**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

Лабораторная работа №2

**Минск 2024**

**СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ 2](#_heading=h.gjdgxs)

[1. Постановка задачи 3](#_heading=h.30j0zll)

[2. Краткая теоретическая справка 3](#_heading=h.1fob9te)

[3. Сетевые утилиты 3](#_heading=h.3znysh7)

[3.1 Утилита hostname 3](#_heading=h.2et92p0)

[3.2 Утилита ipconfig 4](#_heading=h.tyjcwt)

[3.3 Утилита net view (нужны права администратора) 6](#_heading=h.3dy6vkm)

[3.4 Утилита ping 7](#_heading=h.1t3h5sf)

[3.5 Утилита netstat 12](#_heading=h.4d34og8)

[3.6 Утилита tracert 15](#_heading=h.2s8eyo1)

[3.7. Утилита pathping 17](#_heading=h.17dp8vu)

[3.8 Утилита arp 19](#_heading=h.3rdcrjn)

[3.9 Утилита net send (уже устарела) 21](#_heading=h.35nkun2)

[3.10. Утилита Route 21](#_heading=h.1ksv4uv)

[4. Задания и вопросы для выполнения лабораторной работы № 2 22](#_heading=h.44sinio)

[4.1 Варианты ссылок 22](#_heading=h.2jxsxqh)

[4.2 Варианты заданий 23](#_heading=h.z337ya)

[4.3 Задания для отчета 23](#_heading=h.3j2qqm3)

[4.3.1 *Задание 1*. Получение справочной информации по командам 23](#_heading=h.1y810tw)

[4.3.2. *Задание 2.* Получение имени хоста 23](#_heading=h.1pxezwc)

[4.3.3. *Задание 3.* Изучение утилиты ipconfig 23](#_heading=h.49x2ik5)

[4.3.4. *Задание 4*. Тестирование связи с помощью утилиты ping 24](#_heading=h.2p2csry)

[4.3.5. *Задание 5.* Утилита Tracert. Определение пути IP-пакета 24](#_heading=h.147n2zr)

[4.3.6. *Задание 6*. Просмотр ARP-кэша 25](#_heading=h.3o7alnk)

[4.3.7. *Задание 7*. Утилита netstat. 25](#_heading=h.23ckvvd)

[4.3.8. *Задание 8*. 25](#_heading=h.3as4poj)

**1. Постановка задачи**

Используя стандартные сетевые утилиты, проанализировать конфигурацию сети на платформе ОС Windows, т.е. получить свой IP-адрес, узнать имя рабочей группы, имена компьютеров, входящих в группу, просмотреть и при необходимости подключить общие ресурсы, определить причину возможных неполадок, так же получить информацию об использовании портов и т.д. Выполнить задания, ответить на вопросы и предоставить отчет.

**2. Краткая теоретическая справка**

Мониторинг и анализ сети представляют собой важные этапы контроля работы сети. Для решения этих задач регулярно производится сбор данных, который дает базу данных для измерения реакции сети на изменения и перегрузки. Чтобы осуществить сетевую передачу нужно проверить корректность подключения клиента к сети, наличие у клиента хотя бы одного протокола сервера, знать IP-адрес компьютеров сети и т. д. Поэтому в сетевых операционных системах, и в частности, в Windows, существует множество мощных утилит для пересылки текстовых сообщений, управления общими ресурсами, диагностике сетевых подключений, поиска и обработки ошибок.

Утилиты запускаются из сеанса интерпретатора команд Windows   
(**Пуск -> Выполнить -> cmd**).

**3. Сетевые утилиты**

Утилитами называются сравнительно небольшие программы, предназначенные для решения каких-либо *узкоспециализированных* задач. В данной лабораторной работе рассматриваются утилиты операционной системы Windows, используемые для диагностики сетевых подключений. Для диагностики проблем в сетях доступно множество программных средств. Многие из них реализуются операционной системой и доступны в виде команд в интерфейсе командной строки. Синтаксис команд в разных операционных системах различается. Обзор утилит совместим с описанием основ теории компьютерных сетей.

**3.1 Утилита hostname**

Выводит имя локального компьютера (хоста). Она доступна только после установки поддержки протокола TCP/IP. Пример вызова команды **hostname**:

*C:\Documents and Settings\User>****hostname***

**3.2 Утилита ipconfig**

**Утилита ipconfig** предназначена для получения информации о настройках сетевых интерфейсов, выводит диагностическую информацию о конфигурации сети TCP/IP. Эта утилита позволяет просмотреть текущую конфигурацию IP-адресов компьютеров сети. Синтаксис утилиты ***ipconfig****:*

***ipconfig*** *[/all* | */renew* [адаптер] | */release* [адаптер]],

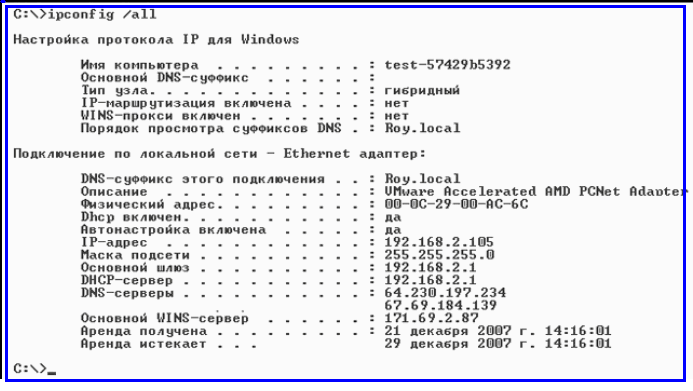
где *all* - выводит сведения об имени хоста, DNS (Domain Name Service), типе узла, IP-маршрутизации и др. Без этого параметра команда *ipconfig* выводит только IP-адреса, маску подсети и основной шлюз;

*/renew* [адаптер] - обновляет параметры конфигурации DHCP (Dynamic Host Configuration Protocol - автоматическая настройка IP-адресов). Эта возможность доступна только на компьютерах, где запущена служба клиента DHCP. Для задания адаптера используется имя, выводимое командой *ipconfig* без параметров;

*/release* [адаптер] - очищает текущую конфигурацию DHCP. Эта возможность отключает TCP/IP на локальных компьютерах и доступна только на клиентах DHCP. Для задания адаптера используется имя, выводимое командой *ipconfig* без параметров. Эта команда часто используется перед перемещением компьютера в другую сеть. После использования утилиты *ipconfig /release,* IP-адрес становиться доступен для назначения другому компьютеру.

Если после отмены конфигурации IP узел не может получить текущую информацию с DHCP-сервера, проблема может состоять в потере связи с сетью. В этом случае необходимо убедиться, что на сетевой плате горит индикатор физического соединения с сетью (LINK). Если описанными мерами устранить проблему не удалось, ее источником может быть DHCP-сервер или сетевые соединения с DHCP-сервером.

