

ИМИТАТОР javaNetSim

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.

Имитатор javaNetSim является объектно-ориентированным и написан на языке Java. Программы написанные на этом языке являются машиннонезависимыми, т.е. имитатор javaNetSim будет работать на любом компьютере, для которого есть виртуальная Java машина. Хотя язык Java является интерпретируемым, это не оказывает существенного

влияния на быстродействие имитатора. Это объясняется тем, что имитатор разрабатывался для моделирования работы небольших сетей, обработка моделей которых не требует больших вычислительных ресурсов.

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. в нем 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). Рабочая область является частью основного окна программы, представленного на рисунке 1. Основное окно программы логически разделено на четыре части:

- рабочая область, обозначенная цифрой 1 – содержит сетевые устройства и линии связи между ними:
 - концентратор на пять сетевых интерфейсов – обозначен числом 13;

- маршрутизатор соединяющий две подсети – обозначен числом 14;
- компьютер (конечный узел сети) – обозначен числом 15;
- Линия связи между двумя сетевыми устройствами – обозначен числом 16.
- область вывода результатов, обозначенная цифрой 2 – содержит две вкладки:
 - вкладка "консоль", обозначенная числом 11 – содержит журнал передачи пакетов по сети
 - вкладка "информация об устройствах", обозначенная числом 12 – для каждого интерфейса всех сетевых устройств содержит IP адрес, маску подсети и шлюз по умолчанию.
- главное меню, обозначенное цифрой 3 – содержит основные действия по управлению имитатором;
- линейка инструментов – содержит следующие кнопки:
 - кнопка "создать пустую конфигурацию" – обозначена цифрой 4;
 - кнопка "открыть существующую конфигурацию" – обозначена цифрой 5;
 - кнопка "сохранить текущую конфигурацию" – обозначена цифрой 6;
 - кнопка "создать компьютер" – обозначена цифрой 7;
 - кнопка "создать маршрутизатор" – обозначена цифрой 8;
 - кнопка "создать концентратор" – обозначена цифрой 9;
 - кнопка "создать соединение" – обозначена числом 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.

1. 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.

