Шамшин Ю.В.

Методические указания к лабораторной работе

"Статическая маршрутизация в IP-сетях"

1. Цель работы.

- Изучение маршрутизации по умолчанию и статической маршрутизации в IP-сетях.
- Получение навыков управления таблицами маршрутизации на узлах сетевого уровня (хостах и маршрутизаторах).

2. Задание на выполнение работы.

1. **Определите свой вариант задания**.

   Для определения варианта задания:
   - запишите ваше имя в латинской транскрипции,
   - определите порядковый номер в алфавите для первой буквы в имени,
   - разделить это число по модулю на 10 и добавьте 1.

   Например, для Li Yurij первая буква в имени Y имеет номер 25; вычисляем (25 MOD 10)+1 = 5+1 = 6. Получили вариант 6.

2. **Запустите программу javaNetSim** и откройте свой вариант файла **varN.jnst** с исходной конфигурацией сети. Программа эмуляции работы компьютерных сетей (javaNetSim 0.41) написана на Java и может быть запущена на Windows, Mac, Linux. Загрузить эту программу вместе с вариантами заданий нужно с сайта курса.

3. **Для всех узлов** сети установите корректные IP-адреса и маски подсетей так, чтобы добиться успешного прохождения эхо-запросов (ping) в локальных сегментах сети - между узлами находящимися в одной подсети. (1 БАЛЛ)

4. **Настройте шлюзы по-умолчанию** на хостах PC1-PC4 и на конечных (тупиковых) маршрутизаторах сети. (1 БАЛЛ)

5. **Настройте таблицы маршрутизации (ТМ)** на промежуточных маршрутизаторах так, чтобы добиться доставки пакетов между всеми хостами PC1-PC4 туда и обратно. (4 БАЛЛА)

6. **Перенастройте ТМ** на маршрутизаторах R1-R8 так, чтобы обеспечить **кратчайшую доставку** пакетов между всеми узлами PC1-PC4, если кратчайшую доставку ещё не удалось обеспечить в пункте 3. (1 БАЛЛ)

7. **Сформируйте отчёт**. (3 БАЛЛА)
   Отчёт сдаётся в электронном виде. К отчёту приложить итоговый файл сети в формате javaNetSimulator (.jnst).
   В отчёте привести:
   - вычисление варианта задания;
   - эскиз схемы сети;
   - конфигурацию IP для каждого из узлов;
   - таблицы маршрутизации для маршрутизаторов R1-R8;
   - результаты эхо-запросов между узлами PC1-PC4;
   - обоснование правильности и оптимальности выбранных маршрутов.

3. Краткие теоретические сведения.

Маршрутизация – важнейший процесс в IP-сетях. Чтобы некоторый узел в сети смог передать данные другому, должен быть определен механизм передачи пакет от одного узла к другому узлу даже если они находятся не в одном сегменте сети. Такой механизм и называется маршрутизацией.

Маршрутизация выполняется на конечных и промежуточных узлах специальными программными или аппаратными средствами. Маршрутизация, осуществляемая IP – это процесс поиска в таблице маршрутизации интерфейса, куда будет послан пакет.

Маршрутизатор – сетевое устройство, используемое в компьютерных сетях передачи данных, которое, на основе информации о топологии сети, которая заносится и хранится в таблицах маршрутизации, и определенных правил, принимает решения о пересылке сетевых пакетов их получателю.

Стандартный механизм маршрутизации пакетов в Internet — per hop behavior - каждый узел в сети принимает решение куда ему отправить пакет на основе информации, полученной от протоколов динамической и статической маршрутизации.

Существует три типа маршрутизации:

- **Маршрутизация по умолчанию** осуществляется на основе конфигурации IP на узле и с использованием указания шлюза по умолчанию.
- **Статическая маршрутизация** осуществляется на основе таблиц маршрутизации обычно задаваемых вручную администратором.
- **Динамическая маршрутизация** осуществляется на основе таблиц маршрутизации формируемых автоматически с помощью протоколов маршрутизации.

Протокол маршрутизации это сетевой протокол, используемый маршрутизаторами для определения возможных маршрутов следования данных в составной компьютерной сети. Применение протокола маршрутизации позволяет избежать ручного ввода маршрутов, что, в свою очередь, снижает количество ошибок, обеспечивает согласованность действий всех маршрутизаторов в сети и облегчает труд администраторов.

3.1. Маршрутизация по умолчанию.

Для объединения подсетей в единую сеть в простейшем случае на узле используется маршрутизация по умолчанию посредством шлюзов. Шлюзом будем называть узел внутри подсети, который предоставляет доступ в другую подсеть.

В качестве узла могут выступать как хосты, так и маршрутизаторы. Хост не умеет перекладывать пакеты с одного своего IP интерфейса на другой (осуществлять ip-forwarding), а маршрутизатор может это делать.

Схема работы маршрутизации по умолчанию выглядит следующим образом:

1. Предварительно на узле должны быть сконфигурированы сетевые интерфейсы и задан адрес шлюза по умолчанию.
2. Узел сравнивает сеть IP-адреса назначения с номерами сетей на своих интерфейсах (посредством наложения масок) и в случае совпадения, направляет пакет непосредственно узлу назначения через этот сетевой интерфейс, предварительно определив его физический адрес (например, через ARP).

3. Если совпавших подсетей нет, то пакет отправляется узлу, указанному в качестве шлюза по умолчанию.
4. Если шлюза по умолчанию нет, то пакет отбрасывается (теряется), отправляется ICMP тип=3 код= 6 или 7 (destination network unknown или destination host unknown).

3.2. Статическая маршрутизация.

Чаще всего в качестве шлюза выступает маршрутизатор. Как правило, маршрутизатор имеет несколько интерфейсов, может одновременно находится в нескольких подсетях, сконфигурирован для перекладывания пакетов с одного своего интерфейса на другой и может поддерживать или не поддерживать протоколы динамической маршрутизации.

Информация, на основе которой маршрутизатор принимает решение о дальнейшей пересылке пакетов содержится в таблице маршрутизации.

Схема работы статической маршрутизации выглядит следующим образом:

1. Предварительно на маршрутизаторе должны быть сконфигурированы сетевые интерфейсы и задан адрес шлюза по умолчанию.
2. Маршрутизатор определяет, является ли получатель пакета локальным узлом (т.е. находится ли он в той же подсети, что маршрутизатор). Если получатель находится в одной из подсетей маршрутизатора, то пакет направляется ему напрямую.
3. Если узел-получатель находится в другой подсети, то в таблице маршрутизации ищется путь к этому узлу. При просмотре, сеть узла-получателя сравнивается с записями о сетях в таблице маршрутизации и при обнаружении совпадения пакет направляется маршрутизатору, указанному в соответствующей записи.
4. Если маршрут в таблице маршрутизации не найден, то пакет отправляется шлюзу по умолчанию, указанному на маршрутизаторе.
5. Если маршрут не найден, а запись о шлюзе по умолчанию отсутствует, то пакет отбрасывается (теряется).

3.3. Таблица маршрутизации.

Чтобы по адресу назначения пакета можно было бы выбрать маршрут дальнейшей пересылки, каждый узел, поддерживающий сетевой уровень анализирует специальную информационную структуру, называемую таблицей маршрутизации.

Простейшая таблица маршрутизации включает в себя информацию об узле (или сети) назначения, маске подсети для узла (сети) назначения, интерфейсе, через который следует направить пакет и шлюзе – удаленном узле, которому будет передан пакет. Для указания шлюза по умолчанию, в таблице существует специальная запись default (0.0.0.0/0).

Пусть задана конфигурация сети показанная на рисунке ниже.

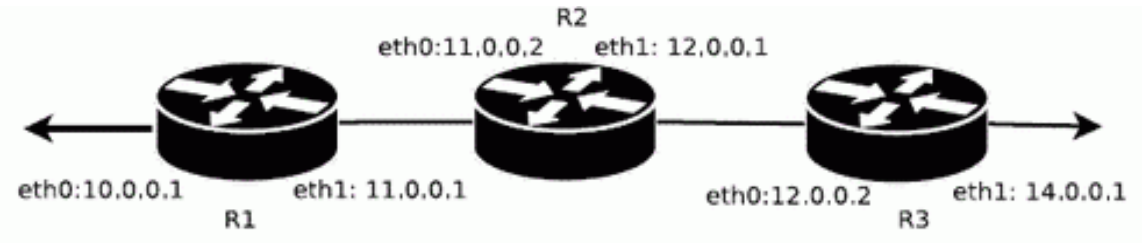


Таблица маршрутизации для R2 может выглядеть например так:

Destination	Gateway	Genmask	Metric	Iface
default	11.0.0.2	0.0.0.0	1	eth0
127.0.0.0	127.0.0.1	255.0.0.0	0	lo
11.0.0.0	11.0.0.2	255.0.0.0	0	eth0
12.0.0.0	12.0.0.1	255.0.0.0	0	eth1
10.0.0.0	11.0.0.1	255.0.0.0	1	eth0
14.0.0.0	12.0.0.2	255.0.0.0	1	eth1

В первой колонке таблице перечисляются номера подсетей, во второй – какому маршрутизатору следует перенаправить пакет для отправки в заданную подсеть. Третья колонка задает маску подсети назначения, в четвертой указывается через какой интерфейс следует направить пакет, в пятой указана метрика.

3.4. Протокол ICMP.

Для проверки корректности соединений и функционирования сети обычно используется протокол ICMP – Internet Control Message Protocol. ICMP – протокол сетевого уровня и работает поверх протокола IP. Он предназначен для обмена информацией об ошибках между маршрутизаторами (шлюзами) сети и узлом-источником пакета.

С помощью специальных пакетов этот протокол сообщает о невозможности доставки пакета, превышении времени жизни, аномальных значениях параметров, изменении маршрута пересылки и т. п.

В простейшем случае, для проверки работоспособности сети используются два сообщения ICMP: эхо-запрос (Echo request) и эхо-ответ (Echo reply). Когда на узел приходит сообщение ICMP типа эхо-запрос, он отправляет сообщение эхо-ответ на тот узел, с которого пришёл запрос. Пример реализации такого обмена представлен в утилите ping, входящей в состав почти любой сетевой ОС.