Пример использования *ipconfig* с параметром */all*:



**Пример** с комментарием использования *ipconfig.*

**C:\Users\gorvv>ipconfig**

**Настройка протокола IP для Windows**

**Ethernet adapter Сетевое подключение Bluetooth:**

**Состояние среды. . . . . . . . : Среда передачи недоступна.**

**DNS-суффикс подключения . . . . . :**

**Адаптер беспроводной локальной сети Беспроводное сетевое соединение:**

**DNS-суффикс подключения . . . . . : Home**

**Локальный IPv6-адрес канала . . . : fe80::595:5d8d:56f5:500c%41**

**IPv4-адрес. . . . . . . . . . . . : 192.168.0.206**

**Маска подсети . . . . . . . . . . : 255.255.255.0**

**Основной шлюз . . . . . . . . . : 192.168.0.10**

**Ethernet adapter Подключение по локальной сети:**

**Состояние среды. . . . . . . . : Среда передачи недоступна.**

**DNS-суффикс подключения . . . . . :**

**Туннельный адаптер isatap.{DDDBA9F8-664B-4B57-B2D0-93DE69D2FBE7}:**

**Состояние среды. . . . . . . . : Среда передачи недоступна.**

**DNS-суффикс подключения . . . . . :**

**Туннельный адаптер Подключение по локальной сети 3:**

**Состояние среды. . . . . . . . : Среда передачи недоступна.**

**DNS-суффикс подключения . . . . . :**

**Туннельный адаптер isatap.{73A36F53-6EE2-4F00-B90B-D11719558242}:**

**Состояние среды. . . . . . . . : Среда передачи недоступна.**

**DNS-суффикс подключения . . . . . :**

**Туннельный адаптер isatap.Home:**

**Состояние среды. . . . . . . . : Среда передачи недоступна.**

**DNS-суффикс подключения . . . . . :Home**

В примере выше команда выполнялась на ноутбуке. Связь с «внешним миром» осуществлена с помощью беспроводной сети. IP адрес интерфейса (Wi-Fi адаптер) ноутбука - **192.168.0.206**, маска - **255.255.255.0**, шлюз (IP-адрес сетевого интерфейса точки беспроводного доступа (WAP. WAP– это разновидность маршрутизатора, применяемая в беспроводных технологиях) - **192.168.0.10**. Интерфейсы ноутбука и маршрутизатора находятся в одной сети **192.168.0.0**. Сетевой кабель не подключен (по Ethernet адаптеру среда передачи недоступна).

**3.3 Утилита net view (нужны права администратора)**

Просматривает список доменов, компьютеров или общих ресурсов на данном компьютере. Синтаксис утилиты ***net view****:*

***net view*** *[\\компьютер | /domain[:*домен]];

*/domain[:*домен] - задает домен (рабочую группу), для которого выводится список компьютеров. Если параметр не указан, выводятся сведения обо всех доменах в сети;

Вызванная без параметров, утилита выводит список компьютеров в текущем домене (рабочей группе).

**3.4 Утилита ping**

Компьютеры и другие узлы сети помимо IP адресов имеют так называемые доменные адреса (символьные адреса). Такие адреса удобны пользователям сети, так как они легче запоминаются. К примеру доменный адрес **mаil.ru** запомнить намного проще чем его IP аналог в виде **94.100.180.70**. За соответствие доменных и IP адресов отвечает **DNS** (Domain Name Service) служба. Когда с компьютера исходит запрос на какой-либо сетевой ресурс по его доменному адресу, то DNS служба позволяет определить соответствующий этому ресурсу IP адрес.

Утилита ping (самая любимая утилита администраторов сети) проверяет соединения с удаленным компьютером или компьютерами. Эта команда доступна только после установки поддержки протокола TCP/IP.

Другими словами, утилита **ping** позволяет проверить доступность какого-либо удаленного узла по сети. С этой целью на указанный узел отправляется сообщение в виде запроса, и утилита переходит в режим ожидания прихода ответного сообщения. По истечении некоторого времени посылается повторное сообщение. По результатам обмена сообщениями выводится статистика о качестве связи между двумя узлами.

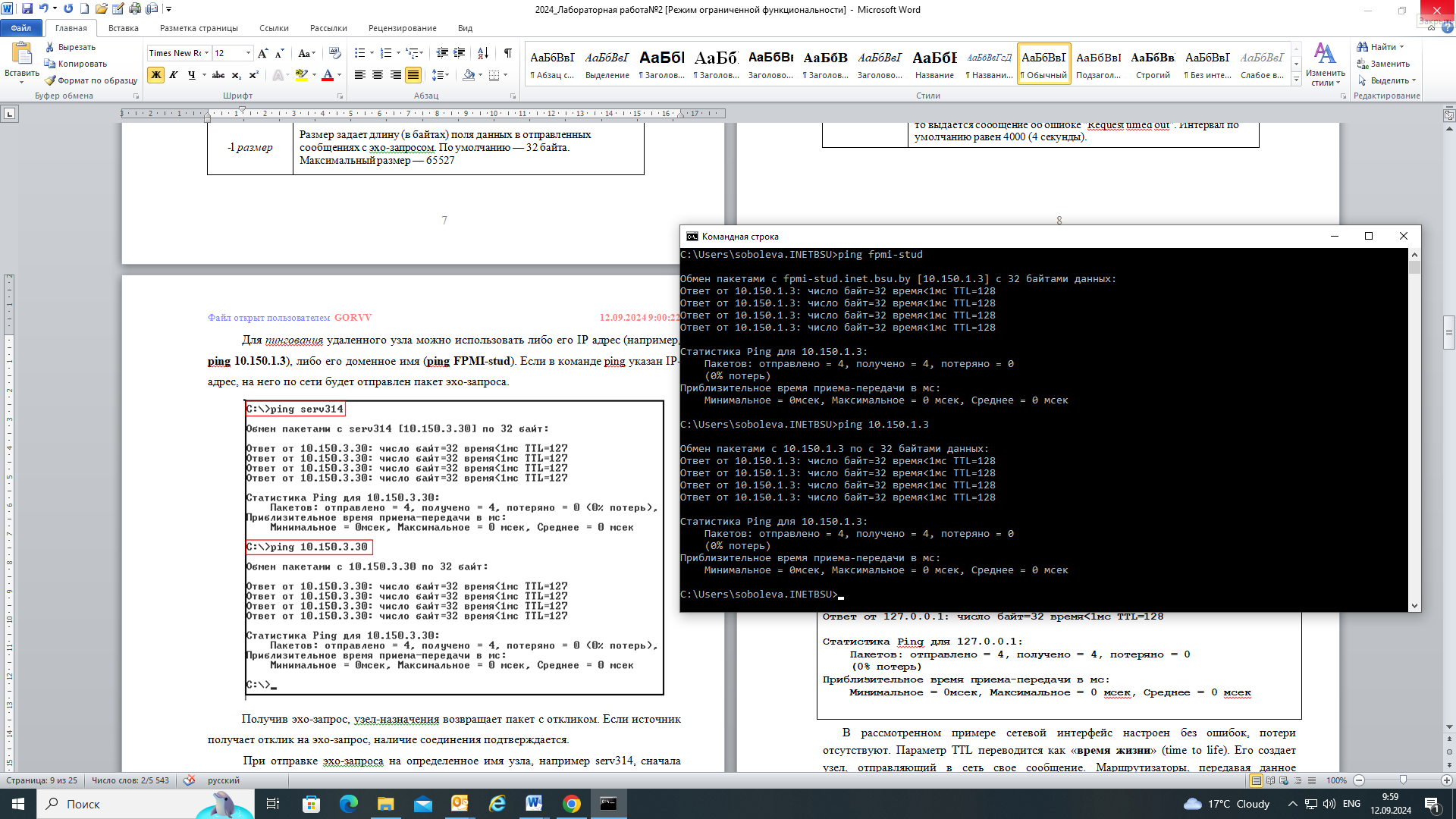
Синтаксис утилиты ***ping****:*

**ping [-t] [-a] [-n число] [-l размер] [-f] [-i TTL] [-v TOS] [-r число] [-s число] [[-j списокУзлов] | [-k списокУзлов]] [-w таймаут] конечноеИмя**