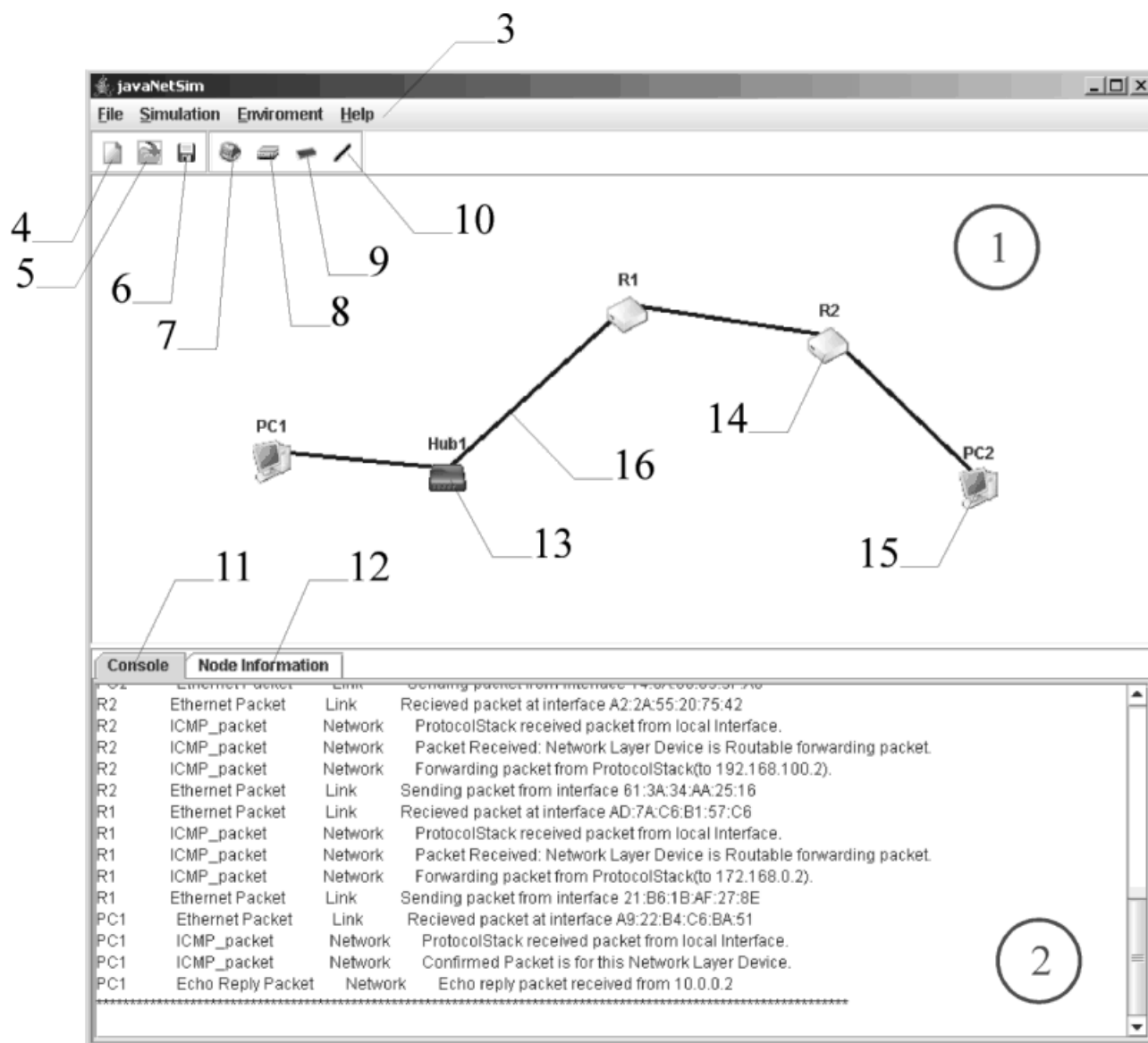


Рис. 1 Основное окно программы

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". В этом окне (рисунок 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.

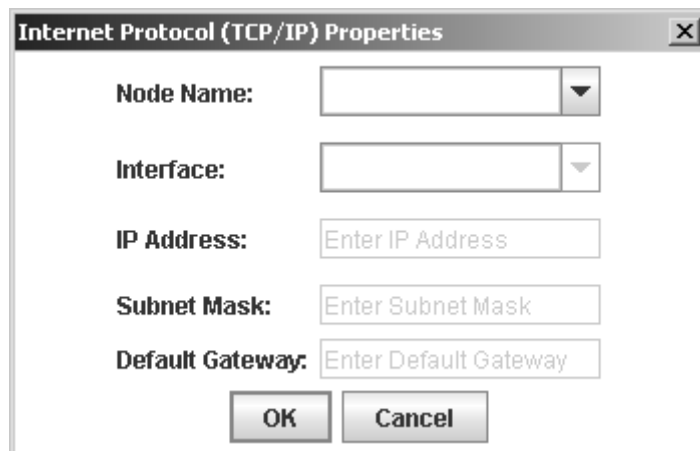


Рис. 2 Установка параметров TCP/IP

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.

