличие Git от других систем. Локальный репозиторий обеспечивает работу без интернета.



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.

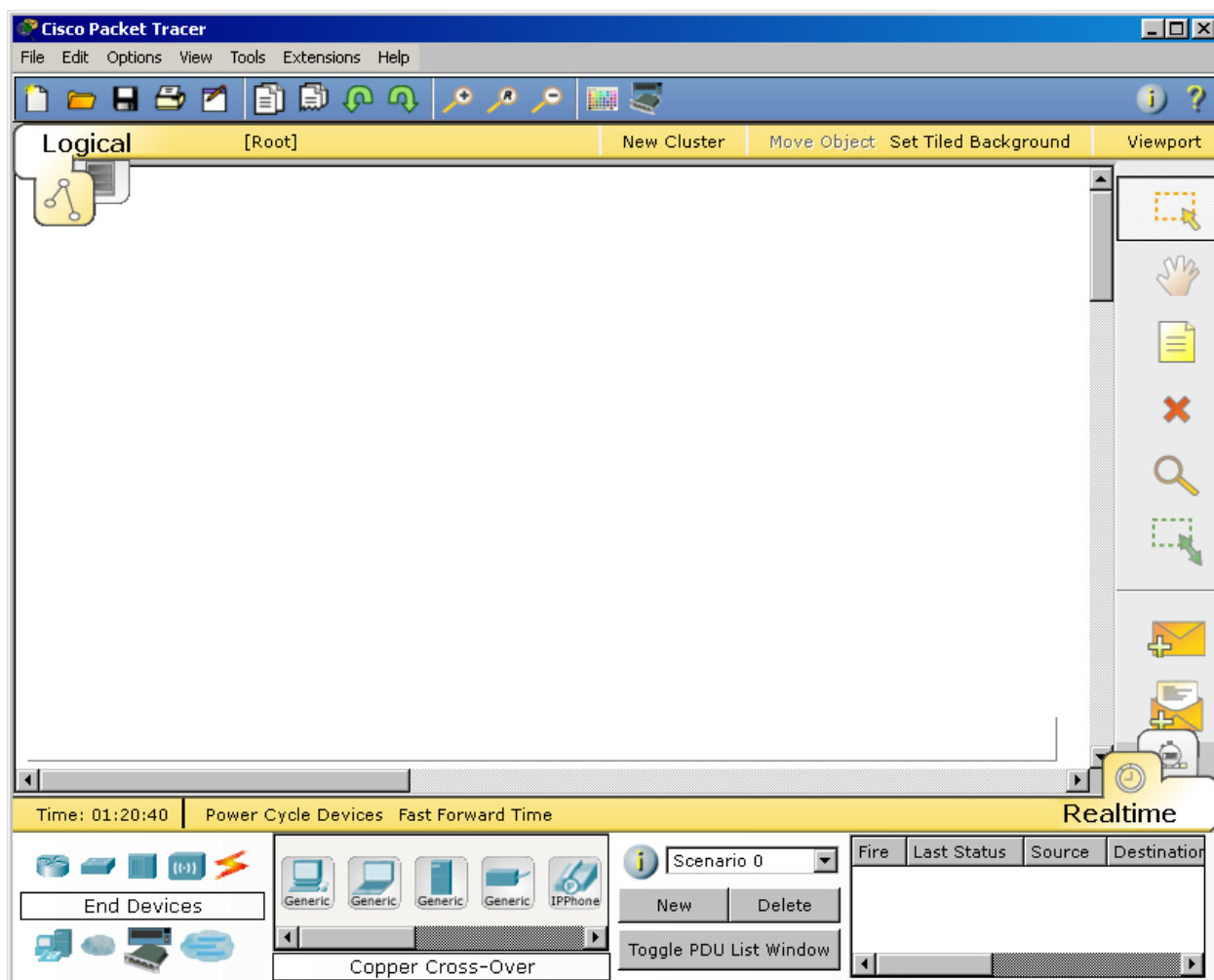
**git pull** - забираем изменения из удаленного репозитория.

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.