3.5. Работа в программе javaNetSim.

3.5.1. Графический интерфейс имитатора javaNetSim.

Основное окно программы представляет собой инструмент взаимодействия пользователя с имитатором. С помощью этого инструмента пользователь может добавлять, удалять и соединять между собой сетевые устройства, а также работать с сетью на любом из четырех уровней стека протоколов TCP/IP.

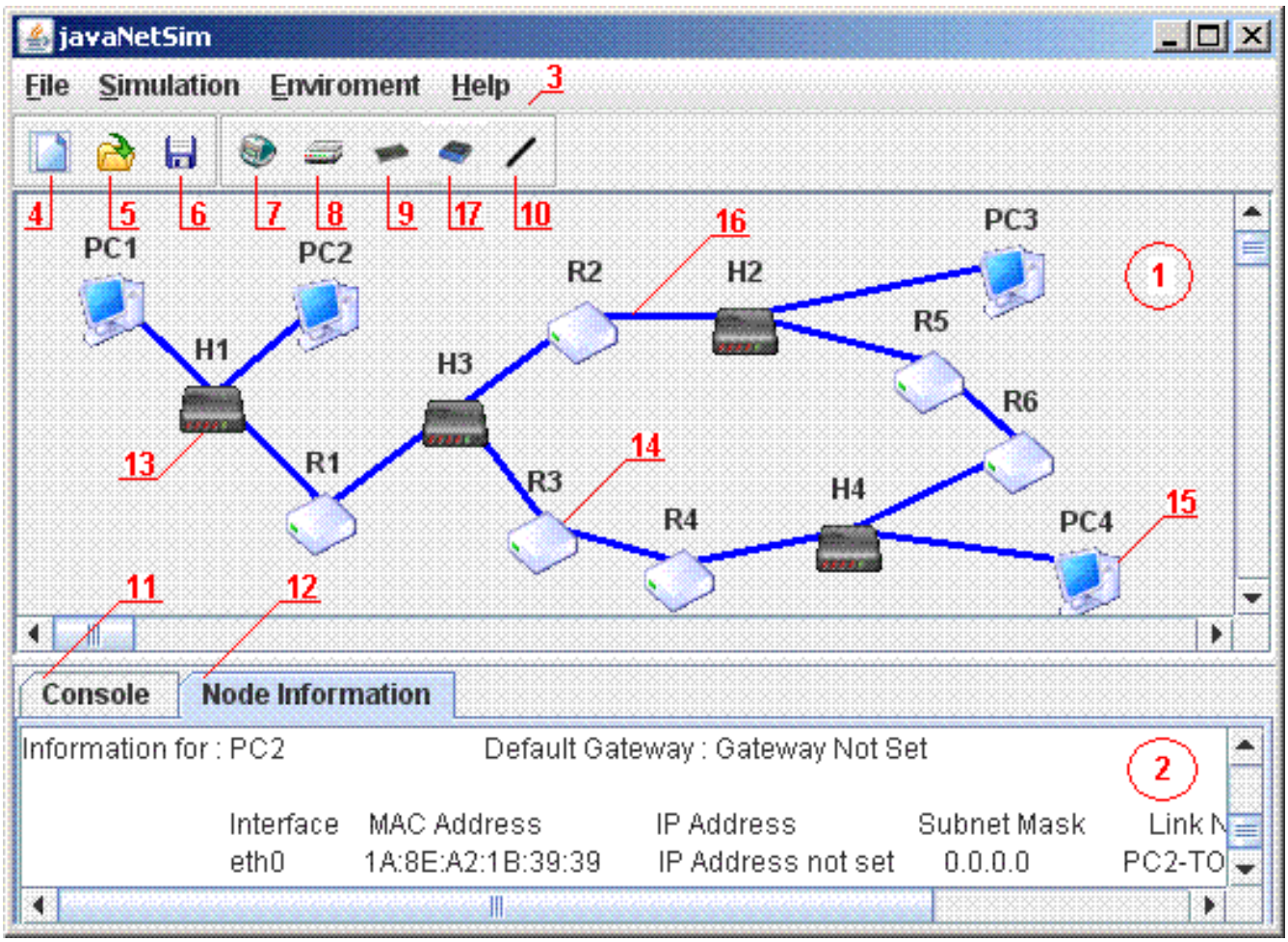


Рис. Основное окно программы javaNetSim.

Основное окно программы логически разделено на четыре части:

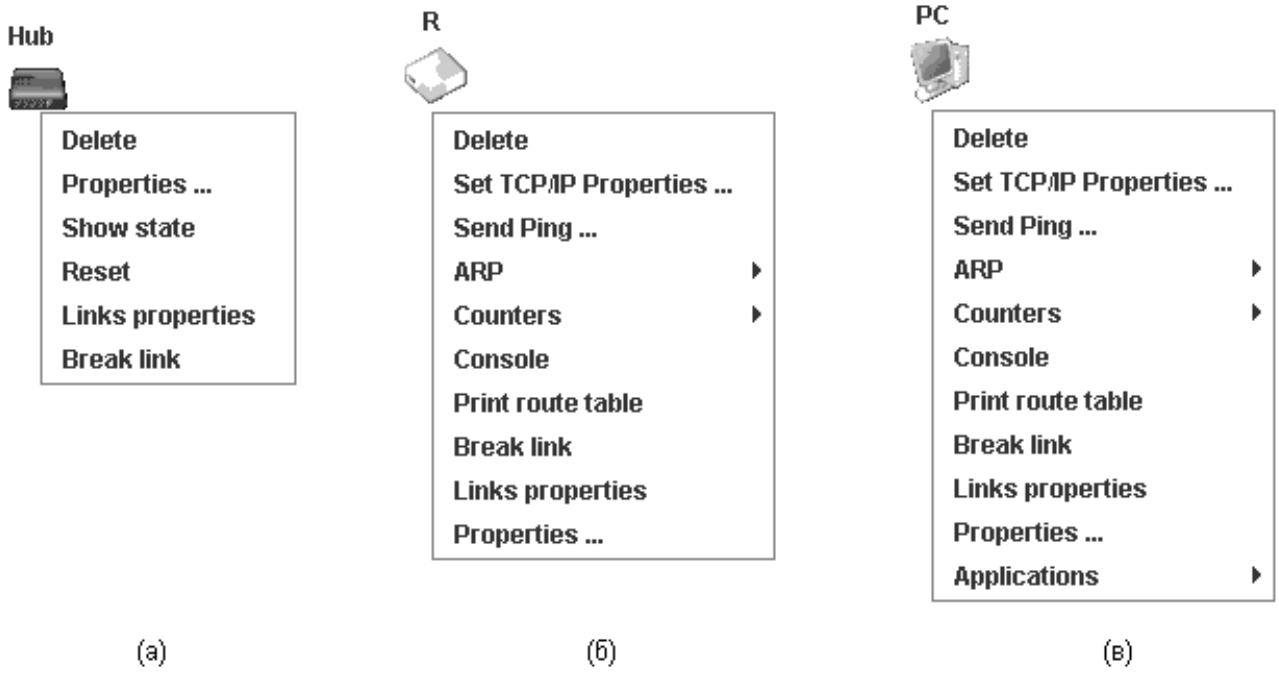
1. рабочая область, обозначенная на рисунке цифрой (1) – содержит сетевые устройства и линии связи между ними:
   - Концентратор на пять сетевых интерфейсов (13).
   - Маршрутизатор соединяющий две подсети (14).
   - Компьютер или конечный узел сети (15).
   - Линия связи между двумя сетевыми устройствами (16).
2. область вывода результатов (2) – содержит две вкладки:
   - вкладка "консоль" (11) – содержит журнал передачи пакетов по сети
   - вкладка "информация об устройствах" (12) – для каждого интерфейса всех сетевых устройств содержит IP-адрес, маску подсети и шлюз по умолчанию.
3. главное меню (3) – содержит основные действия по управлению имитатором;
4. линейка инструментов – содержит следующие кнопки:

- кнопка "создать пустую конфигурацию" (4);
- кнопка "открыть существующую конфигурацию" (5);
- кнопка "сохранить текущую конфигурацию" (6);
- кнопка "создать компьютер" (7);
- кнопка "создать маршрутизатор" (8);
- кнопка "создать концентратор" (9);
- кнопка "создать коммутатор" (17);
- кнопка "создать соединение" (10).

3.5.2. Контекстное меню javaNetSim.

Контекстное меню, вызываемое щелчком правой кнопкой мыши, отличается для устройств работающих на разных уровнях стека протоколов TCP/IP.

На рис. изображены контекстные меню для устройств: физического уровня - концентратор (а), сетевого уровня - коммутатор (б) и прикладного уровня - компьютер (в).



Контекстные меню устройств.

Link Properties (Свойства линий связи) вызывается диалог, который позволяет установить коэффициент пропускания для интерфейса, показывающий какой процент пакетов линия связи подключенная к этому интерфейсу будет пропускать.

Меню ARP позволяет управлять таблицей протокола ARP на выбранном устройстве содержит три подпункта:

- Add static entry to ARP table – вызывает два диалоговых окна: в первом вводится MAC адрес, а во втором IP адрес, после чего в ARP таблицу заноситься статическая запись о связи IP и MAC адресов;
- Remove entry from ARP table – вызывает диалоговое окно, позволяющее ввести IP адрес, для которого будет удалена запись из ARP таблицы;
- Print ARP table – выводит на вкладку "консоль" ARP таблицу выбранного сетевого устройства.

3.5.3. Описание консольного режима (командной строки).

Устройства в javaNetSim конфигурируется в командной строке подобной Cisco IOS. Подсоединение к узлу осуществляется через Telnet или на IP-адрес любого из интерфейсов или через последовательный порт компьютера, связанный с консольным портом (Console).

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 реальному маршрутизатору через Telnet обычно запрещён.

В рамках данного курса конфигурация маршрутизаторов будет осуществляться посредством Console вызываемую через контекстное меню маршрутизатора.

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.