2. Контекстное меню

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

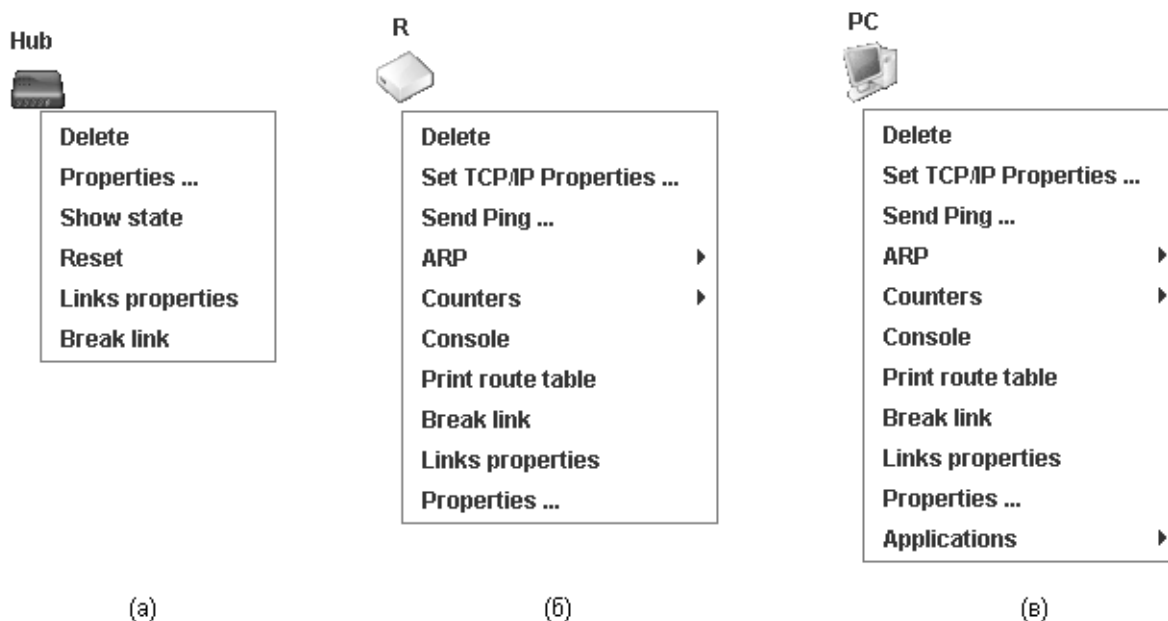


Рис. 3 Контекстное меню

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 пропуска

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 ещё 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 появиться сообщения, аналогичные приведенным ниже:

```
PC1EchoRequestPacketNetworkCreatedEcho
```

```
Requestpacketto10.0.0.2
```

```
...
```

```
PC1EchoReplyPacketNetworkEchoReplypacket
```

Меню ARP позволяет управлять таблицей протокола ARP на выбранном устройстве и 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 заносится 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, добавляется еще один пункт: подменю Applications(Приложения), которое позволяет работать с протоколами: Echo(UDP,TCP), SNMP и TELNET.

3. Командная строка

Для запуска командной строки из контекстного меню выберем "Console", появится окно консоли (рисунок 4). Окно разделено на 2 части:

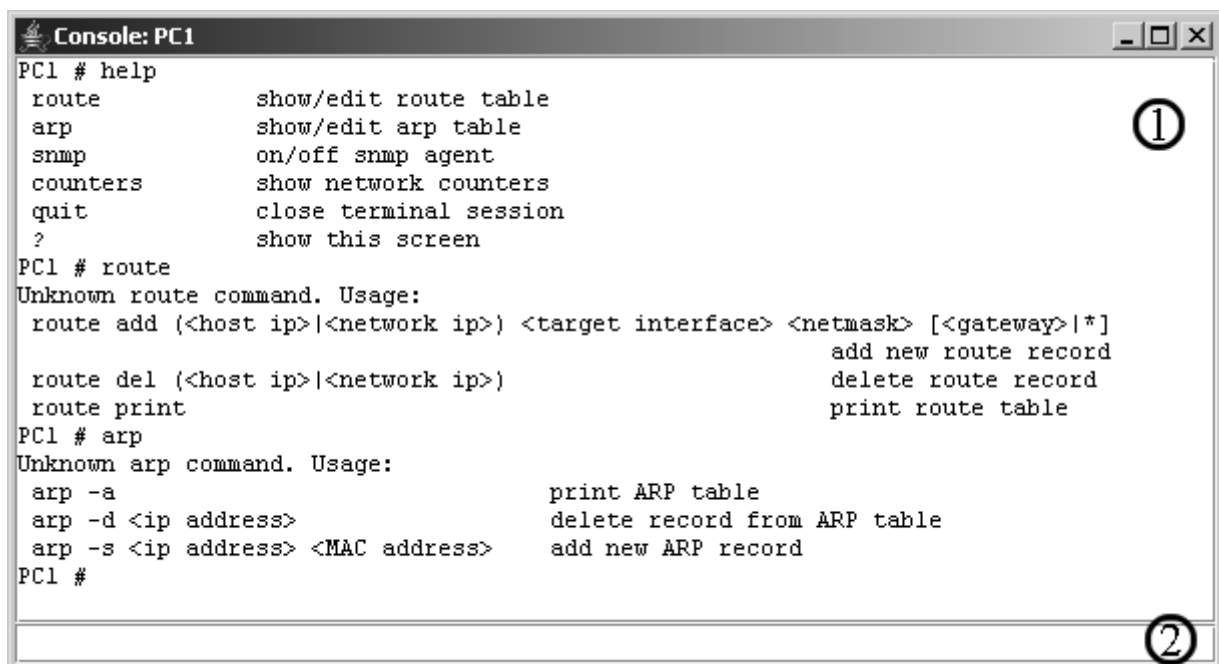


Рис. 4 Консоль

Цифры на рисунке 4 обозначают следующее:

1 – область для сохранения результата выполнения команд;

2 – командная строка, в которой можно вводить команды на выполнение.

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.

4. 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.

4.1. Работа с протоколом Echo

Имитаторе *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 или "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. После этого с любого другого узла можно отсылать сообщения на тот узел, на котором запущен эхо-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 или "Applications > Send data via tcp echo client" для Echo-TCP и ввести четыре параметра:

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

4.2. 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 рисунке 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.

The image shows a Windows-style dialog box titled "SNMP Send Data". It has a close button (X) in the top right corner. Inside the dialog, there are five labeled input fields arranged vertically: "IP Address:", "Destination port:", "SNMP message", "Variables", and "Community name". Each label is followed by a rectangular text input box. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

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 ReceiivedttResponse:
        'IP.Address_Eth0=172.168.0.2' , 'Dev c .Hos name=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.ARPTTable – 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.

