**Лабораторное занятие №1**

**Тема:**Установка и настройка протокола TCP/IP в ОС Windows и Linux.

**Цель работы:** Изучить способы диагностики настроек стека протоколов TCP/ IP; получить сведения о настройке TCP/IP для работы с DHCP сервером.

**Оборудование**: Компьютер в сборе или испытательный стенд.

**Ход работы**

1. Ознакомиться с теоретической частью.

2. Выполнить задания.

3. Ответить на контрольные вопросы.

4. Оформить отчет.

**Теоретическая часть**

На концептуальной модели взаимодействия открытых систем OSI основан стек протоколов ***TCP/IP****(Transmission Control Protocol - протокол управления передачей / Internet Protocol – Интернет-протокол)*, который предоставляет ряд стандартов для связи компьютеров и сетей.

***Стек протоколов TCP/IP*** – промышленный стандарт, который позволяет организовать сеть масштаба предприятия и связывать компьютеры, работающие под управлением различных операционных систем.

Применение стека протоколов TCP/IP дает следующие преимущества:

·поддерживается почти всеми операционными системами; почти все большие сети основаны на TCP/IP;

·технология позволяет соединить разнородные системы;

·надежная, расширяемая интегрированная среда на основе модели «клиент — сервер»;

·получение доступа к ресурсам сети Интернет.

Каждый узел ***TCP/IP*** идентифицирован своим логическим IP-адресом, который идентифицирует положение компьютера в сети почти таким же способом, как номер дома идентифицирует дом на улице.

Реализация ***TCP/IP*** позволяет узлу ***TCP/IP*** использовать статический IP-адрес или получить IP-адрес автоматически с помощью ***DHCP-сервера****(Dynamic Host Configuration Protocol - протокол динамической конфигурации хоста).*

Для простых сетевых конфигураций, основанных на локальных сетях (*LAN, Local Area Network*), он поддерживает автоматическое назначение IP-адресов.

По умолчанию компьютеры клиентов, работающие под управлением ОС **Windows** или **Linux**, получают информацию о настройке протокола ***TCP/IP*** автоматически от службы ***DHCP***.

Однако даже в том случае, если в сети доступен ***DHCP-сервер***, необходимо назначить статический IP-адрес для отдельных компьютеров в сети. Например, компьютеры с запущенной службой ***DHCP*** не могут быть клиентами ***DHCP***, поэтому они должны иметь статический IP-адрес.

Если служба ***DHCP*** недоступна, можно настроить ***TCP/IP*** для использования статического IP-адреса.

Для каждой платы сетевого адаптера в компьютере, которая использует ***TCP/IP***, можно установить IP-адрес, маску подсети и шлюз по умолчанию.

Ниже описаны параметры, которые используются при настройке статического адреса ***TCP/IP***.

|  |  |
| --- | --- |
| Параметр | Описание |
| IP-адрес | Логический 32-битный адрес, который идентифицирует TCP/IP узел. Каждой плате сетевого адаптера в компьютере с запущенным протоколом TCP/IP необходим уникальный IP-адрес, такой, как 192.168.0.108. Каждый адрес имеет две части: ID сети, который идентифицирует все узлы в одной физической сети и ID узла, который идентифицирует узел в сети. В этом примере ID сети — 192.168.0, и ID узла — 108. |
| Маска подсети | Подсети делят большую сеть на множество физических сетей, соединенных маршрутизаторами. Маска подсети закрывает часть IP-адреса так, чтобы TCP/IP мог отличать ID сети от ID узла. При соединении узлов TCP/IP, маска подсети определяет, где находится узел получателя: в локальной или удаленной сети. Для связи в локальной сети компьютеры должны иметь одинаковую маску подсети. |
| Шлюз по умолчанию | Промежуточное устройство в локальной сети, на котором хранятся сетевые идентификаторы других сетей предприятия или Интернета. TCP/IP посылает пакеты в удаленную сеть через шлюз по умолчанию (если никакой другой маршрут не настроен), который затем пересылает пакеты другим шлюзам, пока пакет не достигнет шлюза, связанного с указанным адресатом. |

Если сервер с запущенной службой ***DHCP*** доступен в сети, он автоматически предоставляет информацию о параметрах ***TCP/IP*** клиентам ***DНСР***.

