# Моделирование работы сети в Net-Simulator

**Составитель: Коробецкая А.А.**

## Подготовка к работе. Установка NET-Simulator

NET-Simulator – это бесплатно распространяемая программа, позволяющая имитировать работу компьютерных сетей.

Скачать программу можно с официального сайта: http://www.net- simulator.org

На том же сайте находится инструкция по установке, справка и описание примера сети.

Для работы программы необходимо установить Java-машину с официального сайта: http://java.com/ru/ (для Windows Vista, 7, 8).

Для Windows XP может потребоваться более старая Java SE 7: http://java.com/ru/download/manual\_java7.jsp

Для запуска NET-Simulator просто разархивируйте архив и запустите файл run.bat.

Внимание! Путь к папке с NET-Simulator не должен содержать русских символов!

Если все выполнено правильно, сначала запустится командная строка, а затем откроется окно с примером сети.



На практике используются более сложные симуляторы сетей, в которых доступно множество реальных устройств. Примеры симуляторов:

* ns-3 (бесплатная);
* NetSim (проприетарная);
* HP Network Simulator (бесплатная);
* Cisco CCNA Labs Simulation (проприетарная).

## Задание

Ознакомьтесь с интерфейсом NET-Simulator, используя документацию с сайта и пример сети, открывающийся при запуске NET-Simulator. Найдите ответы на вопросы (устно).

1. Какие сетевые устройства можно использовать в NET-Simulator?
2. Как добавить и удалить устройства в проект?
3. Как подсоединить кабель к устройству?
4. Как запустить терминал для настройки устройств?
5. Какие команды поддерживает терминал NET-Simulator?

Реализуйте следующие примеры сетей.

1. сеть «точка-точка»;
2. сеть с топологией «шина» на общем концентраторе;
3. сеть с топологией «пассивная звезда» с использованием коммутатора;
4. соединение сетей вручную через коммутатор;
5. соединение различных сетей через роутер.

Каждую сеть необходимо сохранить в виде отдельного проекта NET-Simulator и сгенерировать по ней отчет в HTML.

Изучите полученные отчеты.

При сдаче работы необходимо продемонстрировать работоспособность каждой сети. Если работа сдается по почте, сделайте скриншоты для каждой сети, как в методичке.

**Контрольные вопросы** по разработанным проектам сетей:

1. Какие устройства использовались в проекте и каково их назначение?
2. Какой адрес имеет то или иное устройство?
3. Какая топология сети в реализована данном проекте? Есть ли подсети?

## Указания к выполнению работы

В данных указаниях не дублируется информация с сайта NET-Simulator. Самостоятельно ознакомьтесь и при необходимости обращайтесь к документации программы.

Каждая сеть сохраняется в отдельном проекте.

### Сеть «точка-точка» (point-to-point)

«Точка-точка» – простейшая сеть, состоящая из 2-х рабочих станций, соединенных кабелем.

Создайте новый проект.

Разместите на листе 2 компьютера и соедините их кабелем. При правильном подключении на компьютерах загорятся 2 зеленые лампочки.



Дважды кликните по компьютеру “Desktop 0”. Откроется окно терминала.

Введите help чтобы увидеть список доступных команд.



Команда ifconfig позволят посмотреть и настроить параметры сетевых интерфейсов (сетевых карт, разъемов роутера и т.п.).



Пока мы не настраивали свою сеть, сетевые карты компьютеров отключены и не имеют своего адреса. Чтобы это увидеть, введите команду ifconfig с параметром -a:



eth0 – это имя интерфейса (в реальности может быть произвольным);

Link encap: Ethernet – используемый стандарт подключения;

HWaddr – физический адрес (MAC-адрес), неизменяемый;

down – состояние (выкл.);

далее идет статистика передачи данных.

Назначим первому компьютеру IP-адрес 192.168.1.1 с маской 255.255.255.0 (адреса 192.168.x.x/24 стандартно используются для небольших локальных сетей) и посмотрим, что изменилось:



В описание интерфейса добавилась строчка с настройками IP-адреса и статус сменился с down на up.

Аналогично настроим второй компьютер на адрес 192.168.1.2/24 (адрес должен быть из той же сети, например, 192.168.20.2 не подойдет, а 192.168.1.100 – подойдет).