Параметры утилиты **рing** (ввиду ее популярности)для удобства изучения представлены в таблице

| Параметр  /? | Назначение параметра утилиты  Отображает справку в командной строке. |
| --- | --- |
| *имя\_хоста* | Задает точку назначения, идентифицированную IP-адресом или именем узла |
| -t | Отправка сообщений с эхо-запросом к точке назначения до тех пор, пока команда не будет прервана. |
| -a | Задает разрешение DNS имени по IP-адресу назначения. В случае успешного выполнения выводится имя соответствующего узла. |
| -n *число* | Задает число отправляемых сообщений с эхо-запросом. По умолчанию 4. |
| -l *размер* | Размер задает длину (в байтах) поля данных в отправленных сообщениях с эхо-запросом. По умолчанию — 32 байта. Максимальный размер — 65527 |
| -f | Задает отправку сообщений с эхо-запросом с установленным в 1 флагом «Don't Fragment» в IP-заголовке. Сообщения с эхо-запросом не фрагментируются маршрутизаторами на пути к месту назначения. Этот параметр полезен для устранения проблем, возникающих с максимальным блоком данных для канала (Maximum Transmission Unit) |
| -i *TTL* | Задает значение поля TTL в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию берется значение TTL, заданное по умолчанию для узла. Для узлов Windows XP это значение обычно равно 128. Максимальное значение TTL — 255. |
| -r *счетчик* | Задает параметр записи маршрута (Record Route) в IP-заголовке для записи пути, по которому проходит сообщение с эхо-запросом и соответствующее ему сообщение с эхо-ответом. Каждый переход в пути использует параметр записи маршрута. По возможности значение счетчика задается равным или большим, чем количество переходов между источником и местом назначения. Параметр счетчик имеет значение от 1 до 9. |
| -j *список\_узлов* | Указывает для сообщений с эхо-запросом использование параметра свободной маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в списке\_узлов. При свободной маршрутизации последовательные промежуточные точки назначения могут быть разделены одним или несколькими маршрутизаторами. Максимальное число адресов или имен в списке узлов — 9. Список узлов — это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами. |
| -k *список\_узлов* | Указывает для сообщений с эхо-запросом использование параметра строгой маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в списке\_узлов. При строгой маршрутизации следующая промежуточная точка назначения должна быть доступной напрямую (она должна быть соседней в интерфейсе маршрутизатора). Максимальное число адресов или имен в списке узлов равно 9. Список узлов — это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами. |
| -s *счетчик* | Указывает вариант штампа времени Интернета (Internet Timestamp) в заголовке IP для записи времени прибытия сообщения с эхо-запросом и соответствующего ему сообщения с эхо-ответом для каждого перехода. Параметр счетчик имеет значение от 1 до 4. |
| -v *тип* | Задает значение поля типа службы (TOS) в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию это значение равно 0. тип — это десятичное значение от 0 до 255 (см. Таблицу 1). |
| -w *интервал* | Определяет в миллисекундах время ожидания получения сообщения с эхо-ответом, которое соответствует сообщению с эхо-запросом. Если сообщение с эхо-ответом не получено в пределах заданного интервала, то выдается сообщение об ошибке "Request timed out". Интервал по умолчанию равен 4000 (4 секунды). |

Для *пингования* удаленного узла можно использовать либо его IP адрес (например, **ping 10.150.1.3**), либо его доменное имя (**ping FPMI-stud**). Если в команде ping указан IP-адрес, на него по сети будет отправлен пакет эхо-запроса.



Получив эхо-запрос, узел-назначения возвращает пакет с откликом. Если источник получает отклик на эхо-запрос, наличие соединения подтверждается.

При отправке эхо-запроса на определенное имя узла, например fpmi-stud, сначала отсылается пакет на DNS-сервер для преобразования имени в IP-адрес. После определения IP-адреса эхо-запрос пересылается на этот IP-адрес и обрабатывается обычным образом. Если эхо-запрос удается отправить на IP-адрес, но не на имя узла, может иметь место проблема с DNS.

Если эхо-запрос проходит на имя узла и на его IP-адрес, но пользователь не может работать с приложением, источником проблемы с большой вероятностью является приложение или узел назначения. Например, может быть недоступна запрошенная сетевая служба.

Если эхо-запрос не удается отправить ни одним из способов, проблема, скорее всего, локализована на промежуточном участке пути к узлу назначения. В этом случае рекомендуется отправить эхо-запрос на шлюз по умолчанию. Если эхо-запрос проходит на шлюз по умолчанию, проблема не связана с локальной сетью. Если эхо-запрос не проходит на шлюз по умолчанию, проблема имеет место в локальной сети.

В базовой форме команда ping обычно отправляет четыре эхо-запроса и ожидает отклика на каждый из них. Однако эту команду можно сделать более практичной с помощью дополнительных параметров.

**Пример***. Диагностика обратного адреса*.

Адрес 127.0.0.1 является служебным и узлам сети не назначается. Он предназначается для тестирования сетевых плат.

Тестирование адреса обратной связи командой ping 127.0.0.1 может предоставить информацию о корректности инсталляции сетевой платы и некорректности IP – адреса или маски подсети.

**ping 127.0.0.1**

**Обмен пакетами с 127.0.0.1 по с 32 байтами данных:**

**Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128**

**Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128**

**Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128**

**Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128**

**Статистика Ping для 127.0.0.1:**

**Пакетов: отправлено = 4, получено = 4, потеряно = 0**

**(0% потерь)**

**Приблизительное время приема-передачи в мс:**

**Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек**

В рассмотренном примере сетевой интерфейс настроен без ошибок, потери отсутствуют. Параметр TTL переводится как «**время жизни**» (time to life). Его создает узел, отправляющий в сеть свое сообщение. Маршрутизаторы, передавая данное сообщение из одной сети в другую, уменьшают TTL на единицу. Если на каком-то маршрутизаторе TTL будет уменьшено до нуля, то сообщение будет уничтожено. Маршрутизатор, удаливший из сети сообщение, извещает об этом отправителя, указывая свой адрес.

