Сети ЭВМ и телекоммуникации

Лабораторная работа №2.

«Моделирование локальной сети и настройка маршрутизации с использованием пакета моделирования javaNetSim»

Цель:

Научиться настраивать проектировать сети, распределять сетевые адреса и настраивать таблицы маршрутизации для локальной вычислительной сети с использованием учебной программы моделирования «javaNetSim». Дистрибутив системы моделирования можно либо скачать с сайта https://sourceforge.net/projects/javanetsim/, либо воспользоваться дистрибутивом, приложенным к заданию.

Задание:

1. С помощью программы javaNetSim построить сеть, согласно варианту. Сеть состоит из нескольких маршрутизаторов, структура связей который приведена в варианте задания. К каждому маршрутизатору подключается от 1 до 5 хостов (хосты не приведены на схеме, их расположение выбирается самостоятельно, однако общее число хостов должно быть не менее 20). Также в варианте задания указаны адреса для подсети и маска.
2. Задать ip-адреса, маски подсети и шлюзы по-умолчанию для всех узлов сети, чтобы обеспечить корректную доставку эхо-запроса от узла К1 к узлу К2 и эхо-ответа обратно. Пояснить как ы это сделали.
3. Выполнить эхо-запрос с узла К1 на узел К2. На основе полученного результата разобраться и кратко описать последовательность прохождения пакетов по сети.
4. Добавить на узле К1 статическую ARP запись для доступа к узлу К3. Подождать устаревания ARP-таблиц и выполнить эхо-запрос с К1 на К2. Объяснить результат. Выяснить, как изменяется общее количество пакетов, генерируемых на узлах сети, при пустых ARP таблицах, сразу после их динамического заполнения, а также в случае их статического заполнения на ряде узлов сети.
5. Настроить таблицы маршрутизации на маршрутизаторах, чтобы добиться доставки пакетов от узла К1 к узлу К2 и обратно, от узла К2 к К3 и обратно, от узла К3 к К1 и обратно. Пакеты должны доходить до узлов **кратчайшим** путем
6. На компьютере К1 запустить SNMP агента. Порт и имя группы доступа выбираются самостоятельно.
7. С компьютера К2 отправить запрос(ы) get, и получить переменные.
8. С компьютера К2 отправить запрос(ы) getnext для переменных. Объяснить полученные результаты;
9. На компьютере К2 запустить TELNET сервер. Порт и пароль выбрать самостоятельно;
10. С компьютера К3 по протоколу TELNET подключиться к компьютеру К2. Удалить все значения из таблицы маршрутизации и ARP таблицы. Добавить в таблицу маршрутизации и ARP таблицу записи необходимые для корректной работы компьютера К2;

11. С помощью команды TELNET-сервера snmp запустить SNMP агента на К3. Проверить работоспособность snmp-сервера: с компьютера К2 попытаться получить значение SNMP переменной П2;

Примечание:

1. Задание является дифференцированным.
   - На оценку «Удовлетворительно» достаточно выполнить пункты задания 1-4
   - На оценку «Хорошо» необходимо выполнить пункты задания 1-5
   - На оценку «Отлично» необходимо выполнить все пункты задания

Варианты заданий

Варианты заданий приведены в отдельном файле graphs.pdf

Теоретическая часть

Основной задачей имитатора javaNetSim является имитация работы всех уровней стека протоколов TCP/IP. Для этого имитируется работа протоколов каждого из уровней, чем достигается полная имитация работы сети. В связи с этим имитатор javaNetSim удобен для выполнения лабораторных работ. Основные приемы работы с имитатором javaNetSim будут рассмотрены в данной главе.

Архитектура имитатора javaNetSim выглядит следующим образом. В основе лежит класс Simulation (Имитация), который содержит объекты классов Link (Линия) и Node (Узел). Этот класс предназначен для объединения устройств и линий связи в единую сеть. Класс Link содержит ссылки на объекты класса Node, и предназначен для соединения двух узлов между собой. Класс Node содержит ссылки на объекты класса Link и является наиболее общей моделью сетевого устройства.

Все реальные сетевые устройства являются производными от объекта класса Node и соответствуют модели стека протоколов TCP/IP:

- Hub (Концентратор) – DataLink Layer Device (Устройство физического уровня) – имеет пять портов, т.е. к нему возможно подключить до пяти линий связи;
- Router (Маршрутизатор) – Network Layer Device (Устройство сетевого уровня) – имеет два порта, а также стек протоколов TCP/IP (ProtocolStack);
- PC (Компьютер) – Applications Layer Device (Устройство уровня приложений) – имеет один порт, стек протоколов TCP/IP, а также возможность выполнять клиентскую или серверную часть какого-либо приложения.

Для взаимодействия с пользователем каждому сетевому устройству нужно графическое соответствие. Его обеспечивают следующие классы:

- GuiHub (Графический пользовательский интерфейс концентратора);
- GuiRouter (Графический пользовательский интерфейс маршрутизатора);
- GuiPC (Графический пользовательский интерфейс компьютера).

Как сами сетевые устройства, так и графический пользовательский интерфейс сетевых устройств должен быть единым. Этим объединением занимается класс SandBox (Рабочая область).

1. Графический интерфейс имитатора javaNetSim

Рабочая область является частью основного окна программы, представленного на рисунке 1.