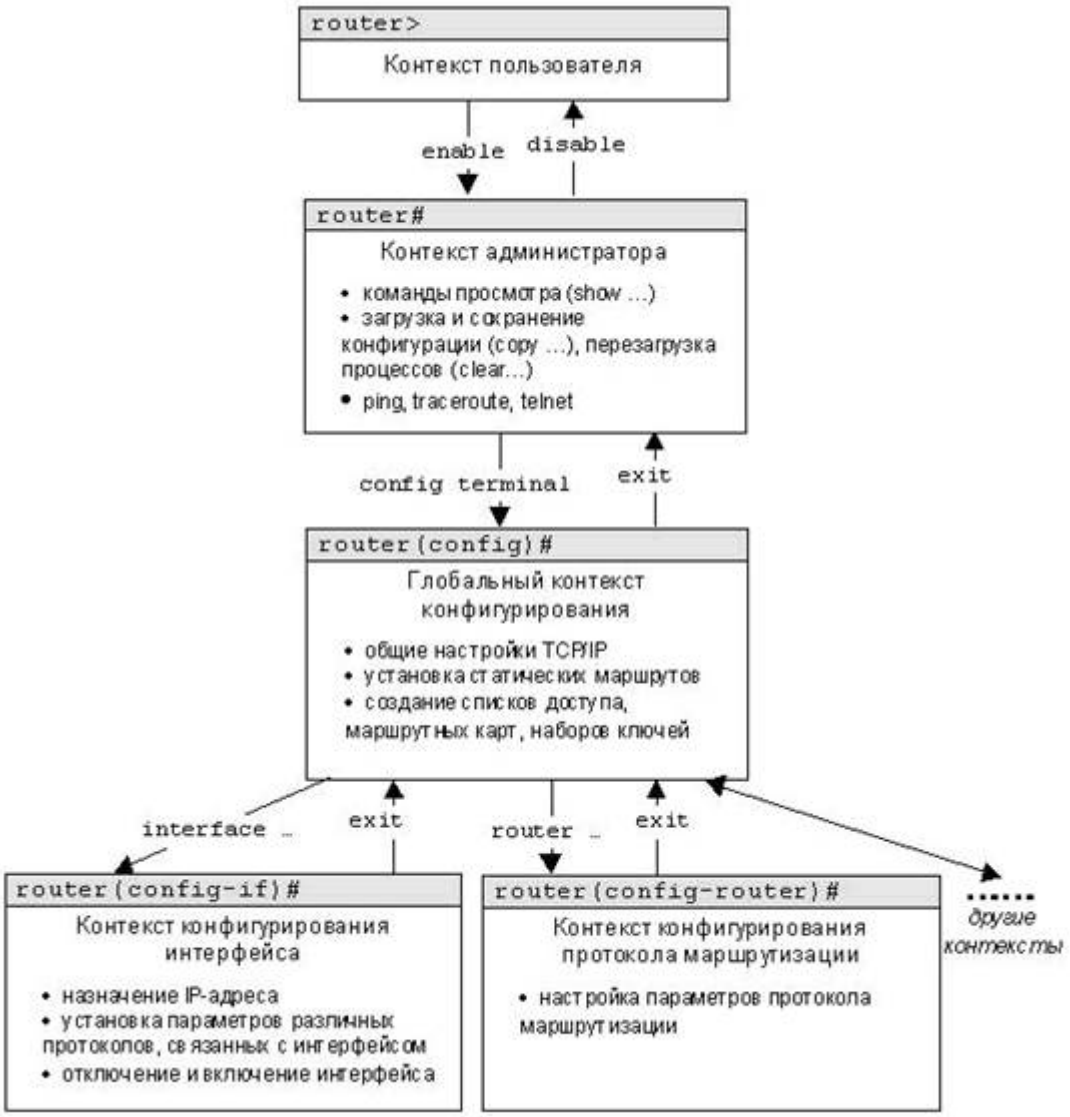
В любом месте командной строки для получения помощи может быть использован **вопросительный знак**:

> 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

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

3.6. Схема контекстов javaNetSim и Cisco IOS.

Существует множество контекстов конфигурирования. Некоторые контексты конфигурирования находятся внутри других контекстов конфигурирования. Наиболее часто используемые контексты показаны на рисунке ниже.



Упрощенная схема контекстов javaNetSim и Cisco IOS.

Вид приглашения командной строки в контекстах конфигурирования, которые будут встречаться наиболее часто:

router(config)# /глобальный/
router(config-if)# /интерфейса/
rounter(config-router)# /динамической маршрутизации/
rounter(config-line)# /терминальной линии

Выход из глобального контекста конфигурирования в контекст администратора, а также выход из любого подконтекста конфигурирования в контекст верхнего уровня производится командой **exit** или **Ctrl-Z**. Кроме того, команда **end**, поданная в любом из контекстов конфигурирования немедленно завершает процесс конфигурирования и возвращает оператора в контекст администратора.

3.6.1. Контекст пользователя.

Контекст 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 (некоторые базовые операции для мониторинга), не влияющие на конфигурацию маршрутизатора. Вид приглашения командной строки:

router>

Вместо слова router выводится имя маршрутизатора, если оно установлено.

3.6.2. 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**.

ВАЖНО. В эмуляторе javaNetSim сразу настроен контекст администратора.

3.6.3. 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, например:

3.6.4. Контекст конфигурирования интерфейса.

Открывается командой **interface имя_интерфейса** (например interface serial0), поданой в глобальном контексте конфигурирования;

3.6.5. Контекст конфигурирования процесса динамической маршрутизации.

Открывается командой **router протокол номер_процесса** (например, router ospf 1, поданной в глобальном контексте конфигурирования.

ВАЖНО! Любая команда конфигурации вступает в действие немедленно после ввода, а не после возврата в контекст администратора.

3.7. Работа в javaNetSim с протоколами уровня приложений.

В имитаторе javaNetSim имеется возможность работы со следующими протоколами уровня приложений стека протоколов TCP/IP:

- Echo (UDP и TCP реализации),
- SNMP,
- TELNET.

3.7.1 Работа с протоколом Echo в javaNetSim.

Имитатор javaNetSim позволяет использовать протоколы UDP или TCP в качестве транспортных протоколов для протокола Echo.

Для установки echo-сервера в режим прослушивания порта надо выбрать пункт:

- "Applications" -> "Start udp echo server to listen" - для Echo-UDP
- "Applications" -> "Start tcp echo server to listen" - для Echo-TCP.

После этого в появившемся диалоговом окне следует ввести номер порта, на котором выбранное приложение будет ожидать сообщения. Теперь с любого другого узла можно отсылать сообщения на этот узел и получать ответы.

Для того, чтобы послать эхо-запрос, необходимо в контекстном меню выбрать

- "Applications" -> "Send data via udp echo client" - для Echo-UDP
- "Applications" -> "Send data via tcp echo client" - для Echo-TCP

и ввести четыре параметра:

1. IP-адрес компьютера, на котором запущен echo-сервер;
2. номер порта на котором echo-сервер ожидает сообщения;
3. сообщение – любой текст;
4. количество посылаемых сообщений.

Протокол Echo обладает простой структурой, поэтому при помощи telnet-клиента можно подключиться к Echo-TCP-серверу. В таком режиме нажатие любой клавиши на клавиатуре будет сопровождаться выводом ее на экран терминала.

3.7.2. Работа с протоколом SNMP в javaNetSim.

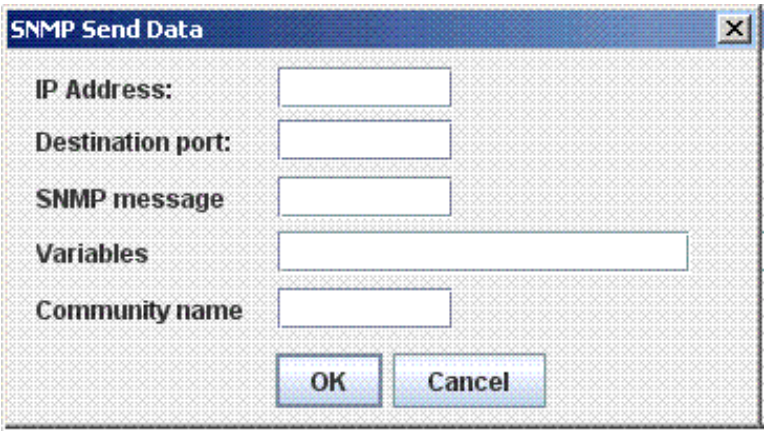
Для запуска SNMP агента: Application -> Start SNMP Agent и задать два параметра:

- порт, на котором SNMP агент будет ожидать пакеты;
- имя группы доступа для SNMP агента.

Чтобы послать запрос SNMP агенту: Application -> Send SNMP message и заполнить поля.

1. IP Address – IP адрес компьютера на котором установлен SNMP агент.
2. Destination Port – порт на котором SNMP агент ожидает пакеты.
3. SNMP message – SNMP запрос, может принимать значения: get, getnext, set.
4. Variables – SNMP переменные описываемые деревом MIB.

5. Community name – имя группы доступа, которое должно совпадать с именем группы доступа установленным при создании агента.



Создание SNMP запроса.

Поле Variables имеет специальный формат, различный для запросов get(getnext) и set. Если SNMP запрос является get или getnext запросом, то строка переменных должна выглядеть следующим образом:

<переменная>[;<переменная>]

Например: ip.address_eth0;device.hostname.

А если SNMP запрос является set запросом, то в строке переменных к каждой переменной добавляется значение:

<переменная>="<значение>" [;<переменная>="<значение>"]

Например: ip.address_eth0="192.168.10.3"

Результаты запроса будут выведены на вкладку "Console" javaNetSim. Например:

```
PC2 SNMP Protocol Data Application Received getResponse:
'IP.Address_Eth0=172.168.0.2' , 'Device.Hostname=PC1'
```

Список SNMP переменных, поддерживаемых имитатором javaNetSim.

Переменные, которые имеют режим доступа "только для чтения".