**Пример**. Второй вариант использования **ping** – это *проверка состояния тупиковой сети*, в которой находится сам узел. С этой целью пингуется основной шлюз:

**ping 192.168.0.10**

**Обмен пакетами с 192.168.0.10 по с 32 байтами данных:**

**Ответ от 192.168.0.10: число байт=32 время=11мс TTL=64**

**Ответ от 192.168.0.10: число байт=32 время=10мс TTL=64**

**Ответ от 192.168.0.10: число байт=32 время=9мс TTL=64**

**Ответ от 192.168.0.10: число байт=32 время=8мс TTL=64**

**Статистика Ping для 192.168.0.10:**

**Пакетов: отправлено = 4, получено = 4, потеряно = 0**

**(0% потерь)**

**Приблизительное время приема-передачи в мс:**

**Минимальное = 8мсек, Максимальное = 11 мсек, Среднее = 9 мсек**

В данном примере маршрутизатор доступен. Он в свои ответные сообщения помещает TTL (64) отличное от TTL (128) сетевого интерфейса компьютера.

**Пример.** *Проверка доступности удаленного хоста*

Как правило, применяются доменные адреса:

**ping esstu.ru**

**Обмен пакетами с esstu.ru [212.0.68.2] с 32 байтами данных:**

**Ответ от 212.0.68.2: число байт=32 время=7мс TTL=57**

**Ответ от 212.0.68.2: число байт=32 время=8мс TTL=57**

**Ответ от 212.0.68.2: число байт=32 время=10мс TTL=57**

**Ответ от 212.0.68.2: число байт=32 время=7мс TTL=57**

**Статистика Ping для 212.0.68.2:**

**Пакетов: отправлено = 4, получено = 4, потеряно = 0**

**(0% потерь)**

**Приблизительное время приема-передачи в мс:**

**Минимальное = 7мсек, Максимальное = 10 мсек, Среднее = 8 мсек**

Удаленный узел доступен. В данном случае мы видим, что DNS служба определила IP адрес узла в виде **212 .0. 68 .2**.

**Пример.** Утилита имеет несколько опций, из которых рассмотрим лишь одну: **-i**, позволяющую задать значение TTL:

**ping -i 1 esstu.ru**

**Обмен пакетами с esstu.ru [212.0.68.2] с 32 байтами данных:**

**Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.**

**Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.**

**Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.**

**Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.**

**Статистика Ping для 212.0.68.2:**

**Пакетов: отправлено = 4, получено = 4, потеряно = 0**

**(0% потерь)**

Здесь TTL был принят равным 1 и сообщение было уничтожено на   
шлюзе (**192. 168 . 0. 10**) . В следующем примере TTL=2

**ping -i 2 esstu.ru**

**Обмен пакетами с esstu.ru [212.0.68.2] с 32 байтами данных:**

**Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.**

**Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.**

**Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.**

**Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.**

**212.0.68.2**

**Статистика Ping для 212.0.68.2:**

**Пакетов: отправлено = 4, получено = 4, потеряно = 0**

**(0% потерь)**

На этот раз сообщение дошло до маршрутизатора (**192. 168. 1. 1**), который был вторым за шлюзом. Так, постепенно меняя значение TTL, можно получить список всех маршрутизаторов, находящихся между компьютером и удаленным узлом **212. 0. 68. 2**.

**3.5 Утилита netstat**

Когда мы говорим «компьютеры обмениваются сообщениями», то это не совсем точное утверждение. На самом деле обмен происходит между сетевыми приложениями. В оперативной памяти компьютера одновременно могут находиться и выполняться несколько программ, получающих сообщения из сети или отправляющих их в сеть.

***Как же сообщения, приходящие из сети в компьютер, распределяются между этими приложениями?*** На этот случай в сообщениях предусмотрены дополнительные адреса, называемые ***портами*.** Здесь уместно привести аналогию с обычной почтовой корреспонденцией. Для того чтобы письмо было доставлено в многоквартирный дом (компьютер), на конверте письма указывается номер дома (IP адрес компьютера). Затем письма необходимо разложить по почтовым ящикам согласно номерам квартир. Номер квартиры, присутствующий на конверте письма, и есть аналог портов. Далее жильцы (т.е. сетевые приложения) забирают эти письма (сообщения).

Когда приложение хочет обменяться сообщениями с другим удаленным приложением, оно должно знать не только IP адрес компьютера данного приложения, но и номер порта, который это приложение использует. Эта связка из двух адресов (IP адрес и порт) называется ***сокетом***. Как определяется номер порта, которое использует удаленное приложение - эта будет рассмотрено позже. Оба приложения устанавливают между собой соединение, используя два сокета. Сокеты можно условно представить в виде двух разъемов (розеток), соединенных между собой неким виртуальным каналом связи. Когда одно приложение «помещает» в сокет свое сообщение, то оно доставляется на другой конец канала - на второй сокет, и попадает, таким образом, другому приложению.

В некоторых случаях требуется определить, какие TCP-соединения открыты и действуют на сетевом узле. Проверить состояние этих соединений помогает утилита – **netstat**. Команда netstat перечисляет используемые протоколы, локальные адреса и номера портов, адрес и номер порта на удаленном узле и сообщает состояние соединений.

*Необъяснимые TCP-соединения могут представлять значительную угрозу безопасности. Они свидетельствуют о наличии посторонних подключений к локальному узлу*. Кроме того, лишние TCP-соединения создают нагрузку на системные ресурсы и способны существенно замедлить работу узла. С помощью команды netstat можно получить информацию об открытых соединениях с узлом в случае заметного ухудшения производительности.

Синтаксис утилиты ***netstat****:*

***netstat*** *[-a] [-e] [-n] [-s] [-p протокол] [-r] [интервал],*

где

*-a* - выводит все подключения и сетевые порты. Подключения сервера обычно не выводятся;

*-e* - выводит статистику Ethernet. Возможна комбинация с ключом *-s;*

*-n* - выводит адреса и номера портов в шестнадцатеричном формате (а не имена);

*-s*- выводит статистику для каждого протокола. По умолчанию выводится статистика для TCP, UDP, ICMP (Internet Control Message Protocol) и IP. Ключ *-p* может быть использован для указания подмножества стандартных протоколов;

*-p протокол* - выводит соединения для протокола, заданного параметром. Параметр может иметь значения *tcp* или *udp.* Если используется с ключом *-s* для вывода статистики по отдельным протоколам, то параметр может принимать значения *tcp, udp, icmp* или *ip; -r* - выводит таблицу маршрутизации;

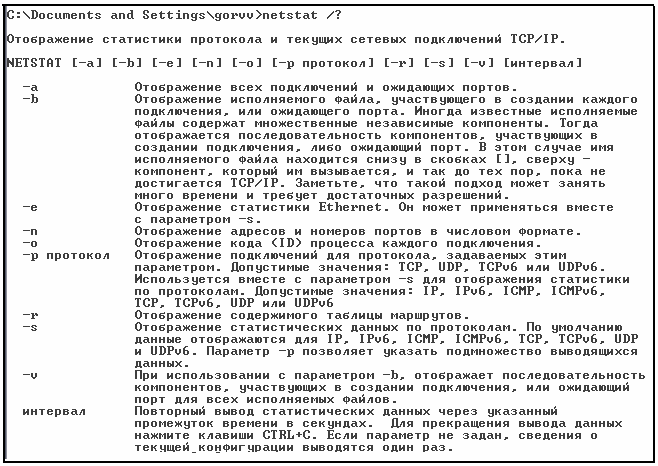
*интервал* - обновляет выведенную статистику с заданным в секундах интервалом. Нажатие клавиш CTRL+C останавливает обновление статистики. Если этот параметр пропущен, *netstat* выводит сведения о текущей конфигурации один раз.