Теперь проверим работоспособность сети с помощью команды ping (Ctrl+C – остановить передачу, всего необходимо отправить 7-10 пробных пакетов). Обратите внимание, пока ведется передача/получение данных, у узлов мигают зеленые лампочки, а кабель подсвечивается синим.





Таким образом, в ходе передачи ни один пакет не был потерян. Мы получили работоспособную сеть «точка-точка».

Сохраните результат в виде отдельного проекта.

Проверьте себя:

Что такое маска сети?

Какой адрес у созданной в примере сети (net id)?

Каковы адреса узлов (host id) в сети?

### Сеть на основе концентратора. Топология «шина» (пассивная звезда)

Мы будем совершенствовать ранее созданную сеть «точка-точка», но сохранить ее нужно в отдельном проекте.

Предположим, мы хотим создать сеть из трех компьютеров. Напрямую соединить их уже не получится, т.к. у каждого компьютера только один интерфейс (сетевая карта).

Даже если в реальной сети у компьютера две сетевых карты, настроить общую сеть без сетевых устройств довольно затруднительно – один из компьютеров придется сделать сервером.

В примере мы реализуем простейшую сеть на основе концентратора. Ее можно рассматривать и как «шину», и как пассивную звезду. Реальную шину с одним общим кабелем в Net-Simulator создать нельзя, т.к. на кабель подключается ровно 2 устройства.

Итак, добавьте к ранее созданной сети «точка-точка» еще один компьютер, концентратор (hub) и соедините кабелем, как показано на рисунке (расположение узлов на листе может быть любым):



Адрес сети мы оставим прежним, поэтому первые два узла перенастраивать не нужно. Они по-прежнему будут работоспособны. Проверьте это с помощью команды ping.

Совет

Для повтора предыдущей команды терминала нажмите на клавиатуре стрелку «вверх».



Необходимо настроить только третий узел, выдав ему адрес из той же сети, например 192.168.1.3:



Сам концентратор не является активным устройством и не настраивается, т.е. не имеет IP-адреса.

Проверим доступность нового компьютера:





Первый пакет с нового компьютера был потерян (возможно, неполадки в сети), затем передача шла без сбоев.

Обратите внимание, при передаче данных мигают лампочки у всех компьютеров и все кабели, т.е. запрос ping отправляется не только тому, кому он предназначен, но и всем остальным устройствам в сети. Поэтому сеть с такой топологией будет весьма загруженной.



Сохраните полученную сеть в отдельный проект.

Аналогично можно добавить четвертый, пятый и т.д. узел. Если число узлов больше числа разъемов концентратора, то можно использовать несколько концентраторов, или даже выделить свой концентратор каждому компьютеру, чтобы сделать сеть похожей на «шину».

Настройка сети будет во всех случаях одинаковой. И в любом случае, сеть можно считать реализованной по топологии «шина».

Примеры (реализовывать не нужно).





Проверьте себя:

1. Какой адрес у концентратора?
2. Сколько узлов может быть в сети с топологией «шина» (в реальной и в модельной)?

### Сети с использованием коммутатора. Пассивная звезда

Чтобы снизить загрузку сети, вместо концентратора можно использовать коммутатор (switch). Это устройство может анализировать физический адрес и передает пакеты не всем узлам, а только конкретному получателю.

Такая сеть имеет топологию «пассивная звезда»: находящийся в центре коммутатор не управляет сетью, но передача идет не на все компьютеры, как в «шине», а только на нужные.

Для этого у коммутатора есть таблица физических адресов (**mactab**), в которой записано, к какому интерфейсу какой узел подключен.

Эта таблица заполняется автоматически. При попытке передачи коммутатор сначала опрашивает все подключенные устройства и узнает их адреса. Адреса заносятся в таблицу, и далее коммутатор ведет передачу только по нужному адресу через нужный интерфейс.

Поскольку устройства могут подключаться и отключаться, MAC-таблица периодически очищается и коммутатор опрашивает устройства снова. Это позволяет поддерживать таблицу в актуальном состоянии.

Для реализации такой сети просто замените концентратор в предыдущем проекте на коммутатор. Перенастраивать компьютеры не нужно.