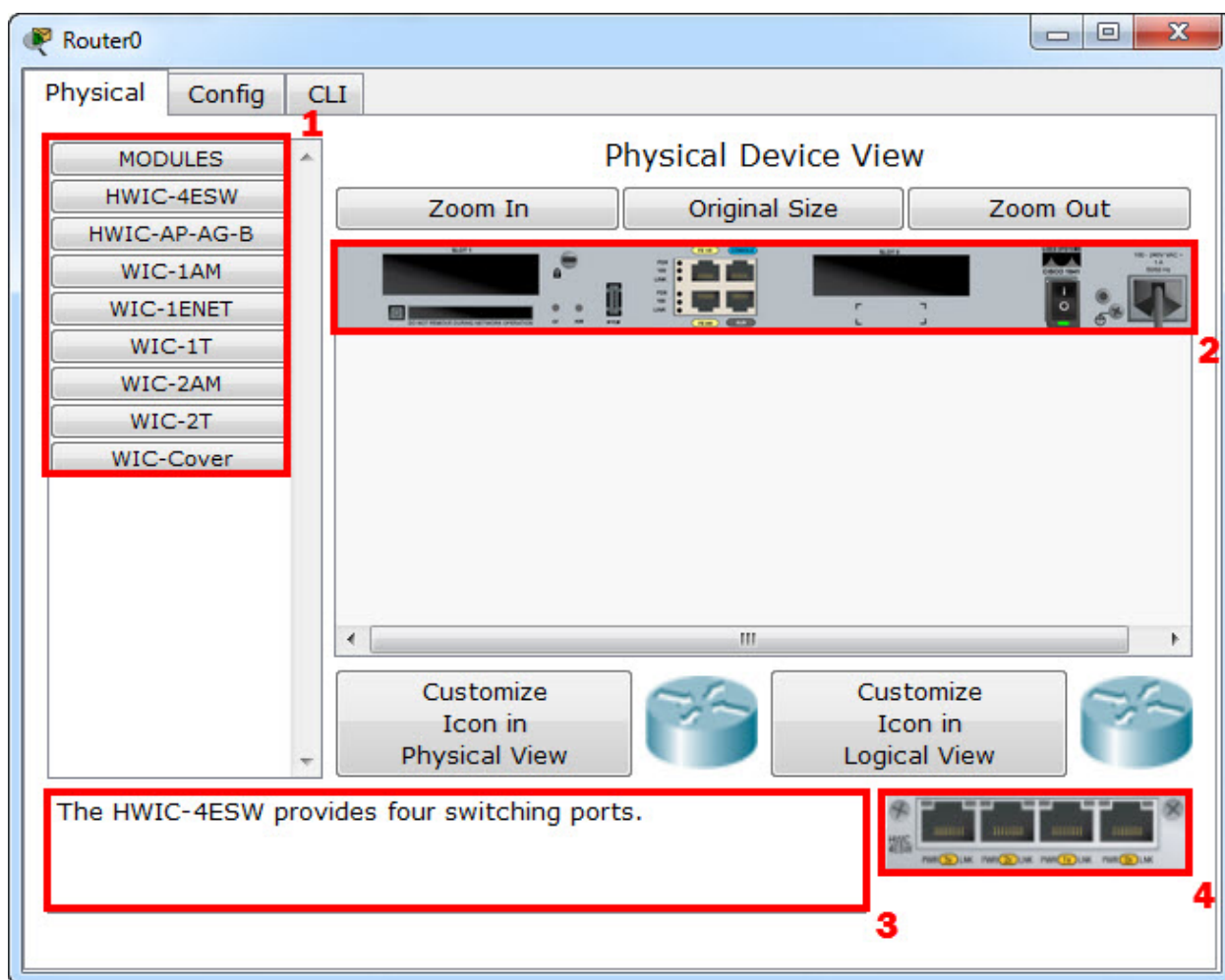
- Медь кроссовер - соединение медным кабелем типа витая пара, концы кабеля обжаты как кроссовер. Подойдет для соединения двух компьютеров, компьютер-роутер, роутер-роутер.
- Оптика - соединение при помощи оптического кабеля, необходимо для соединения устройств имеющих оптические интерфейсы.
- Телефонный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