**Настройка и диагностика сети в ОС Linux**

Для работы с сетевыми протоколами TCP/IP в Linux достаточно наличие только петлевого интерфейса, но если необходимо объединить хосты между собой, естественно, необходимо наличие сетевого интерфейса, каналов передачи данных (например витая пара), возможно, какого-либо сетевого оборудования. Так же, необходимо наличие установленных [утилит для настройки сети](http://www.k-max.name/linux/osnovnye-komandy-linux-ili-shpargalka-nachinayushhego-linuksojda#network) ([/sbin/ifconfig](http://www.k-max.name/linux/osnovnye-komandy-linux-ili-shpargalka-nachinayushhego-linuksojda/#ifconfig), [/sbin/route](http://www.k-max.name/linux/osnovnye-komandy-linux-ili-shpargalka-nachinayushhego-linuksojda/#route) и др.), обычно поставляемые в [пакете net-tools](http://www.k-max.name/linux/upravlenie-programmnym-obespecheniem-v-linux/#pakage). Так же необходимо наличие [конфигурационных файлов](http://www.k-max.name/linux/fajlovaya-sistema-linux-i-struktura-katalogov/) для сети (например /etc/hosts) и поддержку сети [ядром Linux](http://www.k-max.name/category/linux/yadro/).

***Файлы настроек сети в Linux (конфигурационные файлы)***

В целом, вся работа Linux основана на [процессе init](http://www.k-max.name/linux/nachalo-etapy-zagruzki-os-linux-v-sxeme/#init), который рождается при загрузке ОС и плодит своих потомков, которые в свою очередь и выполняют всю необходимую работу. Вся загрузка Linux основана на [скриптах bash](http://www.k-max.name/category/linux/bash/), в которых прописана вся последовательность запуска мелких утилит с различными параметрами, которые последовательно запускаются/останавливаются при запуске/остановке системы. Аналогично запускается и сетевая подсистема Linux.

/etc/init.d/networking

Это скрипт, отвечающий за инициализацию сети.

/etc/hosts

Данный файл хранит перечень IP адресов и соответствующих им (адресам) имен хостов. Исторически, данный файл использовался вместо службы DNS. В настоящее время, файл так же может использоваться вместо службы DNS, но только при условии, что в вашей сети количество машин измеряется в единицах, а не в десятках или сотнях, потому что в таком случае, придется контролировать корректность данного файла на каждой машине.

/etc/networks

Данный файл хранит имена и адреса локальной и других сетей. При использовании данного файла, сетями можно управлять по имени. Например добавить маршрут не route add 192.168.1.12, а route add home-network.

/etc/nsswitch.conf

Файл определяет порядок поиска имени хоста/сети.

/etc/resolv.conf

Данный фал определяет параметры механизма преобразования сетевых имен в IP адреса. Простым языком, определяет настройки DNS.

***Настройка сети***

Чтобы быть уверенным в работоспособности команды в любом дистрибутиве, необходимо пользоваться двумя основными командами. Это [**ifconfig**](http://www.k-max.name/linux/osnovnye-komandy-linux-ili-shpargalka-nachinayushhego-linuksojda/#ifconfig) и [**route**](http://www.k-max.name/linux/osnovnye-komandy-linux-ili-shpargalka-nachinayushhego-linuksojda/#route). Первая команда (**ifconfig**) отвечает за настройку сетевых интерфейсов (ip, маска, шлюз), вторая (**route**) - настройка маршрутизации. Следует заметить, что выполнение данных команд без отключения стандартного скрипта запуска сетевой подсистемы внесет изменения только до первой перезагрузки/перезапуска сетевой службы, т.к. скрипт /etc/init.d/networking при очередном запуске перечитает указанные выше конфиги и применит старые настройки. Соответственно, выход для постоянной установки настроек - либо команда **ifconfig** с соответствующими параметрами - вписать в [rc.local](http://www.k-max.name/linux/nachalo-etapy-zagruzki-os-linux-v-sxeme/#init), либо поправить руками соответствующие конфиги сетевых интерфейсов.