- Counter.InputIP – количество пришедших IP пакетов;
- Counter.OutputIP – количество отправленных IP пакетов;
- Counter.ARP – количество обработанных ARP пакетов;
- Counter.InputTCP – количество пришедших TCP пакетов;
- Counter.OutputTCP – количество отправленных TCP пакетов;
- Counter.ReceiveDuplicatedTCP – количество дублирующихся пакетов TCP полученных устройством;
- Counter.SendDuplicatedTCP – количество дублирующихся пакетов TCP отправленных устройством;
- Counter.SendAckTCP – количество посланных ACK пакетов;
- Counter.InputUDP – количество пришедших UDP пакетов;

- Counter.OutputUDP – количество отправленных UDP пакетов;
- Device.AllInterfaces – список всех возможных интерфейсов устройства;
- Device.AvailableInterfaces – список всех доступных интерфейсов устройства;
- Device.Hostname – имя устройства;
- Device.MACaddress_Eth0 – MAC адрес устройства на интерфейсе Ethernet0;
- IP.AllInterfaces – список всех возможных интерфейсов устройства работающих по протоколу IP;
- IP.ARPTable – ARP таблица для устройства;
- SNMP.revision – версия модификации SNMP;
- SNMP.version – версия SNMP.
- Некоторые SNMP переменные имеют режим доступа "чтение и запись".
- IP.DefaultGateway – шлюз по умолчанию;
- IP.Address_Eth0 – IP адрес интерфейса Ethernet0;
- IP.SubnetMask_Eth0 – маска интерфейса Ethernet0;
- SNMP.CommunityName – имя группы доступа для SNMP агента.

Режим доступа определяет действия, которые можно производить с переменной. Если переменная имеет режим доступа только чтение, то попытка записать новое значение завершиться с ошибкой.

3.7.3. Работа с протоколом TELNET в javaNetSim.

Для запуска TELNET сервера необходимо выбрать пункт контекстного меню "Application" -> "Start telnet server to listen" и задать два параметра:

- порт, на котором TELNET-сервер будет ожидать пакеты;
- пароль для доступа к TELNET-серверу.

Для соединения с TELNET сервером необходимо выбрать пункт контекстного меню "Application" -> "Telnet client" и задать два параметра:

- IP адрес TELNET-сервера;
- порт, на котором TELNET-сервер ожидает пакеты.

После этого откроется окно терминала и если соединение прошло успешно появится приглашение ввести имя пользователя: login. После введения имени появится приглашение ввести пароль: password. После введения пароля, имя пользователя и пароль проверяются и, если они корректны, будет выведено приглашение в виде:

<имя компьютера> #

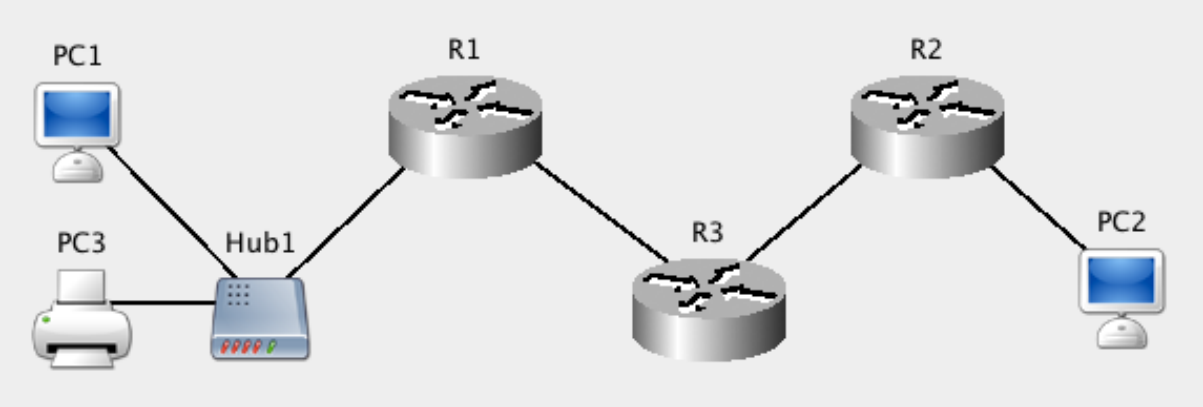
В javaNetSim для TELNET-сервера используется имя пользователя root и пароль, установленный при создании TELNET-сервера. В telnet доступны следующие команды:

- route – просмотр и редактирование сетевых маршрутов;
- arp – просмотр и редактирование ARP таблиц;
- snmp – запуск о остановка SNMP агента;
- counters – просмотр доступных сетевых счетчиков;
- passwd – изменение пароля на доступ к TELNET серверу;
- quit – закрыть TELNET сеанс;
- ? или help – посмотреть список доступных команд.

4. Пример выполнения лабораторной работы.

4.1. Исходные данные.

Исходная конфигурация сети:



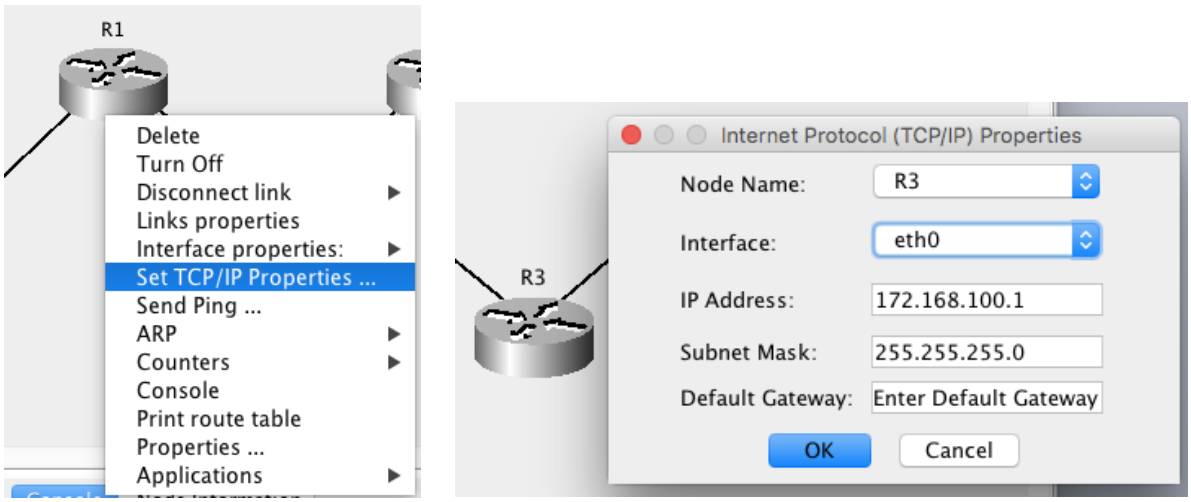
Структура исследуемой сетевой архитектуры (var0.jnst).

1. Файл со схемой сети: var0.jnst
2. Компьютер PC1 имеет IP-адрес 172.168.0.2
3. Компьютер PC2 имеет IP адрес 10.0.0.2
4. Сеть Hub1: 172.168.0.0/24
5. Сеть R1-R3: 172.168.100.0/24
6. Сеть R3-R2: 192.168.0.0/24

4.2. Порядок выполнения задания.

4.2.1. Начальная настройка узлов сети.

Сначала обеспечим корректную доставку пакетов в локальных сегментах сети. Воспользуемся в javaNetSim контекстным меню узла и диалогом Set TCP/IP Properties.



Для активных сетевых интерфейсов на всех узлах сети (на хостах, на конечных (тупиковых) маршрутизаторах, на промежуточных маршрутизаторах) установим корректные значения IP адресов (из заданного диапазона IP-сети), масок подсети и шлюзов по умолчанию.

Как видим, сети Hub1 (172.168.0.0) и R1-R3 (172.168.100.0) при использовании стандартной маски подсети для класса B были бы эквивалентными, поэтому будем использовать маску подсети, отличную от стандартной - маску 255.255.255.0.

Принтеру PC3 назначим любой свободный IP-адрес подсети Hub1, например 172.168.0.3 с маской 255.255.255.0.

Рассуждая подобным образом можно на основании имеющихся исходных данных корректно настроить сетевые параметры интерфейсов для всех узлов в сети.

Результаты начальной конфигурации интерфейсов показаны в таблице.

	IP адрес	Маска подсети	Шлюз по умолчанию:
PC1-eth0:	172.168.0.2	255.255.255.0	172.168.0.1
PC3-eth0:	172.168.0.3	255.255.255.0	172.168.0.1
R1-eth0:	172.168.0.1	255.255.255.0	172.168.100.1
R1-eth1:	172.168.100.2	255.255.255.0	172.168.100.1
R3-eth0:	172.168.100.1	255.255.255.0	-
R3-eth1:	192.168.0.2	255.255.255.0	-
R2-eth0:	192.168.0.1	255.255.255.0	192.168.0.2
R2-eth1:	10.0.0.1	255.0.0.0	192.168.0.2
PC2-eth0:	10.0.0.2	255.0.0.0	10.0.0.1

Такая конфигурация уже обеспечит доставку пакетов от PC1 к PC3, R1 и R3, но не доставку к R2 и PC2.

При попытке отправки пакета от PC1 на PC2, пакет сначала попадет на R1 (т.к. он является шлюзом по умолчанию для PC1), а оттуда на R3. Но на R3 пакет отбрасывается, т.к. маршрут для сети 10.0.0.0/8 на R3 пока отсутствует, а шлюза по умолчанию для R3 нет.

Таблицу маршрутизации можно посмотреть или через контекстное меню **Print route table** или в консоли командой **show ip route**.

Для R3 TM пока выглядит следующим образом:

```
R3# show ip route
Codes: C - connected, S - static, R - RIP,
B - BGP, O - OSPF, * - candidate default
C   192.168.0.2/255.255.255.0 is directly connected, eth1
C   172.168.100.1/255.255.255.0 is directly connected, eth0
R3# exit
```

Работа в консоли узла для программы javaNetSim похожа на реализацию в CISCO. Команды удобно дописывать табом, а помощь получать через «?» и help. Команды управления разбиты на домены, например, для роутинга:

- show ip route – показать записи в TM;
- configure terminal / ip route add – добавить запись в TM;
- configure terminal / no route add – удалить запись из TM.

4.2.2. Настройка промежуточных маршрутизаторов.

Чтобы обеспечить корректную доставку пакетов между хостами сети настроим таблицы маршрутизации на промежуточных маршрутизаторах. У нас такой маршрутизатор только один – R3.