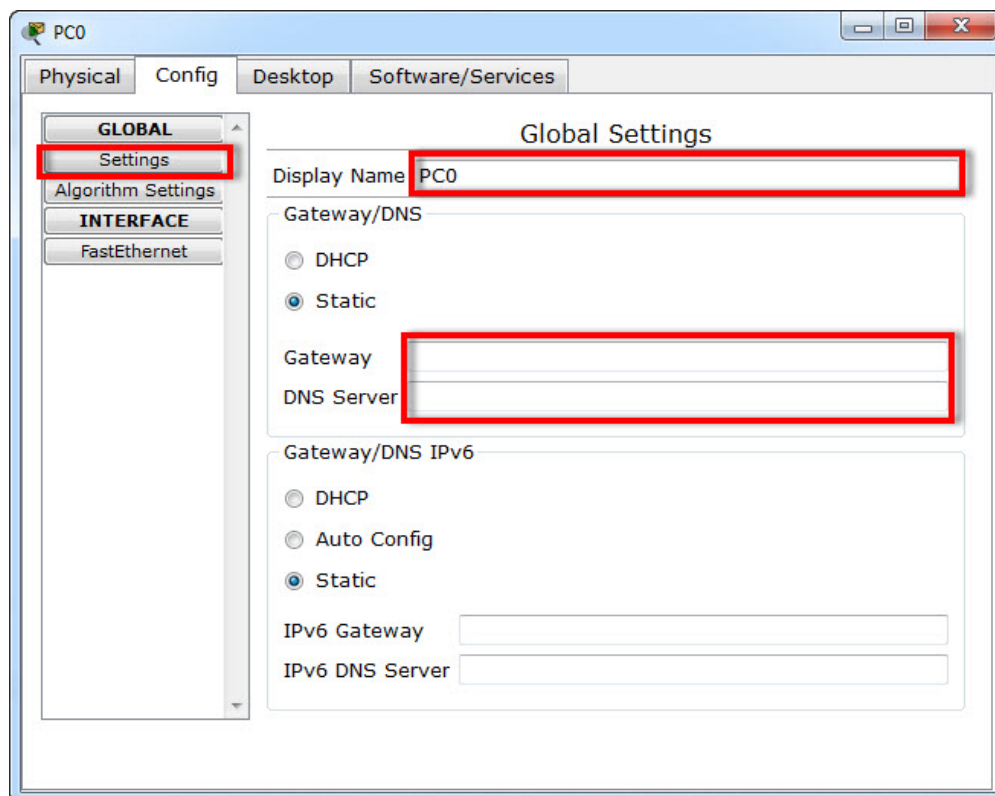
Добавление коммутатора в проект практически идентично добавлению маршрутизатора. Отличия в начальной стадии при выборе панели и в самом окне параметров добавленного устройства.

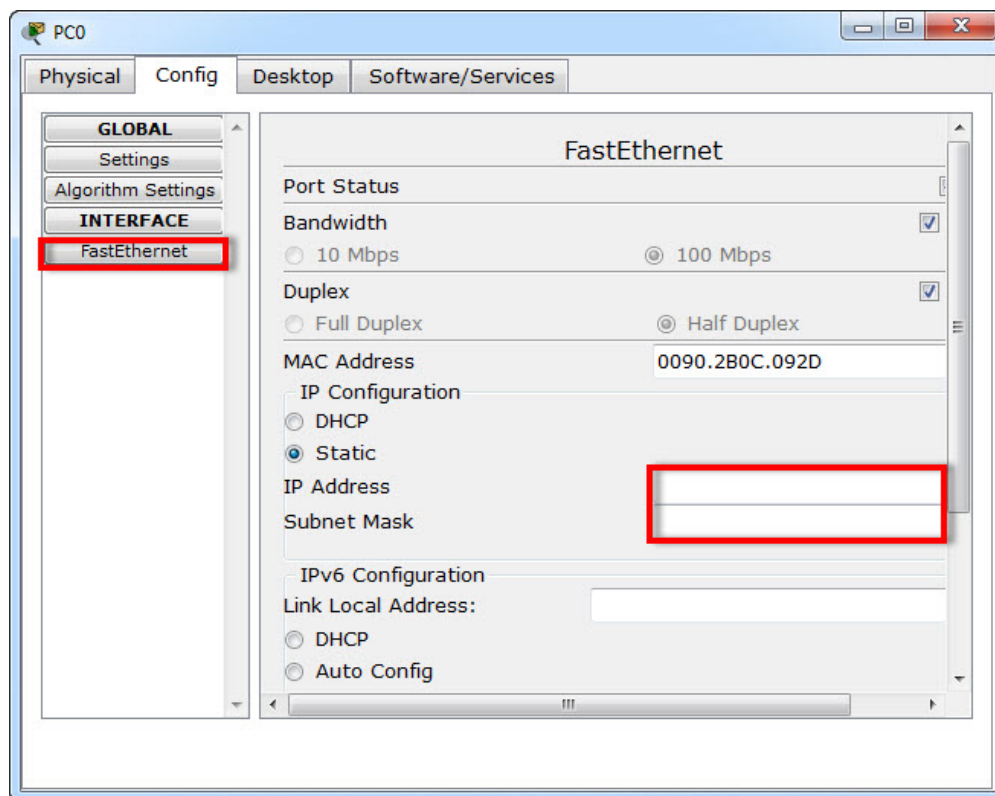
### 3.1.3 Добавление конечных узлов сети

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 выбранного сценария лабораторной работы устройства добавлены в проект, необходимо все единицы оборудования соединить между собой согласно сценарию. 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. В конечном итоге получается следующий проект сети.