Так же, если выполняется команда **ifconfig** с недостающими параметрами (например только IP адрес), то остальные дополняются автоматически (например бродкаст адрес добавляется по умолчанию с хостовым адресом, оканчивающимся на 255 и маска подсети по умолчанию берется 255.255.255.0).

Маршрутизация для имеющихся интерфейсов всегда поднимается автоматически силами ядра. Вернее сказать, прямые марштуры в сеть, в которую смотрит поднятый интерфейс формируются автоматически, силами ядра. Поле gateway (шлюз) для таких записей показывает адрес выходного интерфейса или \*. Если есть необходимость организовать свои маршруты, то необходимо воспользоваться [командой**route**](http://www.k-max.name/linux/osnovnye-komandy-linux-ili-shpargalka-nachinayushhego-linuksojda/#route). Данной командой можно добавлять и удалять маршруты, но опять же, это поможет только до перезапуска /etc/init.d/networking (или другого скрипта, отвечающего за вашу сеть). Чтобы маршруты добавлялись автоматом, необходимо так же, как и с ifconfig - добавить команды добавления маршрутов в rc.local, либо поправить руками соответствующие конфиги сетевых интерфейсов (например в Deb - /etc/network/options).

***Диагностика сети Linux***

Существует большое количество инструментов диагностики сети в Linux, зачастую, они очень похожи на утилиты от Microsoft.

**ping**

Работа этой утилиты заключается в отправке т.н. пакетов ICMP удаленному серверу, который будет указан в параметрах команды, сервер возвращает отправленные команды, а **ping** подсчитывает время требуемое отправленному пакету, чтобы дойти до сервера и вернуться.

[root@proxy ~]# ping ya.ru

PING ya.ru (87.250.251.3) 56(84) bytes of data.

64 bytes from www.yandex.ru (87.250.251.3): icmp\_seq=1 ttl=57 time=42.7 ms

64 bytes from www.yandex.ru (87.250.251.3): icmp\_seq=2 ttl=57 time=43.2 ms

64 bytes from www.yandex.ru (87.250.251.3): icmp\_seq=3 ttl=57 time=42.5 ms

64 bytes from www.yandex.ru (87.250.251.3): icmp\_seq=4 ttl=57 time=42.5 ms

64 bytes from www.yandex.ru (87.250.251.3): icmp\_seq=5 ttl=57 time=41.9 ms

^C

--- ya.ru ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4012ms

rtt min/avg/max/mdev = 41.922/42.588/43.255/0.500 ms

Как видно, из приведенного примера, **ping** выводит нам кучу полезной информации. Прежде всего, мы выяснили, что можем установить соединение с хостом ya.ru (иногда говорят, что «хост ya.ru нам доступен»). Во-вторых, мы видим, что DNS работает корректно, потому что «пингуемое» имя было корректно преобразовано в IP адрес (ping ya.ru (87.250.251.3)). Далее, в поле icmp\_seq= указана нумерация отправляемых пакетов. Каждому отправляемому пакету последовательно присваивается номер и если в данной нумерации будут «провалы», то это нам расскажет о том, что соединение с «пингуемым» неустойчиво, а так же может означать, что сервер, которому шлют пакеты перегружен. По значению time= мы видим, сколько времени пакет путешествовал до 87.250.251.3 и обратно. Остановить работу утилиты **ping** можно клавишами Ctrl+C.