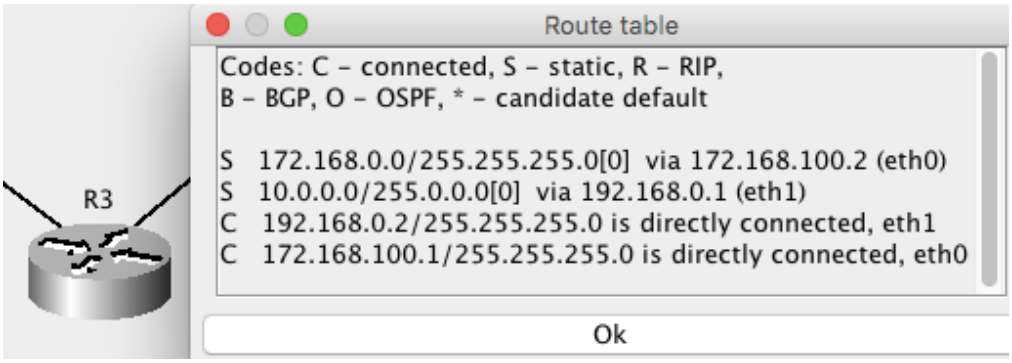
Для того чтобы пакеты между PC1 и PC2 прошли через R3 необходимо на R3 добавить маршруты для сетей 172.168.0.0/24 и 10.0.0.0/8.

```
R3# configure terminal
R3(config)# ip route add 172.168.0.0 255.255.255.0 172.168.100.2 eth0
Route added.
R3(config)# ip route add 10.0.0.0 eth0 255.0.0.0 192.168.0.1 eth1
Route added.
R3(config)# end
R3# exit
```

Чтобы проверить корректность добавления маршрута необходимо вновь вывести таблицу маршрутизации или в консоли маршрутизатора R3:

```
R3# show ip route
S   172.168.0.0/255.255.255.0[0]  via 172.168.100.2 (eth0)
S   10.0.0.0/255.0.0.0[0]  via 192.168.0.1 (eth1)
C   192.168.0.2/255.255.255.0 is directly connected, eth1
C   172.168.100.1/255.255.255.0 is directly connected, eth0
R3# exit
```

или в контекстном меню javaNetSim для нужного маршрутизатора **Print route table**:



Теперь любой пакет, попавший на R3 и имеющий в качестве подсети назначения 172.168.0.0/255.255.255.0 будет направлен на маршрутизатор R1.

Аналогичным образом можно настроить таблицы маршрутизации на остальных промежуточных маршрутизаторах сети.

В данной сети таких маршрутизаторов больше нет.

4.2.3. Проверка настроек сети.

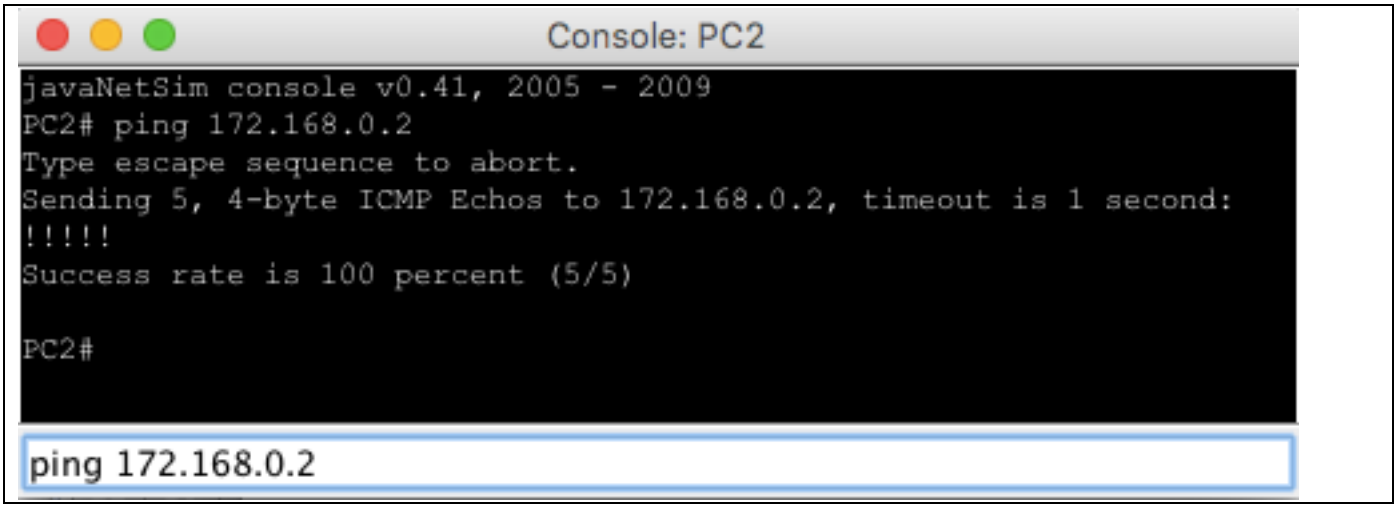
Для проверки корректности настроек выполним эхо-запросы с PC1 на PC2 и наоборот.

Отправим через контекстное меню PC1 эхо-запрос на PC2 и проанализируем результаты в главной Console javaNetSim:

```
PC1 Created Echo Request packet to 10.0.0.2
PC1 Sending packet from ProtocolStack (to 172.168.0.1).
...
PC2 ProtocolStack received packet from local Interface.
PC2 Confirmed Packet is for this Network Layer Device.
PC2 Created Echo Reply packet to 172.168.0.2
PC2 Sending packet from ProtocolStack (to 10.0.0.1).
...
PC1 ProtocolStack received packet from local Interface.
PC1 Confirmed Packet is for this Network Layer Device.
PC1 Echo reply packet received from 10.0.0.2
```

Как видим, на эхо-запрос пришёл ответ, следовательно таблица маршрутизации настроена верно.

Проверку можно выполнять и в консоли соответствующего узла командой ping, например:



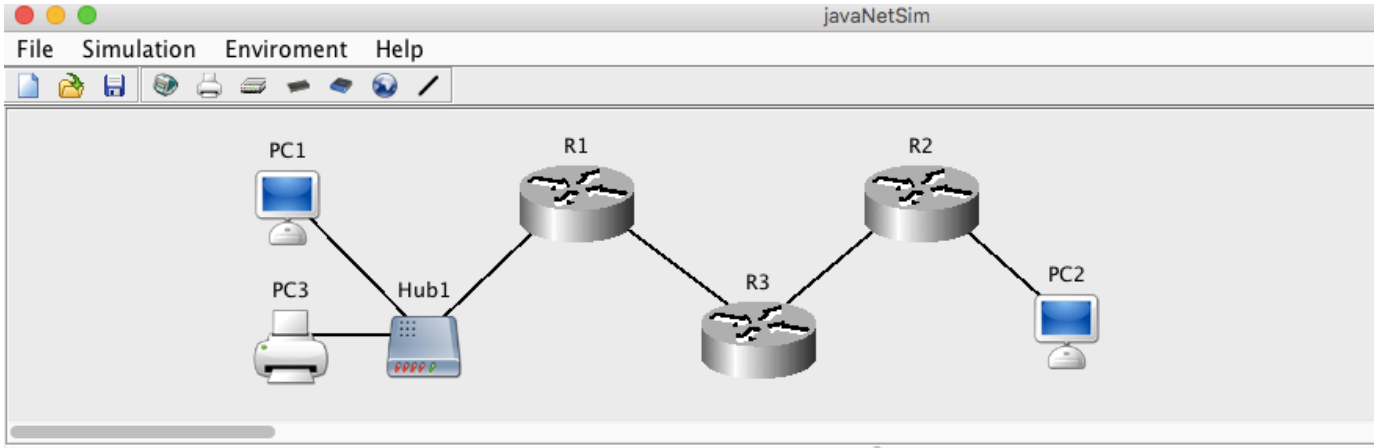
Аналогичным образом можно проверить связь между остальными узлами в сети. В данной сети маршрутизаторов больше нет.

4.3. Пример отчёта.

4.3.1. Вычисление варианта задания.

Li Yurij первая буква в имени Y имеет номер 25; вычисляем (25 MOD 10)+1 = 5+1 = 6. Получили вариант 6.

4.3.2. Эскиз схемы сети и итоговые настройки конфигурация IP узлов сети.



Console | Node Information

Name: R3	Default gateway: Gateway Not Set			
Interface: cua0	Type: Console	MAC address: Not Applicable	IP address: Not Applicable	Subnet mask: Not Applicable
Interface: eth0	Type: Copper Ethernet	MAC address: 80:2C:82:AD:62:70	IP address: 172.168.100.1	Subnet mask: 255.255.255.0
Interface: eth1	Type: Copper Ethernet	MAC address: 7A:64:95:96:83:31	IP address: 192.168.0.2	Subnet mask: 255.255.255.0
Name: R2	Default gateway: 192.168.0.2			
Interface: cua0	Type: Console	MAC address: Not Applicable	IP address: Not Applicable	Subnet mask: Not Applicable
Interface: eth0	Type: Copper Ethernet	MAC address: B4:82:36:4F:71:B6	IP address: 192.168.0.1	Subnet mask: 255.255.255.0
Interface: eth1	Type: Copper Ethernet	MAC address: 44:29:3B:9C:20:7E	IP address: 10.0.0.1	Subnet mask: 255.0.0.0
Interface: ser0	Type: Serial	MAC address: Not Applicable	IP address: IP Address not set	Subnet mask: 0.0.0.0
Name: PC3	Default gateway: 172.168.0.1			
Interface: eth0	Type: Copper Ethernet	MAC address: C0:C6:3A:39:8C:72	IP address: 172.168.0.3	Subnet mask: 255.255.255.0
Name: R1	Default gateway: 172.168.100.1			
Interface: cua0	Type: Console	MAC address: Not Applicable	IP address: Not Applicable	Subnet mask: Not Applicable
Interface: eth0	Type: Copper Ethernet	MAC address: BF:87:56:58:73:C7	IP address: 172.168.0.1	Subnet mask: 255.255.255.0
Interface: eth1	Type: Copper Ethernet	MAC address: 98:70:3A:7F:B9:5E	IP address: 172.168.100.2	Subnet mask: 255.255.255.0
Interface: ser0	Type: Serial	MAC address: Not Applicable	IP address: IP Address not set	Subnet mask: 0.0.0.0
Name: PC2	Default gateway: 10.0.0.1			
Interface: cua0	Type: Console	MAC address: Not Applicable	IP address: Not Applicable	Subnet mask: Not Applicable
Interface: eth0	Type: Copper Ethernet	MAC address: AE:A9:C2:19:97:31	IP address: 10.0.0.2	Subnet mask: 255.0.0.0
Name: PC1	Default gateway: 172.168.0.1			
Interface: cua0	Type: Console	MAC address: Not Applicable	IP address: Not Applicable	Subnet mask: Not Applicable
Interface: eth0	Type: Copper Ethernet	MAC address: 40:98:1A:84:59:80	IP address: 172.168.0.2	Subnet mask: 255.255.255.0

4.3.3. Таблицы маршрутизации для маршрутизаторов R1-R3.