4.3. 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 и 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.

ГЛОССАРИЙ

API Application Programming Interface. Набор функций, используемых прикладным программистом, оформленных в виде библиотеки.

ARP Address Resolution Protocol. Протокол разрешения трансляции MAC-адресов на IP-адреса.

Ethernet Пакетная технология вычислительных сетей.

ICMP Internet Control Message Protocol. Протокол передачи команд и сообщений об ошибках.

IP Internet Protocol. Основной протокол стека TCP/IP, функционирующий на уровне межсетевого взаимодействия.

IP-сокет Номер порта в совокупности с номером сети и номером конечного узла.

javaNetSim Программный эмулятор работы сети, позволяющий работать с сетью на всех четырех уровнях модели стека протоколов TCP/IP.

MIB (Management Information Base) База данных информации для управления сетью, в которой определен стандартный набор информации, дающий возможность SNMP менеджеру собирать информацию о сетевом устройстве.

NIC Network Information Center. Информационный центр Internet, занимающийся распределением доменных имен первого уровня.

OSI Open System Interconnection. Стандарт открытого взаимодействия систем.

RARP Reverse Address Resolution Protocol. Протокол обратного разрешения MAC-адресов.

RFC Request For Comments. Формат документов для публикации стандартов по технологиям Internet.

SNMP (Simple Network Management Protocol) Простой протокол управления сетью, предназначенный для наблюдения и управления сетевыми устройствами работающими на уровне приложений.

SNMP агент Программное обеспечение сетевых устройств, которое на каждый запрос SNMP менеджера формирует отклик, содержащий результат обработки запроса.

SNMP менеджер Приложение управляющее сетью, посредством отправки запросов SNMP агенту.

TCP Transmission Control Protocol. Протокол стека TCP/IP, осуществляющий функции контроля за передачей данных. Функционирует на основном уровне.

TCP-Сегмент Часть информации, поступающей по протоколу TCP в рамках соединения от протоколов более высокого уровня, сформированная для передачи на сетевой уровень.

TELNET Сетевой протокол типа "клиент-сервер" для обеспечения незащищенного удаленного доступа к сетевому устройству с помощью командного интерпретатора.

TFTP Trivial File Transfer Protocol. Простой протокол передачи файлов.

TTL Time-To-Live. Время жизни IP-пакета.

UDP User Datagram Protocol. Протокол передачи пакетов пользователя.

X.25 Стандарт, описывающий канальный, сетевой и физический уровни OSI.

Демультимплексирование Процедура распределения протоколом поступающих от сетевого уровня пакетов между набором высокоуровневых служб, идентифицированных номерами портов.

Коммутатор (switch) Устройство, предназначенное для соединения нескольких сегментов компьютерной сети.

Концентратор (hub) Устройство для объединения нескольких устройств Ethernet в общий сегмент.

Маршрут Последовательность промежуточных узлов, которые проходит IP-пакет в процессе маршрутизации при движении от отправителя к месту назначения.

Маршрутизация Процесс определения маршрута следования пакетов данных в вычислительных сетях.

Маршрутизация динамическая Метод маршрутизации осуществляемый с помощью протоколов маршрутизации.

Маршрутизация статическая Метод маршрутизации осуществляемый на основе таблиц маршрутизации.

Маршрутизатор Сетевое устройство, используемое в компьютерных сетях передачи данных, которое, на основании информации о топологии сети (таблицы маршрутизации) и определенных правил, принимает решения о пересылке пакетов сетевого уровня их получателю.

Мультиплексирование Процедура обслуживания протоколом запросов, поступающих от нескольких различных прикладных служб.

Объект MIB Порция информации, существующая в базе данных MIB.

Порт Очередь, организуемая операционной системой, к точке входа прикладного процесса.

Протокол Совокупность синтаксических правил, определяющая взаимодействие двух узлов вычислительной сети.

Сетевой уровень (internetworking layer) Уровень модели TCP/IP, предназначенный для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной топологией.

Сообщение (message) Единица данных при взаимодействии клиента и сервера посредством протокола прикладного уровня.

Таблица маршрутизации Специальная информационная структура, используемая для определения маршрута следования пакета по адресу его сети назначения.

Уровень приложений Набор сетевых служб, которые предоставляет система сетевым пользовательским приложениям. Уровень приложений обеспечивает набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, преобразование структур данных для пересылки их по сети и поддержку сеансов связи.

Управление сетью Процесс управления отказами, контроля конфигураций, мониторинга производительности, обеспечения защиты и учета деятельности в сети передачи данных.