

Тема 17. Команды ОС Unix конфигурирования и тестирования сетевых интерфейсов

Настройка сетевых интерфейсов

Интерфейсом с точки зрения ОС является устройство, через которое система получает и передает IP-пакеты. Роль интерфейса локальной сети может выполнять одно (или несколько) из следующих устройств: Ethernet-карта, ISDN-адаптер или модем, подключенный к последовательному порту. Каждое устройство (не весь компьютер!) имеет свой IP-адрес. Для выхода в локальные сети используется, как правило, Ethernet-карта, что и будет предполагаться в настоящем разделе.

Расположение конфигурационных файлов

Отметим сразу, что все приводимые ниже команды можно выполнять из командной строки, но тогда придется повторять эти операции при каждом перезапуске компьютера. Поэтому может быть удобнее записать их в один из инициализационных файлов, автоматически запускаемых при старте системы. В разных дистрибутивах процесс загрузки организован по-разному. В "Linux NET-3-HOWTO" приводится следующая таблица:

Таблица 10 – Расположение конфигурационных файлов в основных дистрибутивах

Дистрибутив	Настройка интерфейса и маршрутизации	Запуск демонов
Debian	/etc/init.d/network	/etc/init.d/netbase /etc/init.d/netstd_init /etc/init.d/netstd_nfs /etc/init.d/netstd_misc
Slackware	/etc/rc.d/rc.inet1	/etc/rc.d/rc.inet2
RedHat	/etc/sysconfig/network-scripts/ifup-<ifname>	/etc/rc.d/init.d/network

Обратите внимание, что дистрибутивы Debian и Red Hat содержат отдельный каталог для скриптов запуска системных сервисов (хотя сами файлы настроек находятся в других местах, например, в дистрибутиве Red Hat они хранятся в каталоге /etc/sysconfig). Для понимания процесса загрузки ознакомьтесь с содержимым файла /etc/inittab и документацией по процессу init.

Команда ifconfig

После подключения драйверов вы должны настроить те интерфейсы, которые вы предполагаете использовать. Настройка интерфейса заключается в присвоении IP-адресов сетевому устройству и установке нужных значений для других параметров сетевого подключения. Наиболее часто для этого используется программа `ifconfig` (ее название происходит от "interface configuration").

Запустите ее без аргументов (или с единственным аргументом `-a`) и вы узнаете, какие параметры установлены в данный момент для активных сетевых интерфейсов (в частности, для сетевой карты). Кстати, имеет смысл выполнить эту команду еще до подключения модулей: а вдруг у вас поддержка интерфейсов встроена в ядро и необходимые настройки сделаны в процессе инсталляции системы. Тогда вы в ответ можете получить информацию о параметрах вашей Ethernet-карты и так называемого "кольцевого интерфейса" или "обратной петли" - Local Loopback (интерфейс Ethernet при единственной сетевой карте обозначается как `eth0`, а кольцевой интерфейс - как `lo`). Если же по этой команде вы ничего не получите, то надо переходить к подключению модулей и настройке, и начинать надо с кольцевого интерфейса.

Настройка локального интерфейса lo

Этот интерфейс используется для связи программ IP-клиентов с IP-серверами, запущенными на той же машине, так что его необходимо настроить даже в том случае, если вы вообще не подключаете никаких сетевых устройств.

Локальный интерфейс настраивается очень просто: командой

```
[root]# /sbin/ifconfig lo 127.0.0.1
```

Теперь, чтобы проверить работоспособность протоколов TCP/IP на вашей машине, дайте команду:

```
[root]# ping 127.0.0.1
```

Настройка интерфейса платы Ethernet локальной сети (eth0)

Для того чтобы ваш компьютер вошел в сеть с IP-адресом, полученным вами у администратора (пусть для примера это будет адрес 192.168.0.15), вы должны запустить команду `ifconfig` примерно следующим образом:

```
[root]# /sbin/ifconfig eth0 192.168.0.15 netmask  
255.255.255.0 up
```

Если не указывать маску подсети, то по умолчанию устанавливается маска подсети 255.0.0.0.

В некоторых случаях необходимо бывает изменить адрес прерывания, используемого сетевой картой, порта ввода-вывода или типа соединения, используемого в сети. Это можно сделать, выполнив следующую команду:

```
root# /sbin/ifconfig eth0 irq 5 io_addr 220 media
10baseT
```

Не все устройства (платы) поддерживают динамическое изменение этих параметров (т. е. может потребоваться переустановить переключатели на плате).