Так же, утилита **ping** интересна тем, что может позволить увидеть, где именно возникли неполадки. Допустим, утилита **ping** выводит сообщение network not reachable (сеть недоступна), либо другое аналогичное сообщение. Это, скорее всего, говорит о некорректной настройке вашей системы. В таком случае, можно послать пакеты по IP-адресу провайдера, чтобы понять, в каком месте возникает проблема (между локальным ПК или «дальше»). Если Вы подключены к интернету через маршрутизатор, то можно послать пакеты по его IP. Соответственно, если проблема проявиться уже на этом этапе, это говорит, о неправильном конфигурировании локальной системы, либо о повреждении кабеля, если маршрутизатор отзывается, а сервер провайдера нет, то проблема - в канале связи провайдера и т.д. Наконец, если неудачей завершилось преобразовании имени в IP, то можно проверить связь по IP, если ответы будут приходить корректно, то можно догадаться, что проблема в DNS.

Следует отметить, что данная утилита не всегда надежный инструмент для диагностики. Удаленный сервер может блокировать ответы на ICMP запросы.

**traceroute**

Простым языком, команда называется трассировка маршрута. Как можно понять из названия - данная утилита покажет по какому маршруту шли пакеты до хоста. Утилита **traceroute** несколько похожа на **ping**, но отображает больше интересной информации. Пример:

[root@proxy ~]# traceroute ya.ru

traceroute to ya.ru (213.180.204.3), 30 hops max, 60 byte packets

1 243-083-free.kubtelecom.ru (213.132.83.243) 6.408 ms 6.306 ms 6.193 ms

2 065-064-free.kubtelecom.ru (213.132.64.65) 2.761 ms 5.787 ms 5.777 ms

3 lgw.kubtelecom.ru (213.132.75.54) 5.713 ms 5.701 ms 5.636 ms

4 KubTelecom-lgw.Krasnodar.gldn.net (194.186.6.177) 81.430 ms 81.581 ms 81.687 ms

5 cat26.Moscow.gldn.net (194.186.10.118) 47.789 ms 47.888 ms 48.011 ms

6 213.33.201.230 (213.33.201.230) 43.322 ms 41.783 ms 41.106 ms

7 carmine-red-vlan602.yandex.net (87.250.242.206) 41.199 ms 42.578 ms 42.610 ms

8 www.yandex.ru (213.180.204.3) 43.185 ms 42.126 ms 42.679 ms

Как видно, можно проследить маршрут от маршрутизатора провайдера 243-083-free.kubtelecom.ru (213.132.83.243) (Юг России) до конечного хоста в [www.yandex.ru](http://www.yandex.ru/) (213.180.204.3) в Москве.

**dig**

Данная утилита посылает запросы серверам DNS и возвращает информацию о заданном домене.

*Консольные команды управления сетевым интерфейсом.*

**ifconfig** [опции] – стандартная команда Unix управления сетевым интерфейсом. В Mandriva Linux управляющие возможности этой команды ограничены.

**ifup** <устройство**>** - консольная команда включения сетевого интерфейса.

**ifdown** <устройство**>** - консольная команда выключения сетевого интерфейса.

**ping** <ip адрес**>** - команда проверки функционирования сети с использованием icmp пакета. Для выхода из режима проверки сети необходимо нажать CTRL**+C.**

**arp** [опции]– Команда печати таблицы памяти МАС адресов компьютера.

**Задания на лабораторную работу**

**Задание 1. Проверьте работоспособность стека протоколов TCP/IP в ОС Windows.**

1.   Запустите виртуальную машину и загрузите ОС Windows.

2.   Запустите консоль ***(Пуск/Программы/Стандартные/Командная строка)***.

3.   В командной строке введите **ipconfig /all**.

4.   Используя приведенную ниже информацию, создайте в своей папке текстовый документ со следующими данными:

o Имя компьютера;

o Основной DNS-суффикс;

o Описание DNS-суффикса для подключения;

o Физический адрес;

o DHCP включен;

o Автоконфигурация включена;

o IP-адрес автоконфигурации;

o Маска подсети;

o Шлюз по умолчанию.

5.   Убедитесь в работоспособности стека ***TCP/IP***, отправив эхо-запросы на IP-адреса. Для этого воспользуйтесь командой **ping**:

o отправьте эхо-запросы на локальный адрес компьютера (*loopback*) **ping 127.0.0.1** (на экране должны появиться сообщения о полученном ответе от узла 127.0.0.1);

o отправьте эхо-запрос по другому IP-адресу, например **192.168.10.1**.