Внимание! В NET-Simulator имеется баг. Если удалить какое-то устройство, то кабели, которые были к нему подключены, уже никуда не удастся подсоединить. Прежде, чем удалять концентратор, отключите от него все кабели. Если баг уже возник, кабели тоже придется удалить и подключить новые.



Теперь, если проверить работоспособность сети, сначала произойдет отправка на все хосты, а потом коммутатор будет отправлять данные только на нужный (со 192.168.1.1 на 192.168.1.2):



Не прекращая передачу, проверьте MAC-таблицу коммутатора:



Одновременно, не останавливая предыдущий ping, запустите еще один – с узла 192.168.1.2 на 192.168.1.3 и еще раз проверьте MAC-таблицу:



После остановки передачи, через несколько секунд таблица очистится.



Сохраните полученную сеть в отдельный проект.

Проверьте себя:

В чем заключается отличие концентратора от коммутатора?

### Две сети на общем коммутаторе

Мы можем подключить две различных компьютерных сети к одному общему коммутатору, как если бы это были отдельные сети, не связанные между собой.

Добавьте в предыдущий проект еще два компьютера. Подсоедините новые компьютеры к свободным разъемам коммутатора.



По аналогии с предыдущим заданием, назначьте новым компьютерам адреса 192.168.100.1 и 192.168.100.2 с маской 255.255.255.240.





Таким образом, у нас получилось две подсети:

1. сеть 192.168.1.0 с маской 255.255.255.0, компьютеры Desktop 0, Desktop 1, Desktop 2;
2. сеть 192.168.100.0 с маской 255.255.255.240, компьютеры Desktop 4, Desktop 5.

Если мы проверим работу сети, то увидим, что внутри каждой из подсетей пакеты циркулируют свободно, но не могут попасть из одной подсети в другую, хоть эти сети и подключены к одному и тому же устройству.



Причина в том, что у компьютеров не настроены таблицы маршрутизации, т.е. компьютеры не знают, каким образом передать данные в другую сеть. Они просто не могут начать передачу на неизвестный адрес из неизвестной сети.

Когда мы назначали ip-адреса компьютерам, в их **таблицы маршрутизации** автоматически добавлялась одна единственная строчка: с собственной сетью компьютера.

Просмотреть и настроить таблицу маршрутизации можно командой route. У компьютеров первой подсети она выглядит так:



А во второй подсети так:



Destination – адрес *назначения*, для которого адреса задается маршрут в данной строке;

Gateway – на какой *шлюз* отправлять пакеты, \* - ни на какой, передача внутри локальной подсети;

Flags – *флаги* (проставляются автоматически): U – маршрут активен, G – маршрут использует шлюз, H – адрес назначения является адресом отдельного хоста, а не сети;

Metric – метрика, определяет приоритет маршрутов: чем метрика больше, тем маршрут хуже;

Iface – интерфейс, через который ведется передача.

Т.е., у компьютера Desktop 1 нет в таблице маршрутизации ни адреса сети 192.168.100.0, ни адреса узла 192.168.100.1 (Desktop 4). Когда мы пытаемся запустить передачу к узлу Desktop 4, исходный узел Desktop 1 просматривает свою таблицу маршрутизации, не находит там никакой информации про Desktop 4 и не может начать передачу, т.к. не знает как.

Таким образом, компьютеры первой подсети «знают» только свои локальные адреса из диапазона 192.168.1.0, а второй подсети – только из 192.168.100.0.

Чтобы связать сети между собой, необходимо добавить их в таблицы маршрутизации каждого компьютера.

Управление таблицей маршрутизации осуществляется той же командой route -add *адрес\_назначения* -netmask *маска* -dev *интерфейс*

Каждому компьютеру первой подсети (Desktop 0, Desktop 1, Desktop 2) необходимо добавить сведения о компьютерах второй подсети:



Аналогично компьюетрам второй подсети (Desktop 4, Desktop 5) добавляем сведения про первую подсеть:



Совет

Если вы ошиблись при добавлении маршрута, вам нужно сначала удалить из таблицы неправильный маршрут, а потом добавить правильный:

1. Стрелкой «вверх» пролистайте команды до той, в которой вы ошиблись.
2. Замените -add на -del и выполните команду.
3. Еще раз пролистайте команды и исправьте ошибку.