Интерфейс для последовательного порта

Последовательный порт используется для подключения модема, через который осуществляется соединение с сетью по телефонной линии. Для настройки интерфейса этого типа тоже можно использовать программу ifconfig. Однако, такие программы как rppd и dip, используемые для соединения с сетью по модему, способны автоматически конфигурировать сетевой интерфейс, поэтому обычно для этого случая применять ifconfig не требуется.

Настройка маршрутизации

Правила маршрутизации определяют, куда отправлять IP-пакеты. Данные маршрутизации хранятся в одной из таблиц ядра. Вести таблицы маршрутизации можно статически или динамически. Статический маршрут - это маршрут, который задается явно с помощью команды route. Динамическая маршрутизация выполняется процессом-демоном (routed или gated), который ведет и модифицирует таблицу маршрутизации на основе сообщений от других компьютеров сети. Для выполнения динамической маршрутизации разработаны специальные протоколы: RIP, OSPF, IGRP, EGP, BGP и т. д.

Динамическая маршрутизация необходима в том случае, если у вас сложная, постоянно меняющаяся структура сети и одна и та же машина может быть доступна по различным интерфейсам (например, через разные Ethernet или SLIP интерфейсы). Маршруты, заданные статически, обычно не меняются, даже если используется динамическая маршрутизация.

Для персонального компьютера, подключаемого к локальной сети, в большинстве ситуаций бывает достаточно статической маршрутизации командой route. Прежде чем пытаться настраивать маршруты, просмотрите таблицу маршрутизации ядра с помощью команды netstat -n -r. Вы должны увидеть что-то вроде следующего

```
[root]# netstat -nr
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
10.72.128.101	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.72.128.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	10.72.128.254	0.0.0.0	UG	0	0	0	eth0

Если таблица пуста, то вы увидите только заголовки столбцов. Тогда надо использовать `route`. С помощью команды `route` можно добавить или удалить один (за один раз) статический маршрут. Вот ее формат:

```
[root]# /sbin/route [-f] операция [-тип] адресат
шлюз [dev] интерфейс
```

Здесь аргумент операция может принимать одно из двух значений: `add` (маршрут добавляется) или `delete` (маршрут удаляется). Аргумент адресат может быть IP-адресом машины, IP-адресом сети или ключевым словом `default`. Аргумент шлюз - это IP-адрес компьютера, на который следует пересылать пакет (этот компьютер должен иметь прямую связь с вашим компьютером). Команда

```
[root]# /sbin/route -f
```

удаляет из таблицы данные обо всех шлюзах. Необязательный аргумент тип принимает значения `net` или `host`. В первом случае в поле адресата указывается адрес сети, а во втором - адрес конкретного компьютера (хоста).

Как правило, бывает необходимо настроить маршрутизацию по упоминавшимся выше трем интерфейсам:

- локальный интерфейс (`lo`),
- интерфейс для платы Ethernet (`eth0`),
- интерфейс для последовательного порта (PPP или SLIP).

Локальный интерфейс поддерживает сеть с IP-номером 127.0.0.1. Поэтому для маршрутизации пакетов с адресом 127.... используется команда:

```
[root]# /sbin/route add -net 127.0.0.1 lo
```

Если у вас для связи с локальной сетью используется одна плата Ethernet, и все машины находятся в этой сети (сетевая маска 255.255.255.0), то для настройки маршрутизации достаточно вызвать:

```
[root]# /sbin/route add -net 192.168.36.0 netmask
255.255.255.0 eth0
```

Если же вы имеете несколько интерфейсов, то вам надо определиться с сетевой маской и вызвать команду `route` для каждого интерфейса.

Поскольку очень часто IP-пакеты с вашего компьютера могут отправляться не в одну единственную сеть, а в разные сети (например, при

просмотре разных сайтов в Интернете), то в принципе надо было бы задать очень много маршрутов. Очевидно, что сделать это было бы очень сложно, точнее просто невозможно. Поэтому решение проблемы маршрутизации пакетов перекладывают на плечи специальных компьютеров - маршрутизаторов, а на обычных компьютерах задают маршрут по умолчанию, который используется для отправки всех пакетов, не указанных явно в таблице маршрутизации. С помощью маршрута по умолчанию вы говорите ядру "а все остальное отправляй туда". Маршрут по умолчанию настраивается следующей командой:

```
[root]# /sbin/route add default gw 192.168.1.1 eth0
```

Опция gw указывает программе route, что следующий аргумент - это IP-адрес или имя маршрутизатора, на который надо отправлять все пакеты, соответствующие этой строке таблицы маршрутизации.

После настройки маршрутизации можно проверить, что у вас получилось. Для этого снова дайте команду

```
[root]# netstat -nr
```

Если вывод команды выглядит так, как это было показано выше, но не содержит строки, которая в графе Destination содержит 0.0.0.0, а в графе Gateway указывает на маршрут, используемый для соединений по умолчанию, то вы, вероятно, не задали этот маршрут.

Настройка службы имен

С помощью команды ifconfig вы задали IP-адрес вашего компьютера, но он еще не знает своего имени (при инсталляции системы он получил обезличенное имя localhost). Существует команда hostname, которая позволяет установить (и узнать действующее в данный момент) имя компьютера и имя домена.

Однако установить только имя и только этой командой еще недостаточно, поскольку эта команда меняет имя только на текущий сеанс работы. Поэтому обычно эта команда вызывается в одном из инициализационных файлов, например, /etc/rc.d/rc или /etc/rc.d/rc.local. Вы можете попытаться найти ее там, чтобы изменить должным образом имя компьютера, которое задается в качестве параметра команды hostname. В таком случае требуется перезагрузиться для того чтобы изменения вступили в силу.

Другой способ изменения имени компьютера или домена состоит в том, что эти имена прописываются в файле /etc/sysconfig/network в виде двух строчек примерно следующего вида:

```
HOSTNAME="new_host_name.localdomain.upperdomain"  
DOMAINNAME=localdomain.upperdomain
```

Тогда в процессе инициализации системы эти имена будут восстанавливаться, потому что файл `/etc/sysconfig/network` вызывается из `/etc/rc.d/rc.sysinit`.

Кроме того, имя компьютера должно быть прописано в файле `/etc/hosts`, который связывает имя компьютера с его IP-адресом. Каждая строка файла `/etc/hosts` должна начинаться с IP-адреса, за которым следует имя данного узла. Следом за именем можно записать произвольное число псевдонимов этого узла.

Даже если ваш компьютер не подключен к сети, в файле `/etc/hosts` должна быть прописана хотя бы одна строка следующего вида.

```
127.0.0.1 localhost localhost.localdomain
```

Если же ваш компьютер подключен к TCP/IP сети, то в этом файле дополнительно нужно прописать строку вида

```
192.168.0.15 host_name host_name.localdomain
```

Файл `/etc/hosts` используется в механизмах разрешения имен. В больших сетях трудно было бы поддерживать в актуальном состоянии файлы `/etc/hosts` на всех компьютерах, если бы это был основной инструмент для определения IP-адресов по именам. Поэтому обычно для разрешения имен используются серверы DNS. Однако файл `/etc/hosts` все равно необходим, хотя бы для обращения к серверу DNS. Поэтому в нем имеет смысл указать IP-адреса и соответствующие имена шлюзов и серверов DNS и NIS. А чтобы все приложения использовали этот файл при разрешении имен, должен иметься файл `/etc/hosts.conf`, содержащий строку

```
order hosts,bind
```

которая говорит, что при разрешении имен сначала должен использоваться файл `/etc/hosts`, а затем должно происходить обращение к серверу DNS. В большинстве случаев в файле `/etc/hosts.conf` достаточно иметь две строки:

```
order hosts,bind
multi on
```

Эти параметры указывают системе преобразования имен, что надо просмотреть файл `/etc/hosts` перед тем, как посылать запрос к серверу, и что следует возвращать все найденные в `/etc/hosts` адреса для данного имени, а не только первый.

Но настройка механизма разрешения имен не ограничивается редактированием файлов `/etc/hosts` и `/etc/hosts.conf`. Необходимо еще указать компьютеру имена серверов DNS. Они прописываются в файле `/etc/resolv.conf`. Этот файл имеет весьма простой формат. Это текстовый файл, каждая строка которого задает один из параметров системы

преобразования имен. Как правило, используются три ключевых слова-параметра:

- `domain` - задает имя локального домена.
- `search` - задает список имен доменов, которые будут добавляться к имени машины, если вы не укажете явно имени домена. Это позволяет ограничить область поиска и избежать некоторых ошибок (например, вы ищете компьютер `linux.msk.ru`, а механизм разрешения имен выведет вас на `linux.spb.ru`).
- `nameserver` - этот параметр, который вы можете указывать несколько раз, задает IP-адрес сервера преобразования имен, на который ваша машина будет посылать запросы. Повторяя этот параметр, вы можете задать несколько серверов.

Если вы не собираетесь заводить поддержку сервиса имен для своей сети (что является довольно сложной организационной и технической проблемой), и доверяете ведение своих имен администратору локальной сети или вашему IP-провайдеру, то вам достаточно задать файл `/etc/resolv.conf` примерно следующего вида:

```
domain abcd.ru
search abcd.ru xyz.edu.ru
nameserver 192.168.10.1
nameserver 192.168.12.1
```

В этом примере машина находится в домене `abcd.ru`. Если вы зададите имя машины, не указывая домена, например `"pc1"`, то система преобразования имен попытается сначала найти машину `"pc1.abcd.ru"`, а в случае неудачи - `"pc1.xyz.edu.ru"`. Для преобразования имен ваша машина будет обращаться к серверам по адресам `"192.168.10.1"` и `"192.168.12.1"`.

Тестирование сетевого соединения

Чтобы проверить, соединяется ли ваш компьютер с сетью, попробуйте дать команду `ping`, указав ей в качестве параметра IP-адрес одного из компьютеров сети. Пусть, например, вам известно (узнайте реальный номер и имя у администратора сети), что в сети есть компьютер с IP-адресом `192.168.0.2` и именем `pc1`. Тогда вы должны дать команду:

```
[user]$ ping 192.168.0.2
```

или (тут вы одновременно проверяете и работу службы DNS)

```
[user]$ ping pc1
```

Если соединение с сетью установлено, должны появиться и периодически обновляться строчки примерно такого вида:

```
64 bytes from 192.168.0.2: icmp_seq=0 ttl=32
time=1.2 ms
```

```
64 bytes from 192.168.0.2: icmp_seq=1 ttl=32
time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=32
time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=32
time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=32
time=1.1 ms
```

Это означает, что сетевое соединение работает. Для того чтобы прервать тестирование сети, нажмите комбинацию клавиш <Ctrl>+<C>.

Программы telnet и rlogin

Для того чтобы воспользоваться программой telnet, вам необходимо знать имя или IP-адрес удаленного компьютера, работающего под управлением ОС типа UNIX, на котором для вас открыт пользовательский бюджет. Предположим для примера, что на компьютере linux2 имеется пользователь user5, пароль которого вам известен. В таком случае вы можете дать команду

```
[user]$ telnet linux2
```

Если программе удалось подключиться к указанному компьютеру, на экране появится сообщение "Connected to server linux2" и приглашение к входу в систему, как если бы вы сидели за терминалом компьютера linux2. Вводите имя (user5) и пароль, и вы будете работать на этом компьютере.

Команда rlogin может быть использована для выхода на удаленный компьютер вполне аналогично команде telnet, хотя лучше сразу указать в командной строке имя пользователя:

```
[user]$ rlogin -l user5 linux2
```

Завершив работу, не забудьте закрыть сессию (командой exit). После этого программа telnet (или rlogin) докладывает, что сессия закрыта, и вы возвращаетесь к командной строке локальной оболочки.

Когда вы работаете с программой telnet, вы полностью работаете на удаленном компьютере: команды выполняются в его оперативной памяти, вы видите (по команде ls) каталоги и файлы на дисках удаленного компьютера и т. д. Только вывод результатов осуществляется на ваш монитор. В рамках программы telnet невозможно, например, открыть для просмотра файл, расположенный на локальном диске. Ваш компьютер выполняет только роль удаленного терминала. Если же вы хотите организовать обмен файлами между вашим компьютером и удаленным, можно воспользоваться программой ftp.

Программа ftp

Программа ftp - это пользовательский интерфейс к стандартному протоколу передачи файлов по Интернету - File Transfer Protocol. Программа позволяет передавать файлы на удаленный компьютер и получать файлы с удаленного компьютера. Однако, введя команду ftp, вы запускаете только клиентскую программу. Для того чтобы получить доступ к файлам удаленного компьютера, на нем должен быть запущен ftp-сервер. Кроме того, необходимо знать либо имя и пароль пользователя, либо ftp-сервер должен разрешать анонимный доступ. Предположим, что эти условия выполнены и вы запустили программу ftp (без параметров). Вы увидите приглашение интерпретатора команд этой программы:

```
ftp >
```

Если ввести знак вопроса, программа выдаст перечень возможных команд. Первая команда, которую нужно в этом случае ввести, - команда `open`, после которой надо указать сетевое имя компьютера, на котором запущен ftp-сервер. Если анонимный доступ к этому серверу разрешен, то вы получите запрос на ввод имени и пароля пользователя. По команде `pwd` можно узнать имя текущего каталога на удаленном компьютере, а по команде `dir` - вывести список файлов и подкаталогов этого каталога. Команда `cd имя_каталога` используется для смены текущего каталога на удаленном компьютере.

В любой момент вы можете повторно ввести команду `?` или ее эквивалент `help`, чтобы получить подсказку по возможным командам. Для получения более подробной подсказки по конкретной команде надо ввести имя интересующей вас команды после `help` или `?`, например, так:

```
ftp > help dir
```

Если вы хотите выполнить какую-то команду на локальном компьютере (например, выяснить имя текущего каталога), надо дать соответствующую команду, перед которой поставить восклицательный знак:

```
ftp >! pwd
```

`!` - это команда интерпретатора, вызывающая новый экземпляр оболочки `shell` локального компьютера. Первый аргумент, следующий за `!`, должен быть командой оболочки, а все остальные аргументы - аргументами вызываемой команды. Для смены текущего каталога на локальном компьютере имеется специальная команда `lcd` (очень полезная, поскольку часто до запуска ftp забываешь перейти в тот каталог, куда хочешь скопировать файл с удаленного компьютера; не выходить же из-за этого из программы ftp).

Для пересылки файла на удаленный компьютер используется команда

```
ftp > put имя_файла
```

(или ее синоним send), а для копирования файла с удаленного компьютера в текущий каталог на локальном диске - команда

```
ftp > get имя_файла
```

В принципе этих двух команд вполне достаточно для организации обмена файлами с удаленным компьютером, но как же неудобно ими пользоваться! Приходится набирать полностью имена всех пересылаемых файлов. Поэтому испытываешь воистину большое облегчение, когда узнаешь, что существуют такие команды как mput и mget. Они позволяют задать шаблон имени пересылаемых файлов, и будут дополнительно переспрашивать, надо ли пересылать каждый конкретный файл. Благодаря этому можно (самый крайний случай) заказать пересылку всех файлов:

```
ftp > mget *
```

а потом либо подтверждать пересылку очередного файла, либо отказываться. Конечно, когда файлов в каталоге очень много, то и это окажется утомительной процедурой, но ведь можно задать более разумный шаблон! Так что думайте, как облегчить себе работу.

Перед тем, как начать пересылку файлов, следует еще выполнить одну из команд, определяющих режим пересылки: ascii или binary . По умолчанию программа использует режим "ascii", и это вполне допустимо при пересылке текстовых файлов, но если вы собираетесь передать или получить исполняемый файл, то необходимо задать режим "binary". Процесс пересылки файлов можно прервать с помощью комбинации клавиш <Ctrl>+<C>.

Пока вы находитесь в программе ftp, вы можете выполнить некоторые операции с файлами и каталогами на удаленном компьютере (конечно, для этого надо иметь соответствующие права). По команде

```
ftp > rename from_name to_name
```

осуществляется переименование файла или каталога; команда