R1

```
S*  default/0.0.0.0[0]  via 172.168.100.1 (eth0)
C   172.168.100.2/255.255.255.0 is directly connected, eth1
C   172.168.0.1/255.255.255.0 is directly connected, eth0
```

R2

```
S*  default/0.0.0.0[0]  via 192.168.0.2 (eth0)
C   10.0.0.1/255.0.0.0 is directly connected, eth1
C   192.168.0.1/255.255.255.0 is directly connected, eth0
```

R3

```
S   172.168.0.0/255.255.255.0[0]  via 172.168.100.2 (eth0)
S   10.0.0.0/255.0.0.0[0]  via 192.168.0.1 (eth1)
```

```
C    192.168.0.2/255.255.255.0 is directly connected, eth1
C    172.168.100.1/255.255.255.0 is directly connected, eth0
```

4.3.4. Результаты эхо-запросов между узлами PC1-PC3.

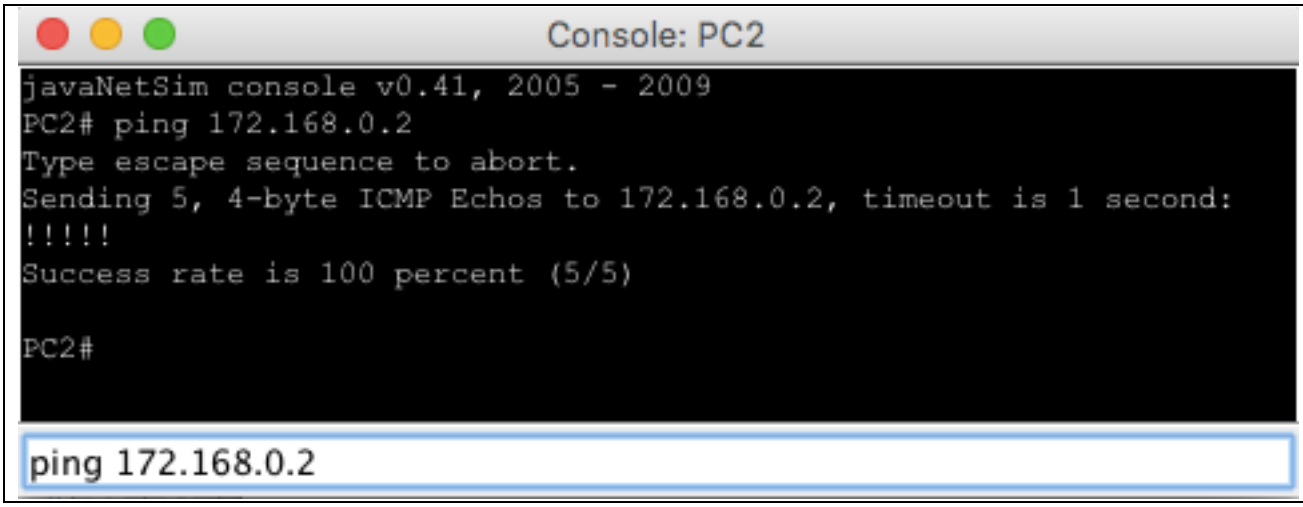
PC1 → PC2

```
PC1# ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 4-byte ICMP Echos to 10.0.0.2, timeout is 1 second:
!!!!!
Success rate is 100 percent (5/5)
```

PC1 → PC3

```
PC1# ping 172.168.0.3
Type escape sequence to abort.
Sending 5, 4-byte ICMP Echos to 172.168.0.3, timeout is 1 second:
!!!!!
Success rate is 100 percent (5/5)
```

PC2 → PC1



PC2 → PC3

```
PC2# ping 172.168.0.3
Type escape sequence to abort.
Sending 5, 4-byte ICMP Echos to 172.168.0.3, timeout is 1 second:
!!!!!
Success rate is 100 percent (5/5)
```

4.3.5. Обоснование правильности и оптимальности выбранных маршрутов.

Так как альтернативных путей доставки пакетов в этой сети нет, то построенные маршруты, как единственно возможные, уже являются самыми короткими и оптимальными.

ПЕРЕД ЗАВЕРШЕНИЕМ СЕАНСА

РАБОТЫ С javaNetSim

СОХРАНЯЙТЕ РЕЗУЛЬТАТЫ РАБОТЫ

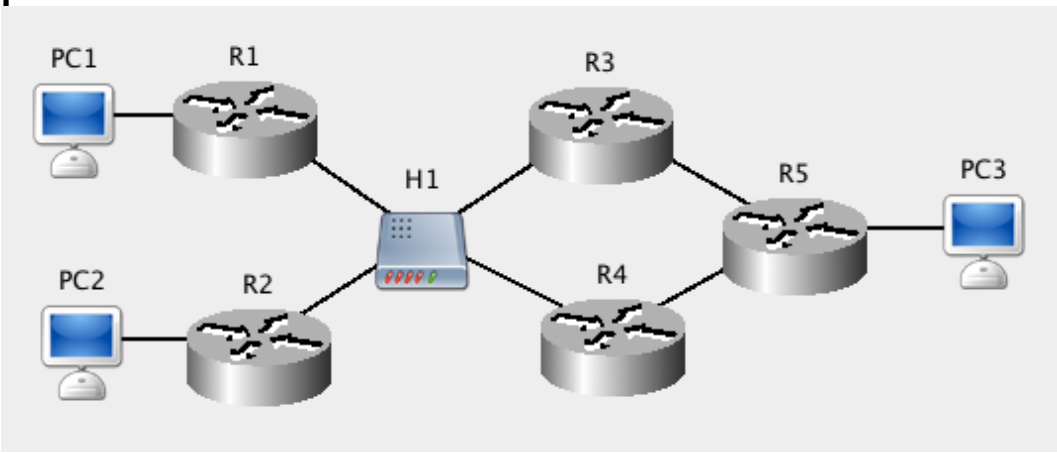
ВСЕГДА С НОВЫМ ИМЕНЕМ ФАЙЛА,

ИНАЧЕ ВНЕСЁННЫЕ ИЗМЕНЕНИЯ

МОГУТ БЫТЬ ПОТЕРЯНЫ!

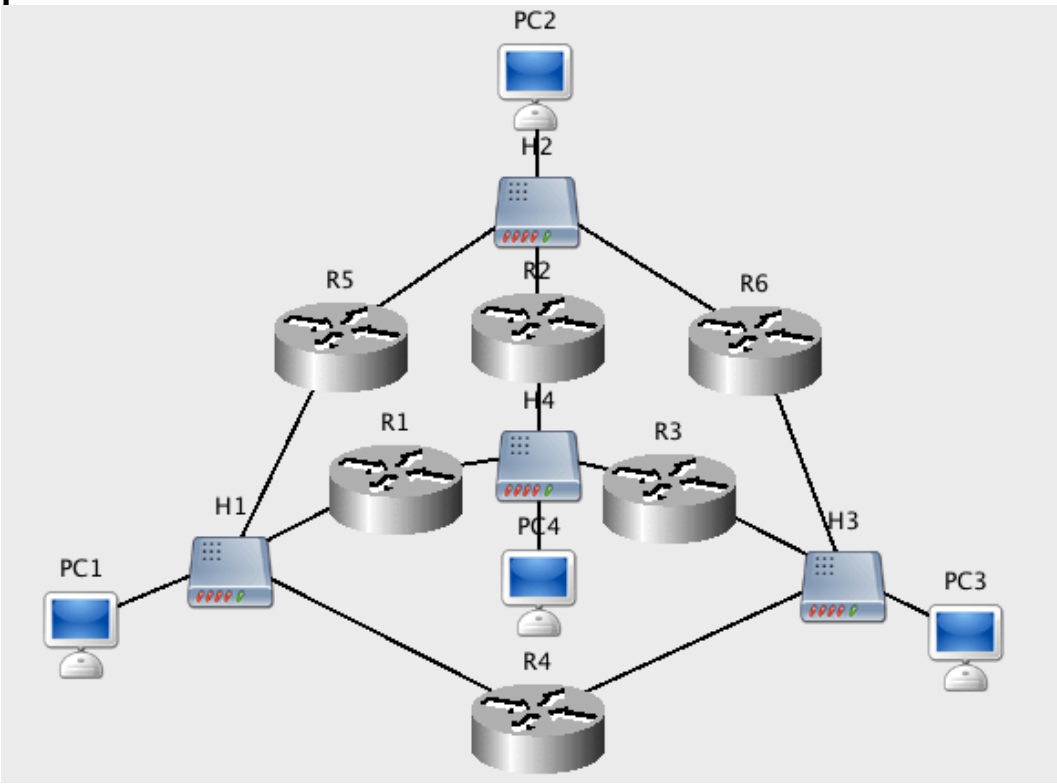
5. Варианты заданий.

Вариант 1.