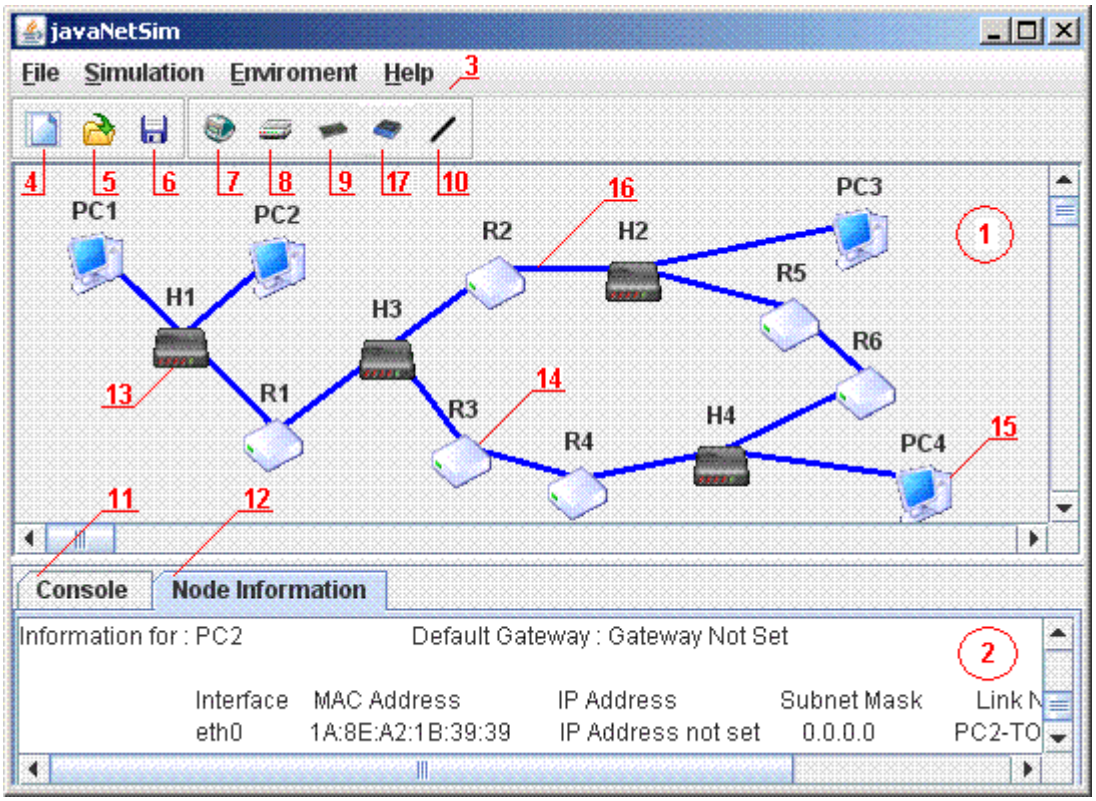


Рис. 1. Основное окно программы javaNetSim.

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).

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.

2. Главное меню программы

Меню File(файл) позволяет создавать, открывать и сохранять конфигурации сетей для их дальнейшего использования. Меню содержит пять пунктов:

- New(Новый) – создать пустую конфигурацию.
- Open...(Открыть...) – открыть существующую конфигурацию.
- Save...(Сохранить...) – сохранить текущую конфигурацию.
- Save As...(Сохранить Как...) – сохранить текущую конфигурацию под новым именем.
- Exit(Выход) – выйти из имитатора javaNetSim.

Режим проектирования сети доступен из меню Simulation(Имитация). Это меню позволяет создавать новые сетевые устройства (такие как: концентратор, маршрутизатор или компьютер) и изменять сетевые параметры уже существующих устройств. Меню содержит два пункта:

- подменю Add(Добавить) – позволяет создать компьютер(PC), маршрутизатор(Router) или концентратор(Hub);
- подменю Tools(Инструменты), в котором есть пункт Set TCP/IP Properties(Установить свойства TCP/IP) позволяющий изменить свойства TCP/IP.

В имитаторе javaNetSim задание IP-адреса узла, маски подсети и шлюза по умолчанию происходит через диалог "Internet Protocol (TCP/IP) Properties", вызов которого осуществляется через меню "Simulation -> Tools -> Set TCP/IP Properties".

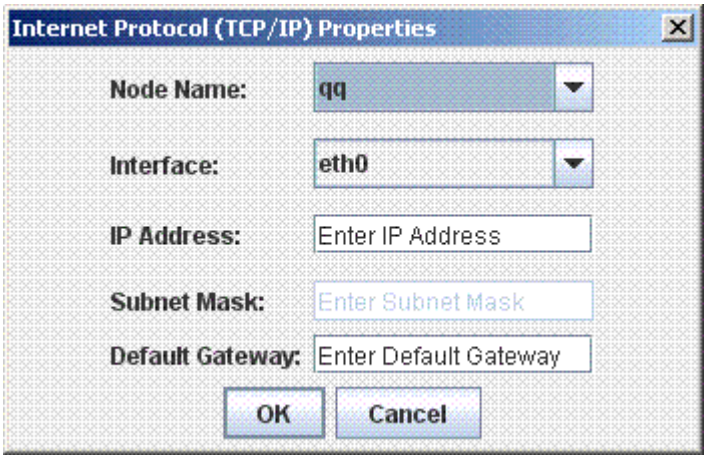


Рис. 2. Установка параметров TCP/IP.

В этом окне (рис. 2) для выбранного устройства (Node Name) и интерфейса (Interface) можно задать IP-адрес (IP Address) и маску подсети (Subnet Mask) для интерфейса и шлюз по-умолчанию (Default Gateway) для узла. Для компьютера доступен всего один интерфейс, для маршрутизатора – два.

Управление параметрами имитатора доступно из меню Environment(Окружение) и позволяет изменять режим отображения информации, а также очищать область вывода результатов. Меню содержит четыре пункта:

- Clear Console(Очистить консоль) – удаляет все записи из вкладки "консоль";
- Clear Node Information(Очистить информацию об устройствах) – удаляет все записи из вкладки "информация об устройствах";

- Show simulation messages for:(Показывать сообщения имитатора для:) – позволяет задать режим вывода на вкладку "консоль" сообщений только определенных уровней стека протоколов TCP/IP.

  Есть возможность выбрать следующие уровни: Link and DataLink Layers(Физический и канальный уровни), Network Layer(Сетевой уровень), Transport Layer(Транспортный уровень), Application Layer(Уровень приложений);
- Show headers:(Показывать заголовки) – позволяет задать режим вывода на вкладку "консоль" сообщений с названиями уровней и/или с типами пакетов.

С помощью меню "Environment -> Show simulation messages for:" можно отключить сообщение от тех уровней стека протоколов TCP/IP в которых нет необходимости. Это уменьшит количество информации выводимой в "консоль" и облегчит поиск нужных данных.

3. Контекстное меню

Контекстное меню, вызываемое щелчком правой кнопкой мыши, отличается для устройств работающих на разных уровнях стека протоколов TCP/IP. На рис. 3 изображены контекстные меню соответственно для устройств: физического уровня - концентратор (а), сетевого уровня - коммутатор (б) и прикладного уровня - компьютер (в).

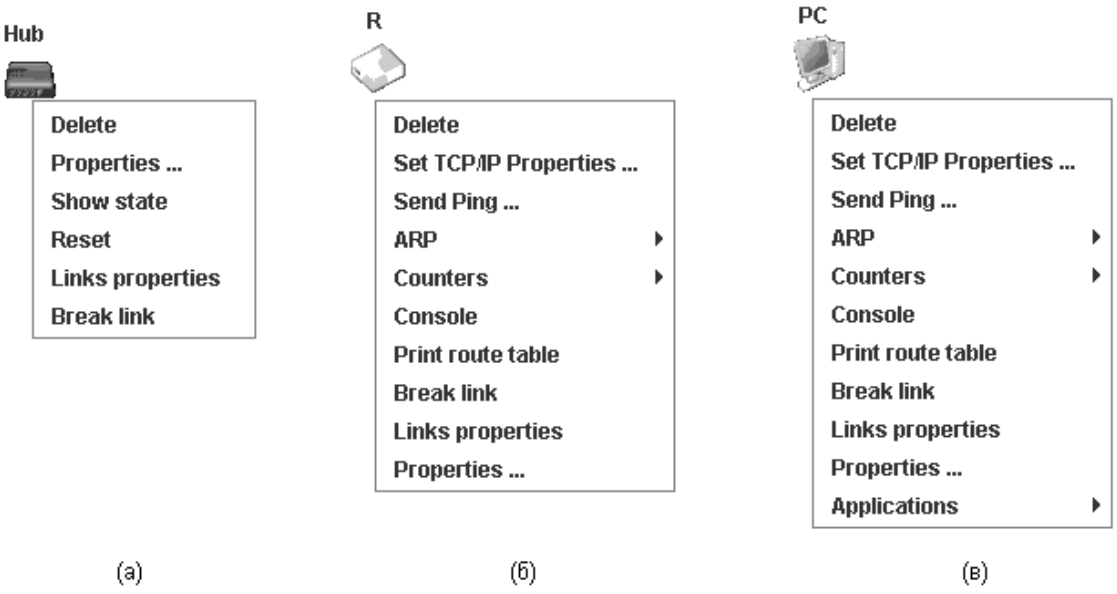


Рис. 3. Контекстное меню.