**Пример**. *Получить список сокетов*

Команда **netstat** позволяет получить список сокетов. Ниже приведен вывод (приведен не весь вывод), полученный c использованием опций **a**, **n**, и **o** ( подсказку по опциям можно получить, введя команду netstat /? или hing \help. См. рисунки ниже). Для других утилит достаточно запустить утилиту без параметров.

Данный вывод показывает, что сокеты обозначаются в виде пары **IP\_адрес : порт**   
(с двоеточием между адресами). Например, **192. 168. 0. 206 : 54842**. Существующий виртуальный канал связи обозначен парой сокетов. Например, **192. 168. 0 .206 : 54842** и  
 **64. 4. 23. 171 : 40013**. Первый сокет открыт на компьютере, другой на удаленном узле. Адрес в виде **0. 0. 0. 0** означает любые IP адреса. Если в качестве номера порта присутствует **0**, то это означает любые значения портов. В колонке "Состояние" отображается состояние соединения (приведены не все варианты):

* LISTENING – ожидание подключения;
* ESTABLISHED – соединение установлено, идет обмен сообщениями;
* TIME\_WAIT– время ответа превышено.



**netstat -ano**

**Активные подключения**

**Имя Локальный адрес Внешний адрес Состояние PID**

**TCP 0.0.0.0:80 0.0.0.0:0 LISTENING 2944**

**TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 892**

**TCP 0.0.0.0:443 0.0.0.0:0 LISTENING 2944**

**TCP 0.0.0.0:445 0.0.0.0:0 LISTENING 4**

**TCP 0.0.0.0:26143 0.0.0.0:0 LISTENING 4**

**TCP 0.0.0.0:45662 0.0.0.0:0 LISTENING 2920**

**TCP 0.0.0.0:49152 0.0.0.0:0 LISTENING 576**

**TCP 0.0.0.0:49153 0.0.0.0:0 LISTENING 1020**

**TCP 0.0.0.0:49154 0.0.0.0:0 LISTENING 724**

**TCP 0.0.0.0:49155 0.0.0.0:0 LISTENING 660**

**TCP 0.0.0.0:49160 0.0.0.0:0 LISTENING 640**

**TCP 0.0.0.0:61741 0.0.0.0:0 LISTENING 2944**

**TCP 127.0.0.1:5939 0.0.0.0:0 LISTENING 2580**

**TCP 127.0.0.1:10000 0.0.0.0:0 LISTENING 2920**

**TCP 127.0.0.1:49156 127.0.0.1:49157 ESTABLISHED 2952**

**TCP 127.0.0.1:49157 127.0.0.1:49156 ESTABLISHED 2952**

**TCP 127.0.0.1:49158 127.0.0.1:49159 ESTABLISHED 2952**

**TCP 127.0.0.1:49159 127.0.0.1:49158 ESTABLISHED 2952**

**TCP 192.168.0.206:139 0.0.0.0:0 LISTENING 4**

**TCP 192.168.0.206:54842 64.4.23.171:40013 ESTABLISHED 2944**

**TCP 192.168.0.206:54844 157.56.53.42:12350 ESTABLISHED 2944**

**TCP 192.168.0.206:54845 173.252.121.3:5222 ESTABLISHED 2944**

**TCP 192.168.0.206:54850 191.235.188.99:443 ESTABLISHED 2944**

**TCP 192.168.0.206:54893 64.4.61.132:443 ESTABLISHED 2944**

**TCP 192.168.0.206:54919 157.56.194.7:443 ESTABLISHED 2944**

**TCP 192.168.0.206:55125 185.39.80.24:80 ESTABLISHED 3684**

**TCP 192.168.0.206:57762 137.116.224.167:443 TIME\_WAIT 0**

**TCP 192.168.0.206:57770 81.19.104.81:443 TIME\_WAIT 0**

**TCP 192.168.0.206:57792 192.168.0.10:1780 TIME\_WAIT 0**

**TCP 192.168.0.206:57793 192.168.0.10:1780 TIME\_WAIT 0**

**TCP 192.168.0.206:57822 81.19.104.81:443 TIME\_WAIT 0**

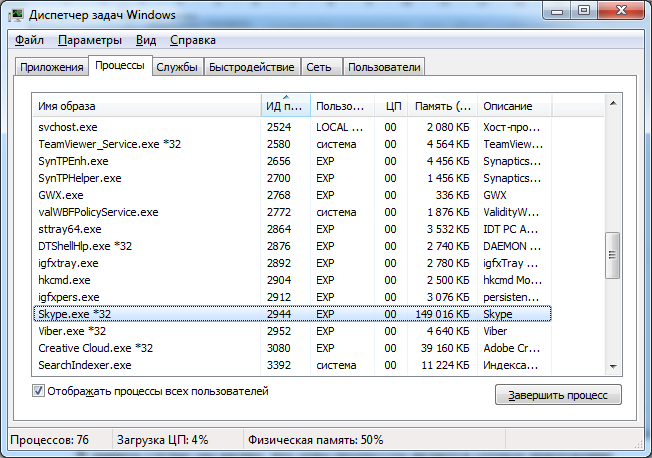
**TCP 192.168.0.206:57845 176.119.71.119:62348 ESTABLISHED 2920**

**<дальнейший вывод был пропущен>**

Первый тип состояния (LISTENING) означает, что сетевое приложение ждет установления соединения по определенному порту. Например, сокет **0.0.0.0:443** означает, что какое-то удаленное приложение может отправить на компьютер сообщение на порт 443 с целью установить виртуальное соединение.

В последней колонке (**PID**) выводятся номера процессов. Под процессами будем для упрощения понимать приложения. Из вывода мы видим, что процесс 2944 ждет подключения по портам 80, 443 и 61741. Как выше было сказано, какая-то программа с другого узла может отправить запрос на установление соединения с процессом 2944. Такая программа своё сообщение может адресовать на любой из указанных трех портов.

Чтобы выяснить какая программа запущена под видом процесса 2944, вызовем диспетчер задач (Ctrl+Alt+Delete). В окне диспетчера перейдем на вкладку **Процессы** и войдем в меню **Вид**. Далее выберем строчку **Выбрать столбцы** и активируем чекбокс   
**ИД процесса (PID)**. Щелкнем по **OK**. Затем отсортируем таблицу по столбцу **ИД процесса (PID)**, щелкнув по его названию. Находим запись, соответствующую процессу 2944.