```
ftp > mkdir name
```

создает каталог, а

```
ftp > delete name
```

удаляет файл или каталог. Еще одна интересная команда - system, позволяет выяснить тип операционной системы на удаленном компьютере. Ну, и наконец, команда close (или disconnect) позволяет завершить сеанс работы с удаленным компьютером, не выходя из программы ftp (т. е. предполагается, что после этого вы снова дадите команду open, например, для соединения с другим компьютером). Если же вы хотите вообще выйти из программы, то надо дать команду bye.

Виртуальные терминалы и интерфейс командной строки NetWork Simulator'a.

Виртуальные устройства в NET-Simulator управляются при помощи интерфейса командной строки из виртуальных терминалов. Терминал устройства можно открыть двойным кликом на значке устройства или через контекстное меню. Поддерживается история команд, клавиши вверх/вниз позволяют просматривать историю команд.

Список команд доступных на данном устройстве можно посмотреть командой `help`. Сочетание клавиш `Ctrl+L` очищает терминал. Краткая справка по любой команде выводится при вызове команды с опцией `-h`.

Справочник команд:

- `help`
- `route`
- `ifconfig`
- `ping`
- `arp`
- `mactable`

help — выводит список доступных команд.

`help [-h]` Опции Описание

`-h` Краткая справка.

route — позволяет управлять таблицей маршрутизации устройств поддерживающих протокол IP4.

`route [-h] [{-add|-del} <target> [-netmask <address>] [-gw <address>] [-metric <M>] [-dev <If>]]` Опции Описание

`-h` Краткая справка.

`target` Адрес назначения. Назначением может быть подсеть или отдельный узел в зависимости от значения маски подсети. Если маска равна 255.255.255.255 или отсутствует совсем назначением будет узел, иначе назначением будет сеть.

`-add` Добавляет новый маршрут в таблицу маршрутизации.

`-del` Удаляет маршрут из таблицы маршрутизации.

`-dev <If>` Принудительно присоединяет маршрут к определенному интерфейсу. `If` — имя интерфейса.

-gw <address> Направляет пакеты по этому маршруту через заданный шлюз. address — адрес шлюза.

-netmask <address> Маска подсети используемая совместно с адресом назначения при добавлении маршрута. address — маска. Если маска не задана явно подразумевается 255.255.255.255.

-metric <M> Метрика используемая в данном маршруте. M — целое число большее или равное нулю.

Если route вызывается без параметров, то команда выводит на экран таблицу маршрутизации:

```
=>route
```

```
IP routing table
```

Destination	Gateway	Netmask	Flags	Metric	Iface
10.0.0.0	*	255.0.0.0	U 1	eth0	
11.0.0.0	10.0.0.10	255.0.0.0	UG 1	eth0	
192.168.120.1	10.0.0.10	255.255.255.255	UGH 1	eth0	

Если маршрут не использует шлюз, вместо адреса шлюза выводиться *.
Flags может содержать значение: U — маршрут активен, G — маршрут использует шлюз, H — назначением является узел.

Примеры:

```
=>route -add 192.168.120.0 -netmask 255.255.255.0 -dev eth0
```

```
=>route
```

```
IP routing table
```

Destination	Gateway	Netmask	Flags	Metric	Iface
192.168.120.0	*	255.255.255.0	U 1	eth0	

```
=>
```

```
=>route -add 192.168.121.10 -gw 192.168.120.10
```

```
=>route
```

```
IP routing table
```

Destination	Gateway	Netmask	Flags	Metric	Iface
192.168.120.0	*	255.255.255.0	U 1	eth0	
192.168.121.10	192.168.120.1	255.255.255.255	UGH 1	eth0	

```
=>
```

ifconfig — конфигурирует сетевые интерфейсы.

ifconfig [-h] [-a] [<interface>] [<address>] [-broadcast <address>] [-netmask <address>] [-up|-down] Описания

-h Краткая справка.

-a Показывать информацию о всех интерфейсах. Если данная опция отсутствует выводится информация только об активных интерфейсах.

interface Конфигурировать или показать информацию только о заданном интерфейсе.

address IP-адрес присваиваемый интерфейсу.

-broadcast <address> Широковещательный адрес присваиваемый интерфейсу. address — широковещательный адрес.

-netmask <address> Маска подсети используемая совместно с адресом. address — маска. Если маска не задана явно, маска принимается равной стандартным значения для стандартных классов подсетей А, В и С.

-up Активирует интерфейс. При активизации интерфейса для него автоматически добавляется соответствующий маршрут в таблице маршрутизации.

-down Деактивирует интерфейс. При деактивации интерфейса соответствующий маршрут автоматически удаляется из таблицы маршрутизации.

Если ifconfig вызывается без параметров, то команда выводит на экран данные о состоянии всех активных интерфейсов:

```
=>ifconfig
eth0  Link encap:Ethernet HWaddr 0:0:0:0:CF:0
      inet      addr:192.168.120.1          Bcast:192.168.120.255
Mask:255.255.255.0
      UP
      RX packets:23 errors:0 dropped:0
      TX packets:23 errors:0 dropped:0
      RX bytes:0 TX bytes:0
```

HWaddr — уникальный 6-ти байтовый адрес интерфейса, аналогичный MAC-адресу в Ethernet сетях. Назначается автоматически.

Примеры:

```
=>ifconfig eth0 192.168.120.1 -up
=>ifconfig
eth0  Link encap:Ethernet HWaddr 0:0:0:0:CF:0
      inet      addr:192.168.120.1          Bcast:192.168.120.255
Mask:255.255.255.0
      UP
```

```
RX packets:0 errors:0 dropped:0
TX packets:0 errors:0 dropped:0
RX bytes:0 TX bytes:0
```

ping — использует ICMP протокол что бы проверить достижимость интерфейса удаленного узла. ping посылает удаленному узлу ICMP ECHO_REQUEST и ожидает в течении определенного промежутка времени ICMP ECHO_RESPONSE. В случае получения ответа выводит данные о прохождении ICMP-пакета по сети.

ping [-h] [-i <interval>] [-t <ttl>] <destination>	Опции	Описание
-h		Краткая справка.
-i <interval>		Задаёт частоту ICMP-запросов. interval — интервал между запросами в секундах. По умолчанию отсылается один пакет в секунду.
-t <ttl>		Задаёт значение атрибута Time to Live в генерируемых IP-пакетах. ttl — целое число 0-255. По умолчанию TTL равно 64.
destination		IP-адрес исследуемого узла

Примеры:

```
=>ping 192.168.120.1
PING 192.168.120.1
64 bytes from 192.168.120.1: icmp_seq=0 ttl=62 time=477 ms
64 bytes from 192.168.120.1: icmp_seq=1 ttl=62 time=435 ms
64 bytes from 192.168.120.1: icmp_seq=2 ttl=62 time=234 ms
64 bytes from 192.168.120.1: icmp_seq=3 ttl=62 time=48 ms
64 bytes from 192.168.120.1: icmp_seq=4 ttl=62 time=87 ms
64 bytes from 192.168.120.1: icmp_seq=5 ttl=62 time=56 ms
```

ping выводит результат исследования удаленного узла в следующем формате: 64 bytes from 192.168.120.1 — размер полученного ответа и адрес источника ответа. В NET-Simulator размер пакета имеет условное значение и всегда равен 64В. icmp_seq=0 — номер пакета. Каждый запрос содержит свой номер, как правило формируется инкрементно. ping выводит номер пакета из каждого полученного ответа. ttl=62 — значение TTL из полученного ответа. time=48 ms — время прохождения пакетом полного маршрута (туда и обратно, round-trip time) в миллисекундах.

arp — показывает ARP-таблицу устройства. Кроме того опция -g позволяет сформировать запрос для определения MAC-адреса по явно

заданному IP-адресу. Эта функция обычно отсутствует в реальных устройствах, в NET-Simulator она добавлена для наглядности при изучении протоколов канального и сетевого уровня.

arp [-h] [-r <IP-address> <interface>] Опции Описание

-h Краткая справка.

-r <IP-address> <interface> Прежде чем вывести ARP-таблицу предпринимает попытку найти MAC-адрес по явно заданному IP-адресу. IP-address IP-адрес для которого определяется MAC-адрес. interface имя интерфейса в сети подсоединенной к которому будет происходить поиск.

Если arp вызывается без параметров, то команда выводит на экран ARP-таблицу:

=>arp

Address	HWaddress	iface
10.0.0.10	0:0:0:0:BC:0	eth0
10.0.0.11	0:0:0:0:1F:2	eth0

Примеры:

=>arp -r 192.168.120.12 eth1

Address	HWaddress	iface
10.0.0.10	0:0:0:0:BC:0	eth0
10.0.0.11	0:0:0:0:1F:2	eth0
192.168.120.12	0:0:0:0:12:1	eth1

mactable — показывает таблицу MAC-адресов коммутаторов второго уровня.

mactable [-h] Опции Описание

-h Краткая справка.

Примеры:

=>mactable

MACAddress	port
0:0:0:0:B3:0	0
0:0:0:0:2F:2	0
0:0:0:0:03:0	3

где port — номер порта на коммутаторе. Нумерация портов идет по порядку начиная с нуля.