Основные пункты контекстного меню, общие для всех устройств перечислены ниже.

- Delete(Удалить) – без подтверждения удаляет выбранное сетевое устройство из текущей конфигурации.
- Properties(Свойства) – вызывает диалог, показывающий сетевые настройки выбранного устройства. Для каждого интерфейса показывается MAC адрес, IP адрес, маска подсети, название подключенной линии связи. Также для устройства указаны имя и шлюз по умолчанию.
- Break Link(Разорвать линию связи) – вызывает диалог, в котором можно выбрать интерфейс, линию связи которого требуется разорвать.
- Links Properties(Свойства линий связи) – позволяет установить свойства линии связи.

При выборе пункта 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. Коэффициент пропускания задается для интерфейса (eth0, eth1 и т.д.).

В меню концентратора имеются два дополнительных пункта, позволяющих следить за его состоянием и, в случае необходимости, восстанавливать исходное состояние.

- Show state(Показать состояние) – показывает текущее состояние концентратора, может принимать два значения normal(концентратор работает) и freezed(концентратор был остановлен из-за ошибки).
- Reset(Перезагрузить) – если концентратор находится в состоянии останова, то эта команда вернет его в рабочее состояние.

В меню устройств работающих на сетевом уровне (маршрутизаторы и компьютеры) в дополнение к основным имеются ещj шесть пунктов:

- Set TCP/IP Properties(Установка свойств TCP/IP) – вызывает диалог позволяющий изменить свойства TCP/IP;
- Send Ping...(Послать эхо-запрос) – позволяет послать эхо-запрос адресату;
- ARP – подменю позволяет работать с таблицей протокола ARP на выбранном устройстве;
- Counters – подменю содержит два пункта:
  - Show Packet Counters(Показать счетчики пакетов) – показывает счетчики для пакетов протоколов ARP, IP, UDP, TCP
  - Reset Packet Counters(Сбросить счетчики пакетов) – устанавливает все счетчики на выбранном устройстве в ноль;
- Console – вызывает командную строку, позволяющую настраивать таблицы маршрутизации, ARP таблицы и др.;
- Print route table(Показать таблицу маршрутизации) – выводит на вкладку "консоль" таблицу маршрутизации выбранного сетевого устройства.

Пункт контекстного меню Send Ping...(Послать эхо-запрос) – вызывает диалог, в котором можно настроить параметры эхо-запроса. Во время передвижения пакетов по сети во вкладке "консоль" должны появится сообщения, аналогичные приведенным ниже:

> PC1 Echo Request Packet Network Created Echo
> PC1 Echo Request Packet Network Created Echo
> Request packet to 10.0.0.2
> ...
> PC1 Echo Reply Packet Network Echo reply packet
> received from 10.0.0.2

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.

В контекстное меню компьютеров, т.е. устройств поддерживающих уровень приложений добавляется еще один пункт: подменю Applications(Приложения), которое позволяет работать с протоколами: Echo(UDP,TCP), SNMP и TELNET.

4. Командная строка

Для запуска командной строки из контекстного меню выберем "Console", появится окно консоли (рис. 4).

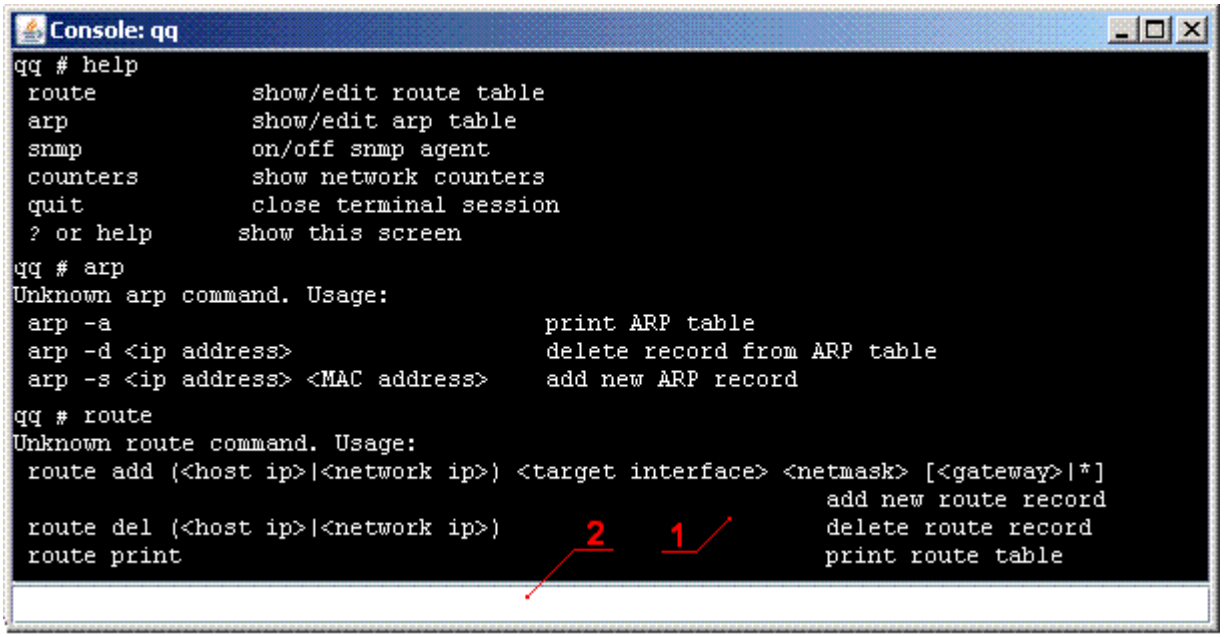


Рис. 4. Консоль.

Окно разделено на 2 части:

- область для сохранения результата выполнения команд (1);
- командная строка, в которой можно вводить команды на выполнение (2).

В консоли могут использоваться следующие специальные клавиши:

- <Enter> – выполнить введенную команду;
- стрелки вверх/вниз – просмотр истории команд;
- <ESC> – очистить командную строку;
- Ctrl+D – закрыть консоль.

В командной строке с помощью команды route можно выполнить настройку статической таблицы маршрутизации. Для этого предназначена команда route.