В данном случае мы видим, что этим процессом является сетевое приложение Skype. Приведенный выше вывод **netstat** показывает, что данная программа поддерживает связь с шестью удаленными skype приложениями других пользователей. Для каждого соединения был создан отдельный сокет.

**3.6 Утилита tracert**

Эта утилита, последовательно применяя пинг с увеличивающимся параметром TTL, позволяет получить список промежуточных маршрутизаторов.

Утилита tracert использует протокол ICMP для определения маршрута прохождения пакета. При отсылке tracert устанавливает значение TTL последовательно от 1 до 30. Каждый маршрутизатор, через который проходит пакет на пути к назначенному хосту, уменьшает значение TTL на единицу. С помощью TTL происходит предотвращение зацикливания пакета в "петлях" маршрутизации, иначе "заблудившиеся" пакеты окончательно перегрузили бы сеть. Однако, при выходе маршрутизатора или линии связи из строя требуется несколько дополнительных переходов для понимания, что данный маршрут потерян и его необходимо обойти. Чтобы предотвратить потерю датаграммы, поле TTL устанавливается на максимальную величину.

Когда маршрутизатор получает IP-датаграмму с TTL, равным 0 или 1, он уничтожает ее и посылает хосту, который ее отправил, ICMP-сообщение "время истекло" (time exceeded). Принцип работы tracert заключается в том, что IP-датаграмма, содержащая это ICMP-сообщение, имеет в качестве адреса источника IP-адрес маршрутизатора.

Теперь легко понять, как работает tracert. На хост назначения отправляется IP- датаграмма с TTL, равным единице. Первый маршрутизатор, который должен обработать датаграмму, уничтожает ее (так как TTL равно 1) и отправляет ICMP-сообщение об истечении времени (time exceeded). Таким образом определяется первый маршрутизатор в маршруте. Затем tracert отправляет датаграмму с TTL, равным 2, что позволяет получить IP-адрес второго маршрутизатора. Так продолжается до тех пор, пока датаграмма не достигнет хоста назначения. Утилита tracert может посылать в качестве такой датаграммы UDP-сообщение с номером порта, который заведомо не будет обработан приложением (порт выше 30000), поэтому хост назначения ответит "порт недоступен" (port unreachable). При получении такого ответа делается вывод, что удаленный хост работает корректно. В противном случае максимального значения TTL (по умолчанию 30) не хватило для того, чтобы его достигнуть.

Синтаксис утилиты ***tracert***:

***tracert*** *[-d] [-h макс\_узл] [-j список компьютеров] [-w интервал] точка назн,*

где *-d* - отменяет разрешение имен компьютеров в их адреса;

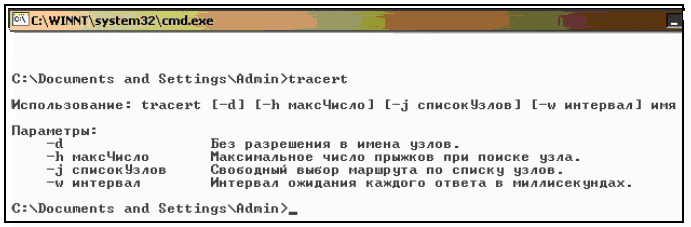
*-h макс\_узл* - задает максимальное количество ретрансляций, используемых при поиске точки назначения;

*-j список компьютеров* - задает список\_компьютеров для свободной маршрутизации;

*-w интервал* - задает интервал в миллисекундах, в течение которого будет ожидаться ответ;

*точка назн* - указывает имя конечного компьютера.

В базовой форме команда tracert прослеживает не более 30 участков маршрута от источника к адресату. При превышении этого числа участков она сообщает о недоступности адресата. Число участков настраивается параметром -h. Также доступны другие модификаторы, приведенные в числе параметров на рисунке.

****

**Пример** *Трассировка маршрута к esstu.ru***,** применяя утилиту *tracert:*

**tracert esstu.ru**

**Трассировка маршрута к esstu.ru [212.0.68.2]**

**с максимальным числом прыжков 30:**

**1 1 ms 1 ms 1 msWRT54GL [192.168.0.10]**

**2 3 ms 2 ms 2 ms 192.168.1.1**

**3 8 ms 4 ms 5 ms ULND-BRAS3.sib.ip.rostelecom.ru [213.228.116.203]**

**4 17 ms 5 ms 8 ms 213.228.114.27**

**5 8 ms 3 ms 3 ms core-gi-0-2.burnet.ru [212.0.64.90]**

**6 7 ms 5 ms 6 ms ws-70-71.burnet.ru [212.0.70.71]**

**7 8 ms 4 ms 3 ms 86.110.127.129**

**8 8 ms 4 ms 4 ms 212.0.68.2**

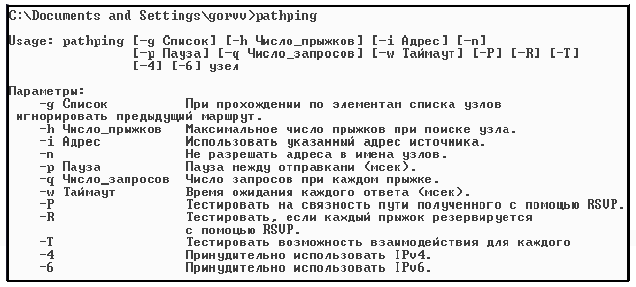
**Трассировка завершена.**

Между двумя узлами в данном случае находится 7 маршрутизаторов.

**3.7. Утилита pathping**

Утилита **pathping** сочетает в себе черты команд **ping** и **tracert**, позволяя получить дополнительную информацию, которую не обеспечивают две последние. Команда определяет процент потерь сообщений на всех переходах, **выявляя** самые **медленные** и **ненадежные** участки маршрута. Заметим, что команда выполняется относительно долго.