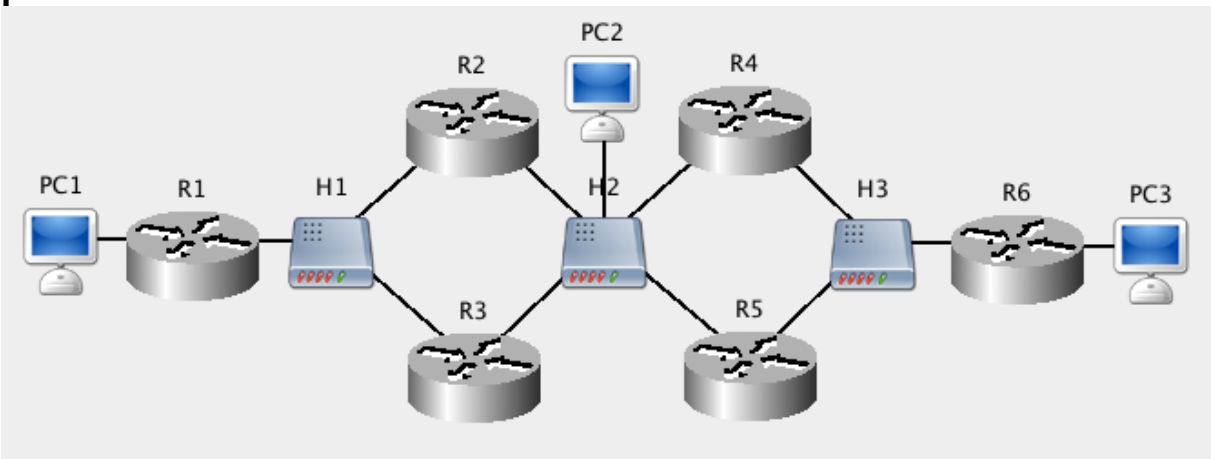
1. Файл со схемой: var1.jnst
2. Компьютер PC1: 10.0.0.5
3. Компьютер PC2: 10.0.0.130
4. Компьютер PC3: 10.0.0.194
5. Сеть H1: 192.168.0.0
6. Сеть R3-R5: 192.168.8.0
7. Сеть R4-R5: 192.168.16.0

Вариант 2.



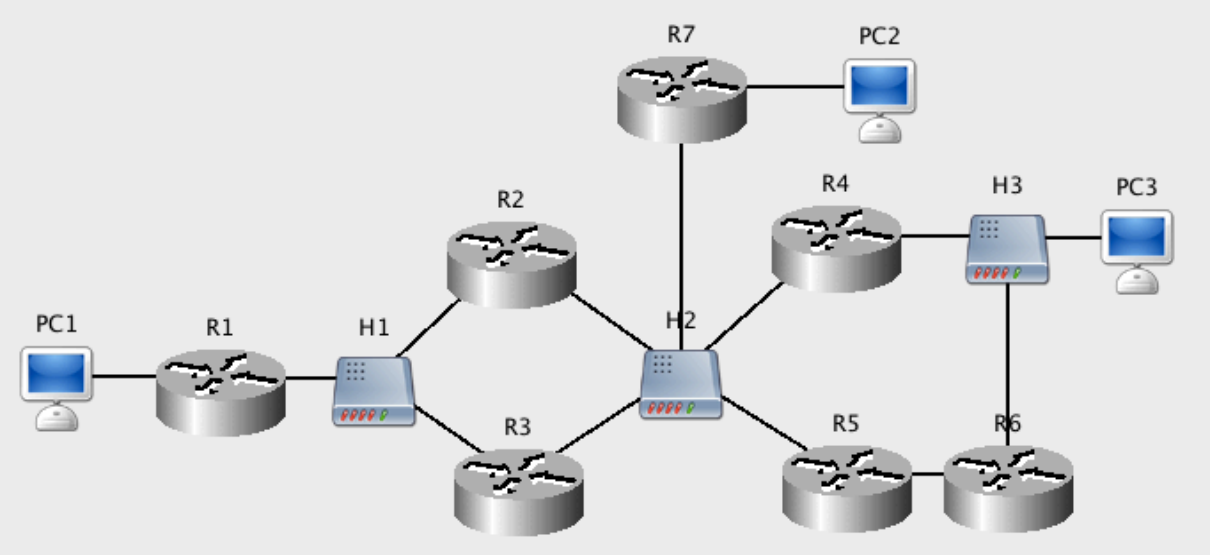
1. Файл со схемой: var2.jnst
2. Компьютер PC1: 204.188.45.1
3. Компьютер PC2: 204.188.45.65
4. Компьютер PC3: 204.188.45.129
5. Компьютер PC4: 204.188.45.196
6. Сеть H4: 204.188.45.192

Вариант 3.



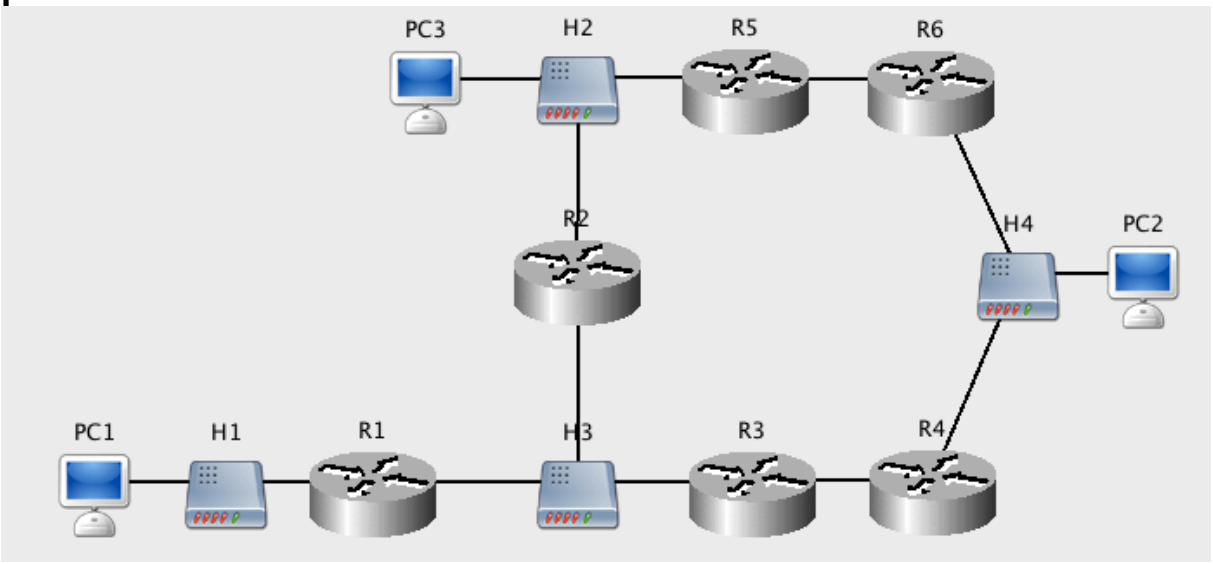
1. Файл со схемой: var3.jnst
2. Компьютер PC1: 192.115.128.1
3. Компьютер PC2: 192.115.112.4
4. Компьютер PC3: 192.115.88.2
5. Сеть H1: 192.115.120.0
6. Сеть H2: 192.115.112.0
7. Сеть H3: 192.115.96.0

Вариант 4.



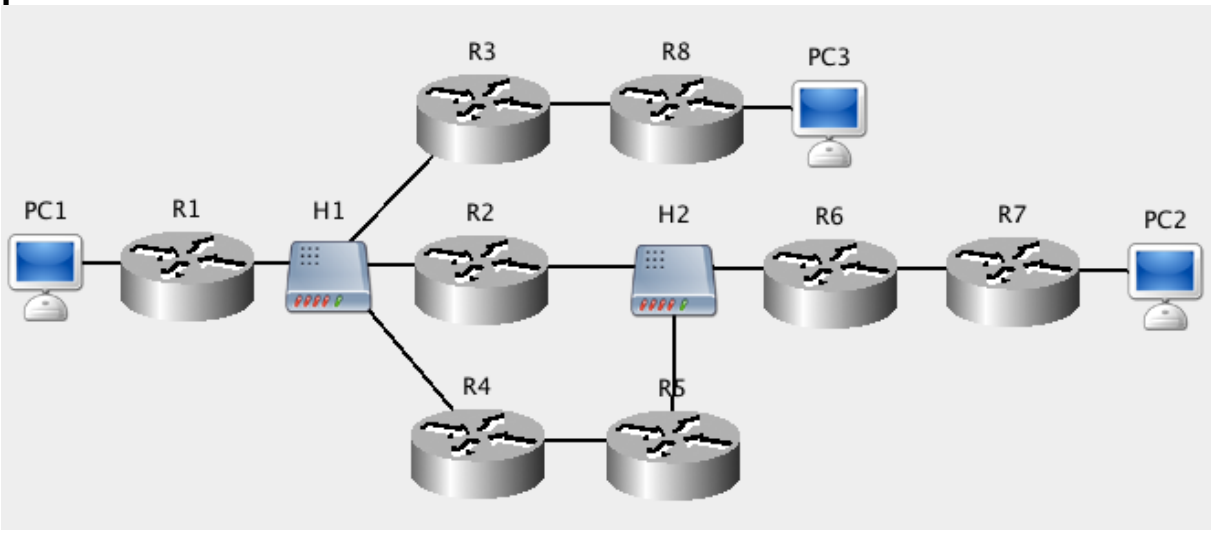
1. Файл со схемой: var4.jnst
2. Компьютер PC1: 172.168.0.100
3. Компьютер PC2: 172.168.1.100
4. Компьютер PC3: 172.168.2.100
5. Сеть H2: 172.168.3.0
6. Сеть R5-R6: 172.168.4.0

Вариант 5.