Описание синтаксиса команды route:

- route add (<ip адрес устройства>|<ip адрес сети>) <интерфейс> <маска подсети> [<шлюз>| *] – добавить новый маршрут для сети или устройства;
- route del (<ip адрес устройства>|<ip адрес сети>) – удалить существующий маршрут для указанного IP адреса устройства или сети;
- route print – просмотреть список существующих маршрутов.

С помощью команды arp можно выполнить настройку таблицы ARP:

- arp -a просмотреть ARP таблицу;
- arp -d <ip address> удалить из ARP таблицы запись об IP адресе;

- arp -s <ip address> <MAC address> добавить ARP запись связывающую IP и MAC адреса.

С помощью команды snmp можно управлять snmp агентом:

- snmp (on|<port number>) [community name] включить SNMP агента. Если порт не указан (значение on), то по умолчанию выбирается порт 161. Если не указано имя группы доступа, то берется значение по умолчанию public;
- snmp off выключить SNMP агента.

При вводе команд route, arp и snmp без параметров будет выведена краткая информация по их использованию.

5. Работа 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 и TELNET.

5.1 Работа с протоколом Echo

Имитатор javaNetSim позволяет использовать протоколы UDP или TCP в качестве транспортных протоколов для протокола Echo. Для установки echo-сервера в режим прослушивания порта в контекстном меню 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 вести номер порта, на котором выбранное приложение будет ожидать сообщения. После этого с любого другого узла можно отсылать сообщения на тот узел, на котором запущен echo-сервер 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:

- IP-адрес компьютера, на котором запущен echo-сервер;
- номер порта на котором echo-сервер ожидает сообщения;
- сообщение – любой текст;
- количество посылаемых сообщений, т.е. количество копий сообщения отправляемых echo-серверу.

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.

5.2. Работа с протоколом SNMP

В имитаторе javaNetSim предусмотрено несколько функций для работы с протоколом SNMP:

- запуск SNMP агента на объекте управления;
- остановка SNMP агента на объекте управления;

- посылка SNMP запросов агенту.

Для запуска SNMP агента необходимо выбрать пункт контекстного меню "Application" -> "Start SNMP Agent" и задать два параметра:

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

Для остановки SNMP агента необходимо выбрать пункт контекстного меню "Application" -> "Stop SNMP Agent".

Для того, чтобы послать запрос SNMP агенту необходимо выбрать пункт контекстного меню "Application" -> "Send SNMP message" и заполнить поля диалога, приведенные на рис. 5.

- IP Address – IP адрес компьютера на котором установлен SNMP агент.
- Destination Port – порт на котором SNMP агент ожидает пакеты.
- SNMP message – SNMP запрос, может принимать значения: get, getnext, set.
- Variables – SNMP переменные описываемые деревом MIB.
- Community name – имя группы доступа, которое должно совпадать с именем группы доступа установленным при создании агента.

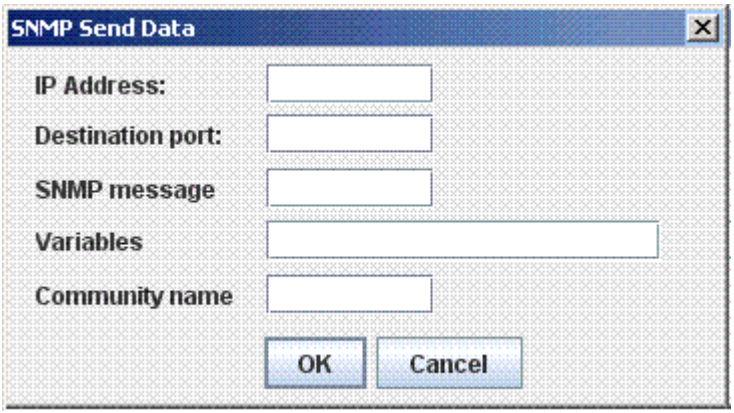


Рис.5. Создание SNMP запроса.

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 "консоль". Например:

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

Список SNMP переменных, поддерживаемых имитатором javaNet- Sim, которые имеют режим доступа "только для чтения"приведен ниже.

- 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.

5.3. Работа с протоколом TELNET

В имитаторе javaNetSim предусмотрены следующие функции для работы с протоколом TELNET:

- запуск TELNET сервера на управляемом компьютере;
- остановка TELNET сервера;
- запуск TELNET клиента.

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 остановки TELNET сервера необходимо выбрать пункт контекстного меню "Application" -> "Stop telnet server".

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 сеансе 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.

После завершения работы необходимо закрыть сеанс telnet. Закрытие сеанса telnet можно произвести тремя способами:

- набрать команду quit.
- нажать комбинацию клавиш Ctrl+D.
- просто закрыть окно терминала.

Несмотря на то, что протокол TELNET в javaNetSim реализован на очень простом уровне, это не мешает ему выполнять свои функции. В качестве примера можно привести подключения telnet-клиента к Echo- TCP-серверу.

Пример выполнения Лабораторной работы

1. Зададим IP-адреса и маски подсети для маршрутизаторов R1 и R2.

   Как видно, сети 172.168.100.0 (Hub1) и 172.168.0.0 (R1-R2) при использовании стандартной маски подсети для класса B были бы эквивалентными, поэтому будем использовать маску подсети, отличную от стандартной, а именно маску 255.255.255.0.

   Зададим для маршрутизатора R1 на интерфейсе eth0 адрес 172.168.0.1 и маску подсети 255.255.255.0. На интерфейсе eth1 установим адрес 172.168.100.2 и маску 255.255.255.0.

   Теперь необходимо сконфигурировать маршрутизатор R2. На его интерфейсе eth0 зададим IP 172.168.100.1 и маску 255.255.255.0. Установим шлюз по умолчанию в 172.168.100.2. На интерфейсе eth1 для R2 установим любой IP-адрес из диапазона сети PC2, например 10.0.0.1 и соответствующую ему маску: 255.0.0.0.

   Для корректной маршрутизации осталось задать только шлюз по умолчанию для R1. Он будет адресом маршрутизатора R2.

2. Теперь настроим конечные узлы.

   На PC1 зададим маску подсети соответствующую новому адресному пространству – 255.255.255.0. Так как пакеты от узла PC1 в другие сети должны проходить через маршрутизатор R1, зададим шлюз по умолчанию 172.168.0.1 (адрес R1).

Аналогичные операции проведем на PC2 – установим маску подсети в 255.0.0.0, а шлюз по умолчанию в 10.0.0.1. Стоит заметить, что приведенная конфигурация не является единственно верной.