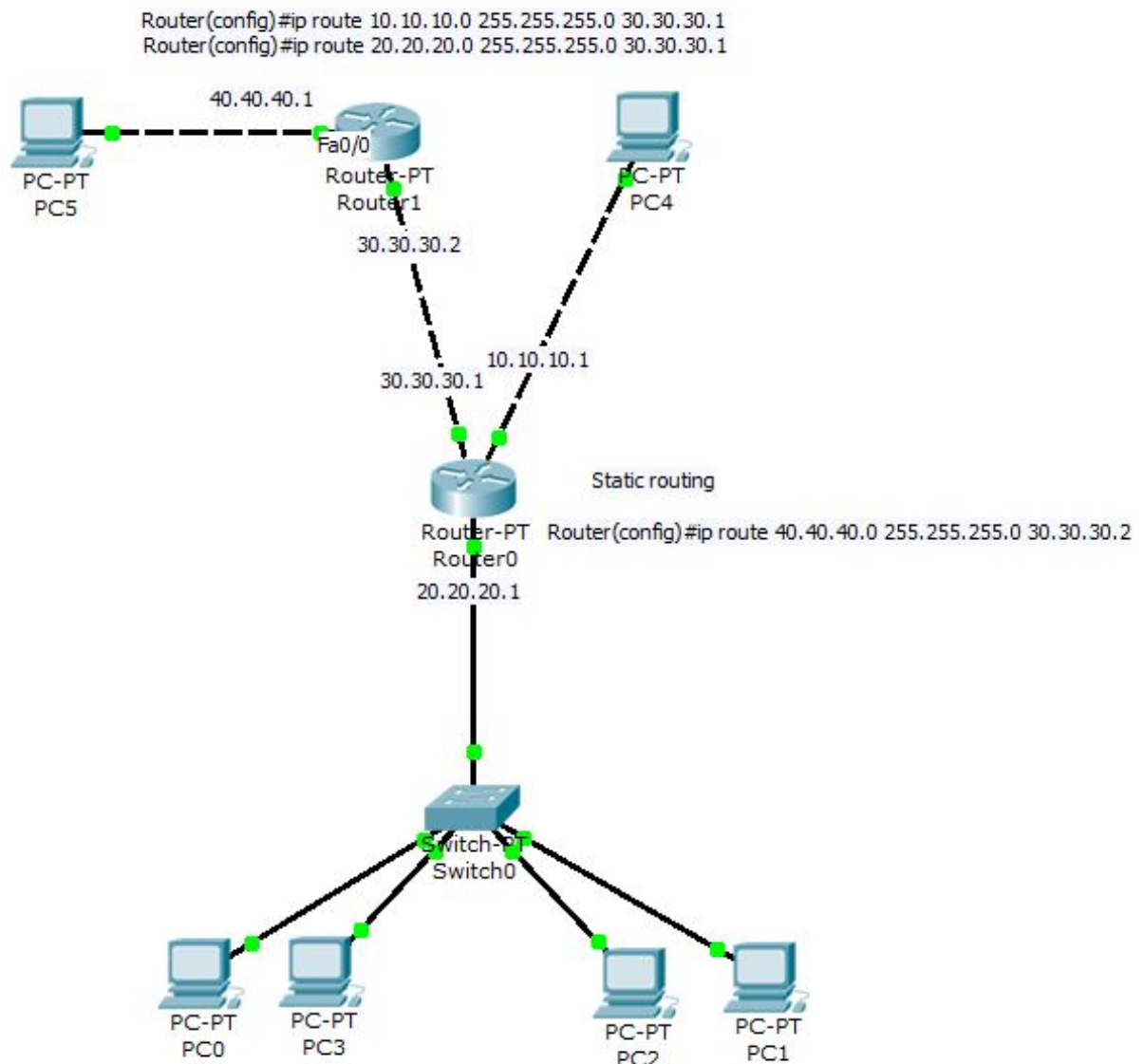
## 3.2 Пример простой компьютерной сети

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