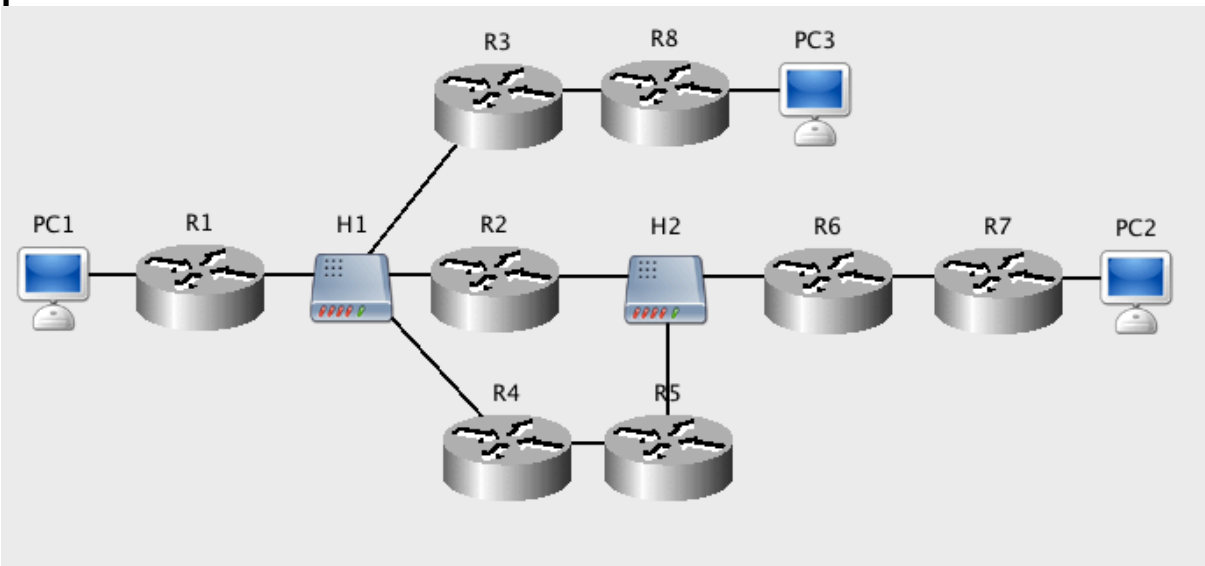
1. Файл со схемой: var5.jnst
2. Компьютер PC1: 192.168.0.100
3. Компьютер PC2: 192.168.2.100
4. Компьютер PC3: 192.168.1.101
5. Сеть H3: 192.168.3.0
6. Сеть R3-R4: 192.168.4.0
7. Сеть R5-R6: 192.168.5.0

Вариант 6.



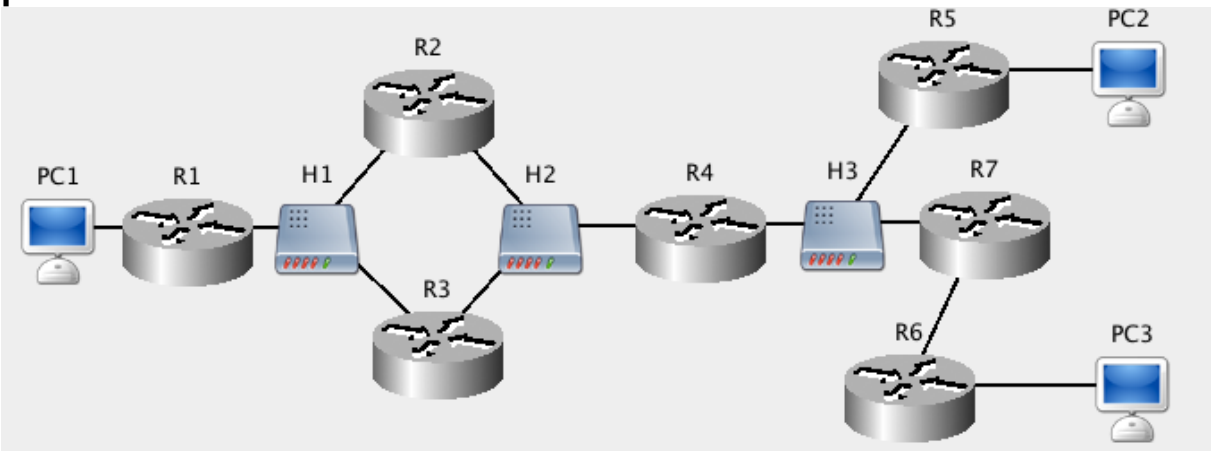
1. Файл со схемой: var6.jnst
2. Компьютер PC1: 192.168.0.2
3. Компьютер PC2: 192.168.0.250
4. Компьютер PC3: 192.168.0.34
5. Сеть H1: 192.168.0.96
6. Сеть H2: 10.120.0.0
7. Сеть R3-R8: 11.0.0.0
8. Сеть R4-R5: 172.168.4.0
9. Сеть R6-R7: 10.159.0.0

Вариант 7.



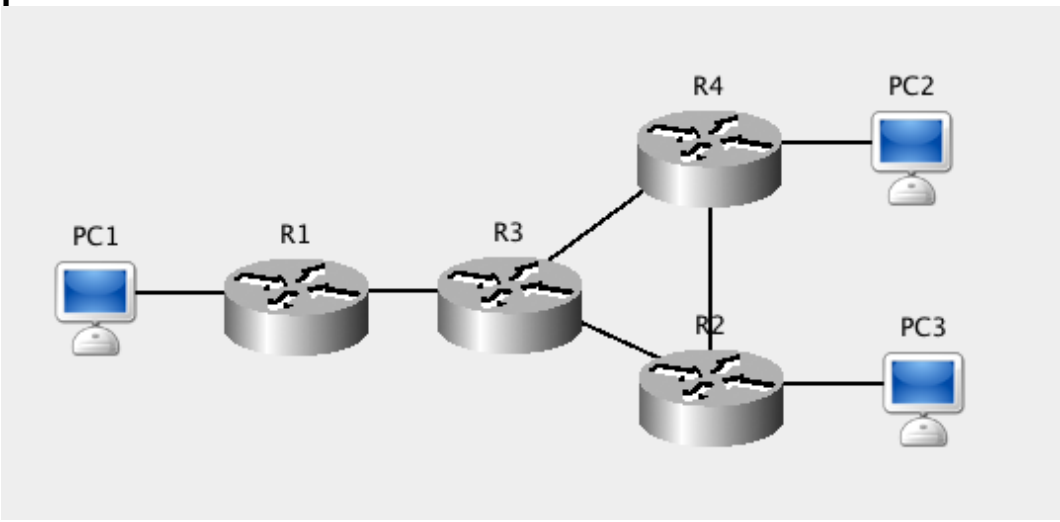
1. Файл со схемой: var7.jnst
2. Компьютер PC1: 199.0.5.2
3. Компьютер PC2: 199.0.5.250
4. Компьютер PC3: 199.0.5.52
5. Сеть H1: 199.0.5.96
6. Сеть H2: 11.120.0.0
7. Сеть R3-R8: 12.0.0.0
8. Сеть R4-R5: 172.168.4.0
9. Сеть R6-R7: 11.159.0.0

Вариант 8.



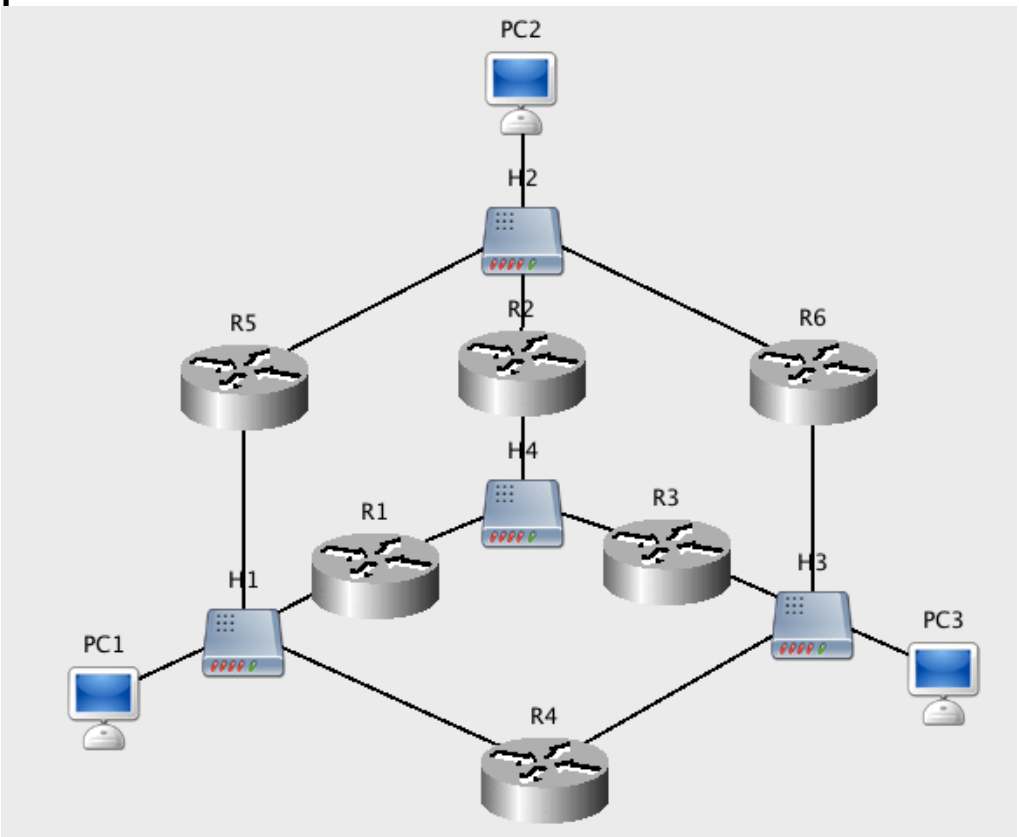
1. Файл со схемой: var8.jnst
2. Компьютер PC1: 192.115.128.1
3. Компьютер PC2: 192.115.100.1
4. Компьютер PC3: 192.115.88.2
5. Сеть H1: 192.115.120.0
6. Сеть H2: 192.115.112.0
7. Сеть H3: 192.115.108.0
8. Сеть R6-R7: 192.115.96.0

Вариант 9.



1. Файл со схемой: var9.jnst
2. Компьютер PC1: 10.0.0.2
3. Компьютер PC2: 10.0.0.6
4. Компьютер PC3: 10.0.0.10
5. Сеть R1-R3: 192.168.0.0
6. Сеть R3-R4: 192.168.1.0
7. Сеть R3-R2: 192.168.2.0
8. Сеть R4-R2: 192.168.3.0

Вариант 10.



1. Файл со схемой: var10.jnst
2. Компьютер PC1: 204.188.45.1
3. Компьютер PC2: 204.188.45.65
4. Компьютер PC3: 204.188.45.129
5. Сеть H4: 204.188.45.192