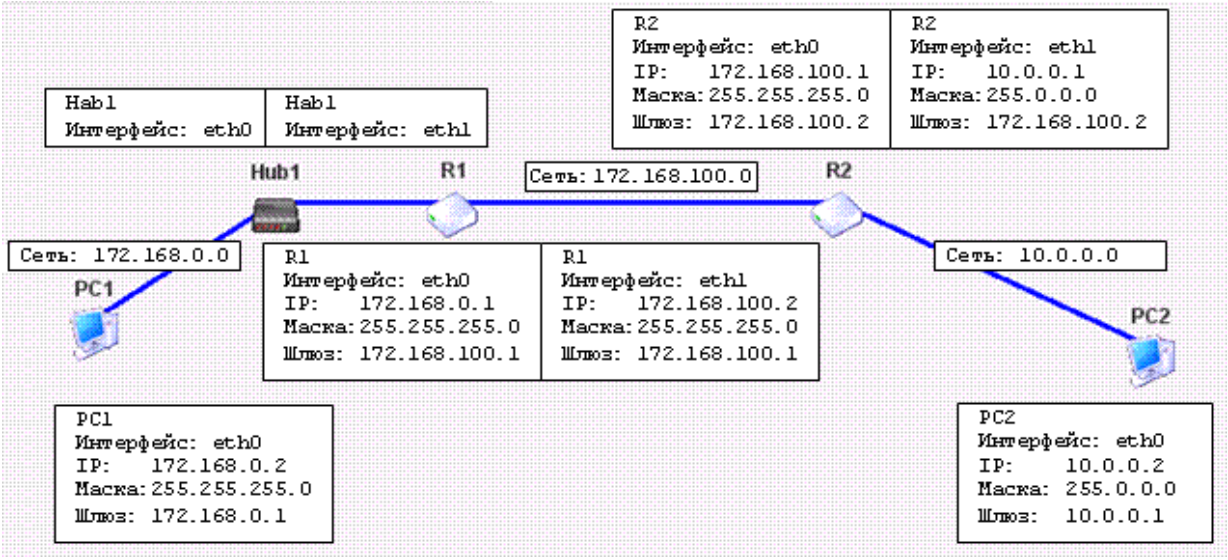


Рис. 1. Вид структуры сети, представляемой в отчет по ЛР.

Информация о состоянии текущих настроек для всех узлов сети отображается на вкладке "Node Information", которая для рассматриваемого примера после проведения настройки всех узлов сети будет иметь вид:

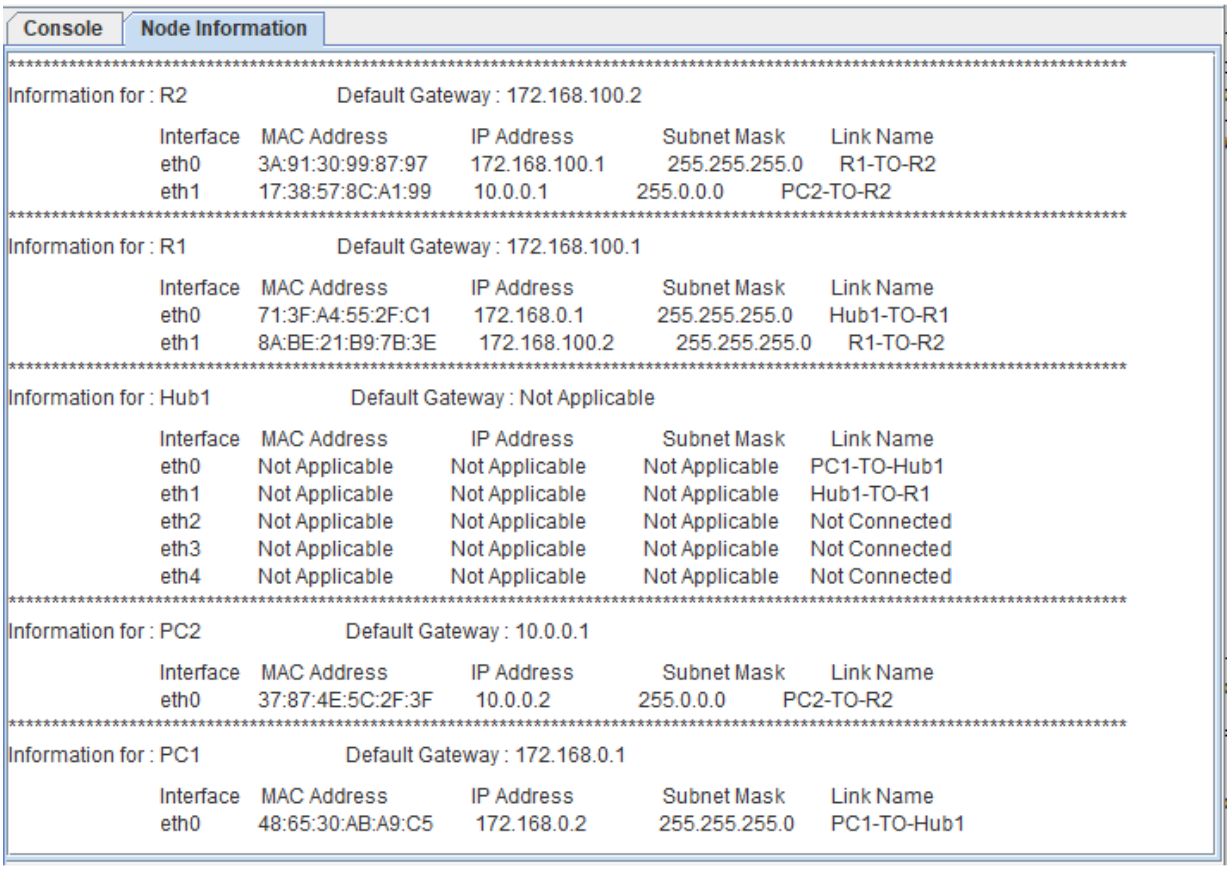
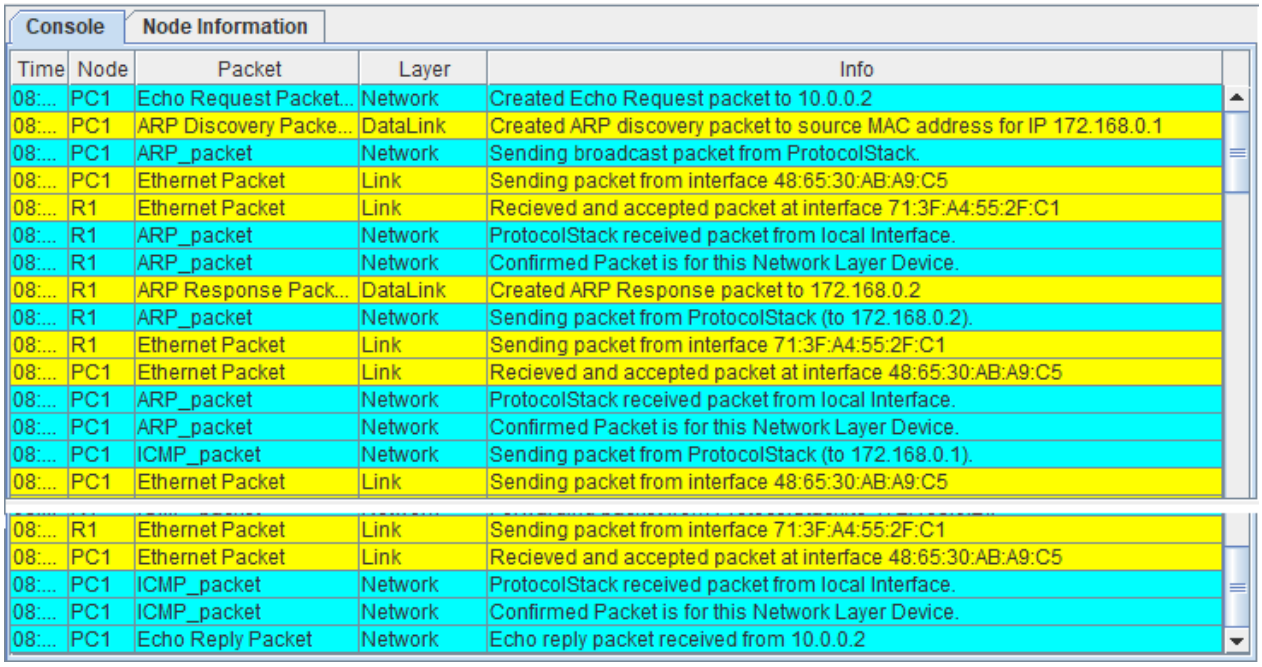