Теперь (только после настройки обеих подсетей!) они могут передавать пакеты друг другу.



Таким образом, мы получили две подсети, подключенных к общему маршрутизатору и соединенные через него.

Сохраните полученную сеть в отдельный проект.

Проверьте себя:

1. Сколько максимум компьютеров можно подключить к сети с маской 255.255.255.240?

2. Есть ли таблица маршрутизации у коммутатора?

### Соединение различных сетей через маршрутизатор

В предыдущем примере мы соединили две маленьких сети с помощью одного коммутатора. Но для больших сетей, в которых много узлов и подсетей, этот вариант не подходит, потому что:

а) MAC-таблица коммутатора становится очень большой, что требует дополнительной памяти и замедляет его работу;

б) для обновления таблицы коммутатор запрашивает физические адреса всех устройств сети, а это большой дополнительный трафик;

в) каждому компьютеру придется вписать в таблицу маршрутизации адреса всех подсетей.

В реальности для объединения сетей обычно используется **маршрутизатор (роутер)**. Он распределяет трафик между подсетями и определяет путь доставки каждого пакета.

С маршрутизатором каждому компьютеру не нужно знать адреса всех сетей, нужно только знать адрес своего бижайшего роутера, который уже решит, куда отправлять пакет.

Коммутаторы и концентраторы в таких сетях тоже используются, но они функционируют в пределах подсети. Создаваемый ими трафик не выходит дальше ближайшего роутера.

Откройте проект, в котором мы создали сеть «пассивная звезда» на коммутаторе, но еще не добавили вторую подсеть.

Добавьте в проект маршрутизатор, еще один концентратор, два компьютера и необходимые кабели, и соедините их как показано на рисунке.



Задайте для двух новых компьютеров адреса 172.16.0.101 и 172.16.0.110 с маской 255.255.0.0. Проверьте работоспособность каждой из подсетей.





Теперь настроим маршрутизатор.

В отличие от компьютеров, маршрутизатор имеет 8 портов, каждому из которых соответствует свой интерфейс (eth0-eth7) и свой ip-адрес.

Мы подключили первую подсеть (192.168.1.0/24) к интерфейсу eth0, а вторую (172.16.0.0/16) – к интерфейсу eth7. Этим интерфейсам нужно выдать адреса из диапазона соответствующей сети, например 192.168.1.100 и 172.16.1.100.

Примечание

В реальных сетях традиционно маршрутизатор получает адрес с последним байтом равным 1 (например, 10.0.0.1), а обычные узлы – начиная со 100 (например, 10.0.0.101, 10.0.0.102 и т.д.). Но это необязательно, поэтому мы сделали наоборот.





Осталось сообщить всем компьютерам адрес их маршрутизатора (внести его в таблицу маршрутизации).

Нам нужно указать, что пакеты для всех адресов, кроме локальных, должны отправляться на маршрутизатор.

«Все адреса» обозначаются в таблице маршрутизации как назначение 0.0.0.0 с маской 0.0.0.0 – «шлюз по умолчанию».

Настройка Desktop 0 (аналогично настраиваются Desktop 1 и Desktop 2):



Настройка Desktop 6 (аналогично настраивается Desktop 7):



Проверка доступности соседней подсети:





Для того, чтобы узнать физический MAC-адрес соседних устройств, маршрутизатор использует запросы по протоколу ARP. Во время передачи можно просмотреть его ARP-таблицу (потом она очищается):



Примечание

У реальных устройств обычно нет команды, аналогичной arp. Она добавлена в Net-Simulator для наглядности.

Полностью настройки сети можно просмотреть через html-отчет (см. пример ниже).

Сохраните проект в отдельном файле и сгенерируйте для него отчет. Отчет для полученной сети представлен ниже:

Проверьте себя

Какие устройства придется добавить и какие настроить, чтобы подсоединить к маршрутизатору еще одну подсеть с адресом 10.10.10.0/24 и тремя узлами?

**NET-SIMULATOR PROJECT REPORT**

**Project file:**

**Author:**

**Description:**

**Project created at:**

**Report generated at**: 2014-10-2 0:56:6



**Desktop**

|  |  |
| --- | --- |
| Name: | Desktop 0 |
| Description: |  |