Скриншот синтаксиса утилиты ***pathping***



**Пример.**

**pathping esstu.ru**

**Трассировка маршрута к esstu.ru [212.0.68.2]**

**с максимальным числом прыжков 30:**

**0 EXPHOME.Home [192.168.0.206]**

**1 WRT54GL [192.168.0.10]**

**2 Broadcom.Home [192.168.1.1]**

**3 ULND-BRAS3.sib.ip.rostelecom.ru [213.228.116.203]**

**4 213.228.114.27**

**5 core-gi-0-2.burnet.ru [212.0.64.90]**

**6 ws-70-71.burnet.ru [212.0.70.71]**

**7 86.110.127.129**

**8 212.0.68.2**

**Подсчет статистики за: 200 сек. ...**

**Исходный узел Маршрутный узел**

**Прыжок RTT Утер./Отпр. % Утер./Отпр. % Адрес**

**0 EXPHOME.Home [192.168.0.206]**

**0/ 100 = 0% |**

**1 6мс 0/ 100 = 0% 0/ 100 = 0% WRT54GL [192.168.0.10]**

**0/ 100 = 0% |**

**2 8мс 0/ 100 = 0% 0/ 100 = 0% Broadcom.Home [192.168.1.1]**

**0/ 100 = 0% |**

**3 11мс 0/ 100 = 0% 0/ 100 = 0% ULND-BRAS3.sib.ip.rostelecom.ru [213.228.116.203]**

**0/ 100 = 0% |**

**4 11мс 0/ 100 = 0% 0/ 100 = 0% 213.228.114.27**

**0/ 100 = 0% |**

**5 10мс 0/ 100 = 0% 0/ 100 = 0% core-gi-0-2.burnet.ru [212.0.64.90]**

**0/ 100 = 0% |**

**6 15мс 0/ 100 = 0% 0/ 100 = 0% ws-70-71.burnet.ru [212.0.70.71]**

**0/ 100 = 0% |**

**7 --- 100/ 100 =100% 100/ 100 =100% 86.110.127.129**

**0/ 100 = 0% |**

**8 12мс 0/ 100 = 0% 0/ 100 = 0% 212.0.68.2**

**Трассировка завершена.**

В настройках некоторых маршрутизаторов может стоять запрет на выдачу ответа на пришедший пинг. В данном примере маршрутизатор с подобной настройкой имеет адрес **86.110.127.129**. Попытка отправить пинг на этот адрес убеждает нас в справедливости этого утверждения:

**ping 86.110.127.129**

**Обмен пакетами с 86.110.127.129 по с 32 байтами данных:**

**Превышен интервал ожидания для запроса.**

**Превышен интервал ожидания для запроса.**

**Превышен интервал ожидания для запроса.**

**Превышен интервал ожидания для запроса.**

**Статистика Ping для 86.110.127.129:**

**Пакетов: отправлено = 4, получено = 0, потеряно = 4**

**(100% потерь)**

**3.8**  **Утилита arp**

Сетевые интерфейсы, такие как Ethernet, Wi-Fi и WiMAX, имеют вшитые в их микросхемы адреса. Пример подобного адреса: **70-F3-95-A6-FE-0C**. Эти адреса, называемые *аппаратными, физическими* или***MAC*** (Media Access Control — управление доступом к среде), должны добавляться к сообщениям, прежде чем они будут переданы через сеть. Не все сети используют такие адреса, но в тупиковых они, как правило, применяются. Узел, собирающийся отправить сообщение другому узлу (оба узла находятся в одной и той же локальной сети!), должен предварительно узнать MAC адрес получателя сообщения. Для решения данной проблемы узел применяет технологию ARP (Address Resolution Protocol — протокол определения адреса), отправляя запрос другим узлам своей локальной сети. Данный ARP запрос содержит IP адрес получателя. Из всех узлов, получивших данный запрос, отвечает лишь тот, у кого требуемый IP адрес. В своем ответе (ARP отклике) тот узел сообщает свой MAC адрес. И лишь после этого первый узел ему сможет отправить свое сообщение. В тупиковых сетях компьютеры чаще всего отправляют свои сообщения маршрутизатору и, следовательно, в своих ARP запросах они указывают адрес основного шлюза. Для уменьшения ARP трафика компьютеры хранят в своей памяти таблицу с IP и MAC адресами тех устройств, с которыми они в последнее время обменивались сообщениями. Таким образом, В оперативной памяти компьютера находится ARP-таблица. В ней содержатся MAC-адрес удаленной машины и соответствующий ему IP-адрес. Для просмотра этой таблицы используется команда **arp**.

**Пример**,  Вывести все известные MAC- адреса.

**arp -a**

**Интерфейс: 192.168.0.206 --- 0xe**

**адрес в Интернете Физический адрес Тип**

**192.168.0.10 20-aa-4b-2a-d5-21 динамический**

**192.168.0.255 ff-ff-ff-ff-ff-ff статический**

**224.0.0.22 01-00-5e-00-00-16 статический**

**224.0.0.252 01-00-5e-00-00-fc статический**

**239.192.152.143 01-00-5e-40-98-8f статический**

**239.255.255.250 01-00-5e-7f-ff-fa статический**

В данном случае мы видим, что у основного шлюза (сетевой интерфейс маршрутизатора - **192.168.0.10**) MAC адрес равен **20-aa-4b-2a-d5-21**.

Существует два типа записей в ARP-таблице – *статический* и *динамический*. Статическая запись вносится вручную и существует до тех пор, пока вручную же не будет удалена, или компьютер (маршрутизатор) не будет перезагружен.

Динамическая запись появляется при попытке отправить сообщение на IP- адрес, для которого неизвестен MAC-адрес. В этом случае формируется ARP-запрос, который позволяет этот адрес определить, после чего соответствующая динамическая запись добавляется в ARP-таблицу. Храниться там она будет не постоянно. После определенного времени она будет автоматически удалена, если к данному IP-адресу не было обращений. Задержка на получение MAC-адреса составляет порядка нескольких миллисекунд, так что для пользователя это будет практически незаметно, зато появляется возможность отследить изменения в конфигурации сети (в соответствии IP- и MAC-адресов).

Замечание

Аппаратные адреса в некоторых версиях ОС можно вывести утилитой getmac.

**3.9 Утилита net send (**уже устарела)

Отправка сообщения другому пользователю, компьютеру или псевдониму в сети. Служба сообщений должна быть запущена на компьютере для получения сообщений. Синтаксис утилиты **net send**:

**net send** {имя | \* | /domain[:имя] | /users} сообщение,

где

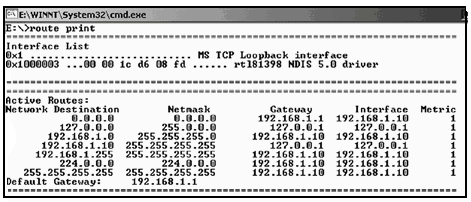
*имя* - указывает имя пользователя, имя компьютера или псевдоним, которому будет отправлено сообщение. Если имя компьютера содержит пробелы, оно должно быть заключено в кавычки (" "). Длинные имена пользователей, введенные в формате NetBIOS, могут привести к возникновению исключительных ситуаций. Имена NetBIOS ограничены 16 символами;

\* - отправляет сообщение всем членам группы;

*/domain[:имя]* - отправляет сообщение всем именам в домене компьюте-ра. Если параметр имя указан, сообщение будет отправлено всем именам заданного домена или рабочей группы;

*/users* - отправляет сообщение всем пользователям, подключенным к серверу; сообщение - указывает текст сообщения.

**3.10. Утилита Route**

Для просмотра и редактирования таблицы маршрутов используется утилита **route**. Типичный пример таблицы маршрутизации на персональном компьютере: для ОС Windows: кoute print 

Заметим, что в первой части таблицы на рисунке выше мы увидим список аппаратных адресов интерфейсов. Во второй части - в таблице маршрутизации указывается сеть, маска сети, маршрутизатор, через который доступна эта сеть, интерфейс и метрика маршрута. Из приведенной таблицы видно, что маршрут по умолчанию доступен через маршрутизатор 192.168.1.1. Сеть 192.168.1.0 с маской 255.255.255.0 является локальной сетью.