Рис. 2. Вид информация о настройках узлов сети.

3. Выполним эхо-запроса с PC1 на PC2

   Для этого на схеме сети выделим PC1, вызовем нажатием правой кнопки мышки всплывающее меню и в нем выберем опцию "Send Ping ...". В появившемся окне

укажем IP-адрес узла PC2 и нажмем кнопку OK. После отправки эхо-запроса с PC1 на PC2 в консоли будет выведен результат прохождения по сети запроса и ответа на него с использованием всех уровней сетевого взаимодействия:



Time	Node	Packet	Layer	Info
08:...	PC1	Echo Request Packet...	Network	Created Echo Request packet to 10.0.0.2
08:...	PC1	ARP Discovery Packe...	DataLink	Created ARP discovery packet to source MAC address for IP 172.168.0.1
08:...	PC1	ARP_packet	Network	Sending broadcast packet from ProtocolStack.
08:...	PC1	Ethernet Packet	Link	Sending packet from interface 48:65:30:AB:A9:C5
08:...	R1	Ethernet Packet	Link	Recieved and accepted packet at interface 71:3F:A4:55:2F:C1
08:...	R1	ARP_packet	Network	ProtocolStack received packet from local Interface.
08:...	R1	ARP_packet	Network	Confirmed Packet is for this Network Layer Device.
08:...	R1	ARP Response Pack...	DataLink	Created ARP Response packet to 172.168.0.2
08:...	R1	ARP_packet	Network	Sending packet from ProtocolStack (to 172.168.0.2).
08:...	R1	Ethernet Packet	Link	Sending packet from interface 71:3F:A4:55:2F:C1
08:...	PC1	Ethernet Packet	Link	Recieved and accepted packet at interface 48:65:30:AB:A9:C5
08:...	PC1	ARP_packet	Network	ProtocolStack received packet from local Interface.
08:...	PC1	ARP_packet	Network	Confirmed Packet is for this Network Layer Device.
08:...	PC1	ICMP_packet	Network	Sending packet from ProtocolStack (to 172.168.0.1).
08:...	PC1	Ethernet Packet	Link	Sending packet from interface 48:65:30:AB:A9:C5
08:...	R1	Ethernet Packet	Link	Sending packet from interface 71:3F:A4:55:2F:C1
08:...	PC1	Ethernet Packet	Link	Recieved and accepted packet at interface 48:65:30:AB:A9:C5
08:...	PC1	ICMP_packet	Network	ProtocolStack received packet from local Interface.
08:...	PC1	ICMP_packet	Network	Confirmed Packet is for this Network Layer Device.
08:...	PC1	Echo Reply Packet	Network	Echo reply packet received from 10.0.0.2

Рис. 3. Результат эхо-запроса с PC1 на PC2 .

Как видно из содержимого консоли (рис. 3) узел PC1 успешно получил эхо-ответ на свой запрос к узлу PC2.

Замечание!!! Следует отметить, что полученный результат может быть сохранен в виде HTML отчета, используя для этого опцию Simulation -> Generate HTML report. Кроме этого, разобравшиись с первым примером, при дальнейших исследованиях можно упростить вид консоли, отключив в ней вывод ряда протокольных стеков, например, Link и DataLink, используя для этого опцию Enviroment -> Show simulation message for.

4. Добавление статических ARP записей.

Если после выполнения эхо-запроса запомнить общее количество пакетов, сгенерированных на всех узлах сети, а затем в течении двух минут повторить эхо-запрос, то можно видеть существенное сокращение общей нагрузки на сеть и ее узлы. Так, например, в этом примере в первом случае было использовано 69 пакетов, в то время, как во втором случае из стало всего 33.

Это связано с тем, что операционные системы узлов сети в своем ARP кзше сохранили сведения о соответствии IP и MAC адресов узлов сети, которые принимали участие в передаче информаии по сети. В этом можно убедиться, если сразу после эхо-запроса выделить на схеме сети любой узел, и во всплывающем меню выбрать опции ARP -> Print ARP Table.

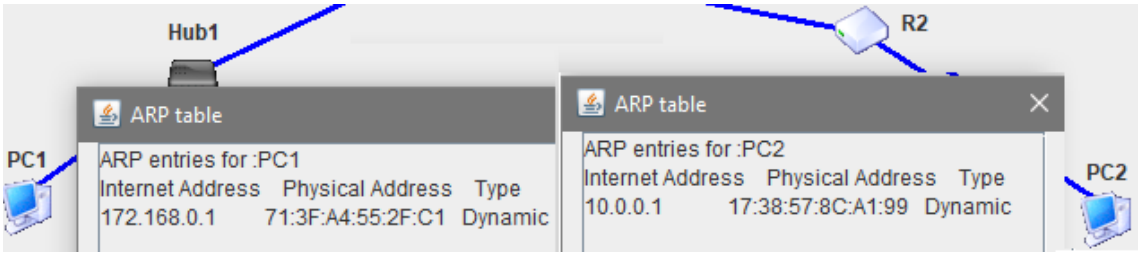


Рис. 4 Вид ARP таблиц узлов сети.

При этом возможен и иной подход, при котором, например, для маршрутизатора R1 во всплывающем меню следует выбрать опцию Console, а при ее открытии ввести команду arp -a.