**Задание 2. Настройте стек протоколов *TCP/IP* для использования статического IP-адреса.**

1.   Откройте окно **Сетевые подключения *(Пуск/Панель управления/Сетевые подключения)***.

2.   Вызовите **свойства подключения по локальной сети**. Для этого можно воспользоваться контекстным меню.

3.   В появившемся диалоговом окне на вкладке **Общие** откройте свойства **Протокол Интернета TCP/IP**.

4.   Щелкните переключатель *Использовать следующий IP-адрес* и введите в соответствующие поля данные:**IP\_адрес; Маску подсети; Основной шлюз; Предпочитаемый DNS**.

5.   Примените параметры кнопкой ***ОК***.

6.   Закройте окно свойств подключения кнопкой ***ОК*** (если потребуется, то согласитесь на перезагрузку компьютера).

7.   Проверьте работоспособность стека протоколов ***TCP/IP***.

**Задание 3. Настройте TCP/IP для автоматического получения IP-адреса.**

1.   Откройте окно **Сетевые подключения**.

2.   Вызовите свойства **Подключения по локальной сети**.

3.   Откройте свойства **Протокол Интернета TCP/IP**.

4.   Установите переключатель *Получить IP-адрес автоматически*.

5.   Закройте диалоговое окно **Свойства: Протокол Интернета TCP/IP** кнопкой ***ОК***.

6.   Примените параметры кнопкой ***ОК***.

7.   Проверьте настройку стека протоколов ***TCP/IP***.

8.   Получите другой адрес для своего компьютера. Для этого:

o запустите консоль (командную строку);

o введите команду для сброса назначенных адресов - **ipconfig /release**;

o введите команду для получения нового адреса **ipconfig /renew**;

9.   Проверьте работоспособность стека протоколов ***TCP/IP***.

**Задание 4. Настройка и диагностика сети в ОС Linux.**

 1 Загрузите ОС Linux

 2 Изучите файл настройки сетевого интерфейса: /etc/sysconfig/network-scripts/ifcfg-eth0

 3 Изучите руководство программы **ifconfig**.

 4 Изучите руководство программы **ifcfg**.

 5 Просмотрите сетевые интерфейсы при помощи команды **ifconfig**.

 6 Выключите интерфейс eth0, применив консольную команду **ifdown**.

 7 Просмотрите изменение параметров настройки сетевого интерфейса eth0 при помощи команды **ifconfig**.

 8 Сделайте копию экрана во время выполнения операции для отчета по лабораторной работе.

 9 Включите интерфейс eth0 при помощи команды **ifup**.

 10 Присвойте интерфейсу eth0 IP-адрес из диапазона 192.168.0.190-192.168.0.200 с маской 255.255.255.0 при помощи команды **ifconfig**.

 11 Просмотрите изменение параметров настройки сетевого интерфейса eth0 при помощи команды **ifconfig**.

 12 Выведите кэш МАС адресов при помощи команды **arp**.

 13 Сделайте копию экрана во время выполнения операции для отчета по лабораторной работе.

 14 Проверьте функционирование сетевого интерфейса при помощи команды **ping: ping**<адрес вашей машины>.

 15 Проверьте функционирование сети при помощи команды **ping: ping**<адрес другой включенной машины класса> .

 16 Сделайте копию экрана во время выполнения операции для отчета по лабораторной работе.

**Содержание отчета**

Тема, цель, оборудование, порядок выполнения заданий, ответы на контрольные вопросы, вывод.

**Контрольные вопросы**

1.   Что означает понятие «статический IP-адрес».

2.   Как происходит автоматическое получение IP-адреса.

3.   Назначение маски подсети.

4.   Как выполнить тестирование протокола TCP/IP.

5.     Что такое интерфейс **lo**? Какой у него адрес?

6.   Почему **ping** до вашей сетевой платы идет быстрее чем до другого компьютера класса?