При добавлении маршрута можно использовать следующую команду.

route ADD 157.0.0.0 MASK 255.0.0.0 157.55.80.1

Здесь: 157.0.0.0 – удаленная сеть, 255.0.0.0 – маска удаленной сети, 157.55.80.1 – маршрутизатор, через который доступна эта сеть.

Примерно такой же синтаксис используется при удалении маршрута:

route DELETE 157.0.0.0.

**4. Задания и вопросы для выполнения лабораторной работы № 2**

**4.1 Варианты ссылок**

| **Порядковый номер ссылки** | **Ссылки** |
| --- | --- |
| **1** | **pogoda.by** |
| **2** | **oma.by** |
| **3** | **onliner.by** |
| **4** | **rambler.ru** |
| **5** | **mail.ru** |
| **6** | **beltelecom.by** |
| **7** | **basnet.by** |
| **8** | **a1.by** |
| **9** | **mts.by** |
| **10** | **google.by** |
| **11** | **abw.by** |
| **12** | **kufar.by** |
| **13** | **url.by** |
| **14** | **iptel.by** |
| **15** | **rabota.by** |

**4.2 Варианты заданий**

| **Номер варианта** | **Порядковый номер ссылки** | | **Номер варианта** | **Порядковый номер ссылки** | |
| --- | --- | --- | --- | --- | --- |
| **1** | **8** | **6** | **15** | **14** | **3** |
| **2** | **5** | **3** | **16** | **5** | **1** |
| **3** | **7** | **1** | **17** | **8** | **10** |
| **4** | **6** | **10** | **18** | **11** | **6** |
| **5** | **12** | **7** | **19** | **1** | **13** |
| **6** | **2** | **3** | **20** | **2** | **7** |
| **7** | **2** | **9** | **21** | **4** | **5** |
| **8** | **14** | **15** | **22** | **14** | **10** |
| **9** | **13** | **5** | **23** | **9** | **12** |
| **10** | **9** | **1** | **24** | **15** | **4** |
| **11** | **15** | **12** | **25** | **3** | **15** |
| **12** | **3** | **15** | **26** | **13** | **7** |
| **13** | **10** | **7** | **27** | **7** | **11** |
| **14** | **6** | **15** | **28** | **1** | **2** |

**4.3 Задания для отчета**

**4.3.1 *Задание 1*. Получение справочной информации по командам**

* Выведите на экран справочную информацию по утилитам arp, ipconfig, nbtstat, netstat, nslookup, route, ping, tracert, hostname. Для этого в командной строке введите имя утилиты без параметров или с /?.
* Изучите ключи, используемые при запуске утилит.
* В отчет приложите скриншот получения справочной информации об одной из утилит на ваш выбор

**4.3.2. *Задание 2.* Получение имени хоста**.

Выведите на экран и запишите имя локального хоста (желательно и личного компьютера), на котором вы работаете с помощью команды (какой?).

**4.3.3. *Задание 3.* Изучение утилиты ipconfig**

Проверьте конфигурацию TCP/IP с помощью утилиты ipconfig. Утилиту выполните на компьютере в компьютерном классе ФПМИ и на личном ноутбуке. Заполните соответственно таблицу. Обратите внимание на значения в последних двух справа столбцах. Проанализируйте отличия в заполненных столбцах:

|  | **ПК аудитории** | **Личный ноутбук в сети БГУ** |
| --- | --- | --- |
| Имя компьютера |  |  |
| Описание адаптера |  |  |
| Физический адрес сетевого адаптера |  |  |
| IP-адрес |  |  |
| Маска подсети |  |  |
| Основной шлюз |  |  |
| Используется ли DHCP (адрес DHCP-сервера) |  |  |
| Адрес DNS-сервера |  |  |
| Адрес WINS-сервера |  |  |

**4.3.4. *Задание 4*. Тестирование связи с помощью утилиты ping**.   
 Проверьте правильность установки и конфигурирования TCP/IP на локальном компьютере. С помощью команды ping проверьте перечисленные ниже адреса и для каждого из них отметьте TTL (Time To Live) и время отклика. Попробуйте увеличить время отклика.

10.150.1.3, 10.150.1.34, 10.0.0.20

Задайте различную длину посылаемых пакетов (можно только на любом одном из примеров выписать результат для отчета).

Выпишите ответы на следующие задания:

* Определите DNS-имя любого соседнего компьютера по его IP-адресу.
* Проверьте доступность сайта поисковой системы Yandex в сети Internet через две точки ya.ru и yandex.ru , а также узнайте их IP-адреса.
* Пропинговать сетевой интерфейс локального компьютера.
* Отправить на адрес согласно вашему варианту n сообщений (n- номер варианта) с эхо-запросом, каждое из которых имеет поле данных из 1000 байт.
* Что такое TTL

**4.3.5. *Задание 5.* Утилита Tracert. Определение пути IP-пакета**

* Определите список маршрутизаторов на пути следования пакетов от локального компьютера до адресов согласно вашему варианту без преобразования IP-адресов в имена DNS. (Выпишите команду с помощью которой это можно выполнить.)
* С помощью команды tracert проверьте, через какие промежуточные узлы идет сигнал. Выпишите ***первые три*** и ***последние два*** промежуточных узла на каждый из ваших вариантов заданий.
* Можно ли утилитой ***tracert*** задать максимальное число ретрансляций, если можно, то выпишите как.

**4.3.6. *Задание 6*. Просмотр ARP-кэша**

* С помощью утилиты **arp** просмотрите и выпишите ARP-таблицу локального компьютера (несколько записей).
* Прокомментируйте какая информация хранится в ARP- таблице.

**4.3.7. *Задание 7*. Утилита netstat.** Получение информации о текущих сетевых соединениях и протоколах стека TCP/IP.

* Получите список активных TCP-соединений локального компьютера. (Выпишите команду с помощью которой это можно выполнить.)
* Получите список активных TCP-соединений локального компьютера без преобразования IP-адресов в символьные имена DNS. (Выпишите команду с помощью которой это можно выполнить.)
* Какой результат выдаст утилита netstat с параметрами -a -s -r (**три параметра одновременно**)? Поясните полученный результат.

**4.3.8. *Задание 8*.** Получите таблицу маршрутизации локального компьютера. Как это можно сделать.

Подготовить электронный вариант отчета с заданиями 1-9.

***Замечание.***

1. Отчет подготовить в формате **.doc** и положить на образовательный портал ФПМИ. Имя файла должно быть задано по правилу: **12\_Лаб\_02\_Иванов** (здесь студент **Иванов** из **12**-ой группы создал отчет по лабораторной работе №**02**). Если отчетные документы состоят из нескольких файлов, то по аналогичному правилу создайте папку, в которую положите отчетные документы (отчет и файлы моделей КС (.pkt - файлы).

В дальнейшем в имени будете менять только номер лабораторной работы. Прошу четко соблюдать эти правила.