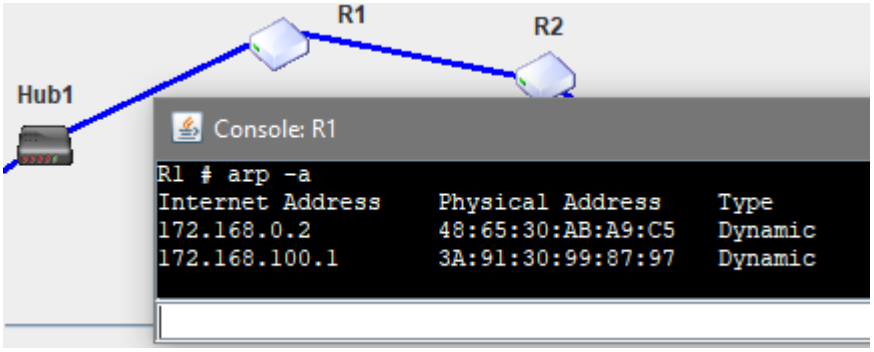


Рис. 5. Просмотр ARP таблицы из консоли.

Из приведенных выше рисунков видно, что во всех случаях записи в ARP таблицах являются динамическими. Они сохраняются операционной системой в течении двух-трех минут, после чего уничтожаются. Если структура сети является стабильной, то для уменьшения загрузки сети имеет смысл использовать статические записи ARP таблиц. Например, для того же маршрутизатора R1 можно прописать статический набор записей, аналогичных тем, что были получены операционной системой на основе ICMP и ARP пакетов, циркулирующих в сети.

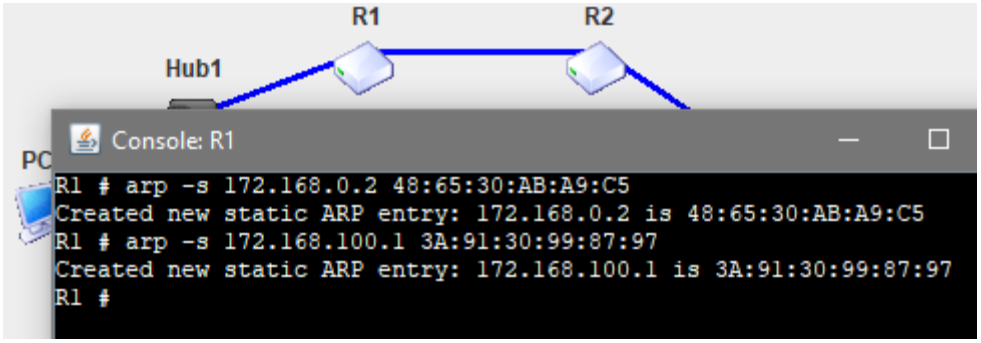


Рис. 6. Добавление статических ARP с помощью консольных команд.

Выполнение команды arp -s <IP address> <MAC address> позволит добавить статическую ARP запись, связывающую IP и MAC адреса ближайших узлов сегмента локальной сети. После выполнения этих команд следует убедиться, что они добавились в состав операционной системы (ARP -> Print ARP Table) и оказывают влияние на общую загрузку сети, снижая количество передаваемых по сети пакетов при выполнении эхо-запросов (Send Ping ...).

Замечание!!! При этом следует обратить особое внимание на то, что статическне ARP записи должны быть дабавленны на двух соседних узлах сети. Если ее добавить на узле R1, но не добавлять на PC1, то на этом участке сети будет работать ARP протокол и соответствующая запись на R1 превратиться в динамическую. И никакого снижения загрузки сети наблюдаться не будет.

Учитывая приведенное выше замечание, можно добавить соответствующие статические ARP записи на узлах R1 и R2 и добиться сокращения общего числа пакетов в сети до 57. Это происходит за счет исключения передачи ARP пакетов в сегменте R1-R2. Аналогичные настройки надо выполнить и для других сегментов сети. При этом вместо консоли можно использовать опцию "ARP -> Add static entry to ARP table..." всплывающего меню узла сети.

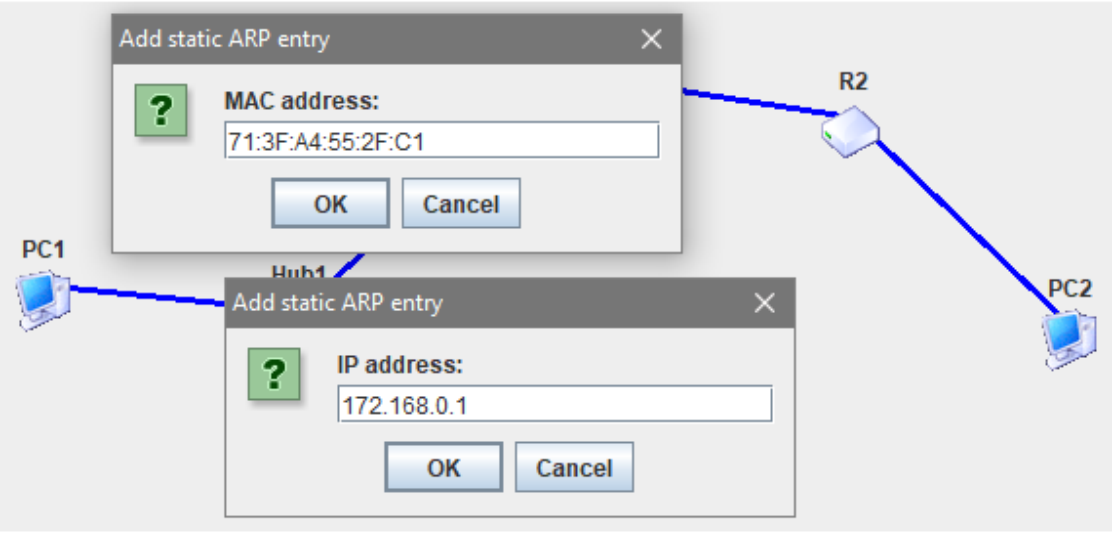


Рис. 7. Добавление статических ARP записей из графической средв.

Установив статические ARP записи на всех узлах, через которые проходит эхо-запрос, можно убедиться в полном отсутствии на этом пути использования пакетов ARP протокола.

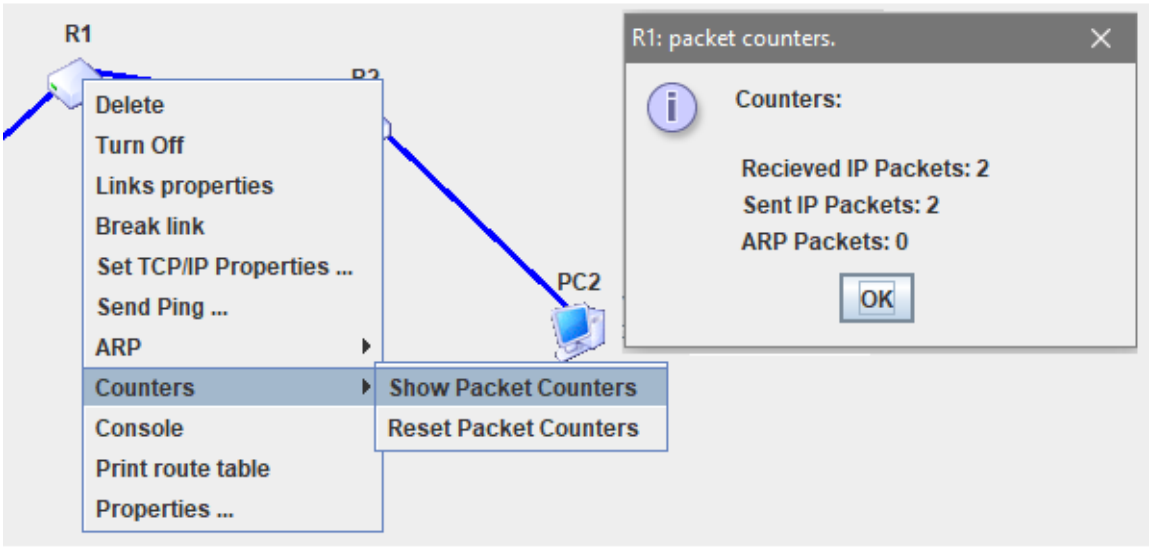


Рис. 8. Просмотр статистики работы узла сети.

5. Выполнение эхо-запроса для несуществующего узла сети с IP-адресом 192.168.0.1.

   Для этого выполним на PC1 последовательность действий, аналогичную предыдущему пункту, вместо адреса 10.0.0.2 используя адрес 192.168.0.1:

   > PC1 Created Echo Request packet to 192.168.0.1
   >
   > PC1 Sending packet from ProtocolStack (to 172.168.0.1).
   >
   > ...
   >
   > R1 ProtocolStack received packet from local Interface.
   >
   > R1 Packet Received: Network Layer Device is Routable forwarding packet.
   >
   > R1 Forwarding packet from ProtocolStack (to 172.168.100.1).
   >
   > R2 ProtocolStack received packet from local Interface.
   >
   > R2 Packet Dropped: Hop count exceeded.
   >
   > Host 192.168.0.2 Unreachable

Как видно, пакет попал в "петлю" между двумя маршрутизаторами и находился там, пока у него не закончилось время жизни (TTL).

6. Запустим на PC1 SNMP агент с параметрами:
   - Порт на котором SNMP агент будет ожидать пакеты: 161.
   - Имя группы доступа для SNMP агента: defgroup.
7. Выполним с PC2 запрос SNMP-агенту на PC1 со следующими параметрами:
   - IP адрес компьютера на котором установлен SNMP агент: 172.168.0.2.
   - порт на котором SNMP агент ожидает пакеты: 161;
   - SNMP запрос: get.
   - SNMP переменные: ip.address_eth0; device.hostname.
   - Имя группы доступа: defgroup (или сокращенно, def).

Результаты запроса будут выведены в консоль:

Console | Node Information

...	Node	Packet	Info
...	PC2	SNMP	Sending getRequest message `00 0F 02 02 00 00 05 01 00 03 64 65 66 02 00 03 00` to 172.168.0.2:161
...	PC2	UDP	Created UDP packet for 172.168.0.2:161.
...	PC1	UDP	UDP packet received from 10.0.0.2:3000 message: "□□□□□□def□□". UDP Port 161 has status "busy" from now.
...	PC1	SNMP	Received getRequest: `00 0F 02 02 00 00 05 01 00 03 64 65 66 02 00 03 00`
...	PC1	SNMP	Sending getResponse message `00 17 02 02 00 00 05 05 00 00 00 00 00 00 00 02 00 03 00 02 00 03 50 43 31` to manager
...	PC1	UDP	Created UDP packet for 10.0.0.2:3000.
...	PC2	UDP	UDP packet received from 172.168.0.2:161 message: "□□□□□□□□□PC1".
...	PC2	SNMP	Received getResponse: 'Device.Hostname=PC1'
...	PC1	SNMP	SNMP agent closed connection.
...	PC1	UDP Ap	Application is now listening on port 161.
...	PC1	SNMP	SNMP agent listen on port 161

Рис. 9. Результат работы запроса к SNMP-агент.

Из анализа содержимого консоли следует, что:

- Протокол SNMP в качестве транспортного использует UDP протокол.
- В UDP-дейтаграмму вкладывается SNMP-DPI сообщение, структура которого для текущего Get запроса имеет вид, представленный строкой шестналцатиричных кодов в первой строке консоли.
- В соответствии с параграфом 4.3.2 данное SNMP-DPI сообщение имеет следующий формат:

0 1	2	3	4	5 6	7	8 9	10 11 12	13 14	15 16
00 0F	02	02	00	00 05	01	00 03	64 65 66	02 00	03 00
A	B	C	D	E	F	G	H	I	J
15	2	2		5		3	d e f	Device	Hostname

Рис. 10. Формат get запроса SNMP-агенту на PC1.

- Откуда следует, что это get-запрос (F=1) версии SNMPv2 (B=2) длиной 17 байт (A=15) с уникальным идентификатором сообщения равным 5 (E=5), который будет увеличиваться на 1 при каждом следующем запросе.
- Длина поля "Имя группы доступа (H)" равна трем байтам (G=3) и имеет значение H="def". При этом у SNMP-агента запрашивается значение параметра объекта J=3 из группы I=2. Это переменная Device.Hostname.
- В ответ на этот запрос SNMP-агент, используя UDP транспорт, вернет значение запрашиваемой переменной, а именно – 'Device.Hostname=PC1'.

Аналогично можно получить значения сразу двух переменных. Для этого надо повторить запрос, изменив в нем строку переменных, введя через точку с запятой

имена переменных: ip.address_eth0; device.hostname. Результат запроса будет представлен в консоли в виде:

```
PC2  Received getResponse:'IP.Address_Eth0=172.168.0.2',
'Device.Hostname=PC1'
```

8. Запустим Telnet-сервер со следующими параметрами:
   - Порт, на котором Telnet-сервер будет ожидать пакеты: 23.
   - Пароль для доступа к Telnet: 234.
9. Запустим Telnet-клиент с параметрами:
   - IP адрес Telnet сервера: 10.0.0.2.
   - Порт, на котором Telnet-сервер ожидает пакеты: 23.

В ответ на приглашение к авторизации в системе необходимо ввести имя пользователя root и пароль 234. После входа в систему будет выведено приглашение командной строки:

```
pc1 #
```

Просмотрим записи в таблице маршрутизации:

```
pc1 # arp -a
Internet Address Physical Address Type
10.0.0.1 A2:2A:55:20:75:42 Dynamic
```

Как видно из вывода команды arp, в кэше находится лишь одна динамическая запись. Ее можно удалить следующим образом:

```
pc1 # arp -d 10.0.0.1
```

Для добавления статической записи в кэш ARP необходимо использовать ключ -s команды arp:

```
pc1 # arp -s 10.0.0.1 A2:2A:55:20:75:42
```

Таким образом была добавлена статическая запись для компьютера PC1. После завершения работы закрываем сеанс Telnet.