Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**Лабораторная работа №1**

по дисциплине: «Сети и телекоммуникации».

Выполнил:

студент 4 курса, гр. ИВТАПбд-41

Кондратьев Павел Сергеевич.

Проверил:

преподаватель кафедры ВТ

Мартынов Антон Иванович.

г. Ульяновск, 2019

**1) Задание:**

Необходимо научиться пользоваться вышеперечисленными утилитами. Типовыми

заданиями являются:

* Просмотр и анализ настроек стека протоколов TCP/IP с использованием утилиты ipconfig.
* Тестирование сетевого соединения утилитой ping (конечный адрес может быть, напр., IP-адрес местного Proxy-сервера, адрес одной из машин в сети, доменное имя некоторого узла в Сети и др.).
* Прослеживание маршрута пакетов до заданного узла утилитой tracert.
* Отображение и изменение таблицы хранения соответствия МАС- и IP адресов с помощью утилиты arp.
* Просмотр текущих сетевых соединений, портов, TCP/IP-статистики утилитой netstat.
* Отображение и изменение таблицы маршрутов утилитой route (необходимо вывести все маршруты, добавить свой маршрут, изменить его, а затем удалить).

**2) Краткие теоретические сведения:**

**1. Утилита ping**

Утилита ping тестирует сетевое соединение путем посылки ICMP пакетов типа 8 (запрос эха), на которые получатель отвечает ICMP-пакетом типа 0 (эхо-ответ). С помощью этой утилиты удобно проверять наличие пути до заданного узла и определять временные характеристики этого пути.

Утилите ping достаточно указать IP-адрес или DNS-имя, однако имеется ряд параметров, позволяющих более тонко управлять ее работой. Утилита ping выводит результат каждого запроса/ответа на отдельной строке, а перед завершением работы выдает статистику: минимальное, максимальное и среднее время передачи пакета, количество и долю потерянных пакетов.

Фактически ping является ‘рабочей лошадкой’ при тестировании сетевых соединений.

**Формальный синтаксис:**

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

**2. Утилита pathping**

Утилита pathping фактически совмещает функциональность утилит ping и tracert и выполняется в две фазы: сначала, подобно tracert, собирается и выводится маршрут до заданного узла (только IP-адреса и имена) и затем, подобно ping, в течение некоторого времени (чем дольше выполнялась трассировка, тем больше будет это время) собирается статистика времен передачи пакетов, количеств и относительных долей потерянных пакетов для каждого из промежуточных маршрутизаторов (а не только для заданного узла, как ping).

**Формальный синтаксис:**

pathping [-g Список] [-h Число\_прыжков] [-i Адрес] [-n] [-p Пауза] [-q Число\_запросов] [-w Таймаут] [-P] [-R] [-T] [-4] [-6] узел

Наиболее полезен результат работы второй фазы утилиты pathping - он наглядно показывает, на каком из маршрутизаторов имеются проблемы с передачей пакетов.

**3. Утилита агр**

Утилита агр дает возможность просматривать и изменять ARP-таблицу, в которой хранятся пары ‘МАС-адрес - IP-адрес’ для тех узлов, с которыми в недавнем происходил обмен данными. Эта таблица формируется автоматически при работе сетевого узла, но администратор сети может вносить в нее записи вручную.

**Формальный синтаксис:**

**ARP –s inet\_addr eth\_addr [if\_addr]**

или

**ARP –d inet\_addr [if\_affr]**

или

**ARP –a [inet\_addr] [-N if\_addr]**

* Параметр if\_addr задает номер интерфейса.
* Параметр ‘-а’ позволяет вывести всю ARP-таблицу на экран.
* Параметр ‘-а <IP-адрес>’ запрашивает вывод записи об узле с заданным адресом на экран.
* Параметр ‘-S <IP-адрес> <MAC-адрес>’ позволяет добавить запись об узле с заданными адресами в ARP-таблицу.
* Параметр ‘-d <IP-адрес>’ служит для удаления записи об узле с заданным адресом из ARP-таблицы.
* Параметр ‘-d \*’ очищает ARP-таблицу.

**4. Утилита ipconfig**

Утилита ipconfig отображает и настраивает настройки протоколов TCP/IP. Без дополнительных параметров выводится IP-адрес, маска подсети и шлюз по умолчанию для всех сетевых интерфейсов. С параметром ‘/all’ кроме сказанного, выводятся МАС-адреса сетевых интерфейсов, имя узла, адреса серверов DNS и WINS и некоторая другая информация.

**Формальный синтаксис:**

ipconfig [/? | /all | /release [адаптер] | /renew [адаптер] | /flushdns | displaydns /registerdns | /showclassid адаптер | /setclassid адаптер [устанавливаемый\_код\_ класса\_dhcp]]

* Параметр ‘/flushdns’ очищает кэш разрешенных имен DNS.
* Параметр ‘/displaydns’ выводит кэш разрешенных имен DNS на экран.
* Параметр ‘/release [адаптер]’ освобождает арендованный по DHCP (Dynamic Host Configuration Protocol) IP-адрес (если указан адаптер, то только для этого адаптера, иначе для всех адаптеров).
* Параметр ‘/renew [адаптер]’ запрашивает обновление аренды по DHCP (если указан адаптер, то только для этого адаптера, иначе для всех адат'еров).
* Параметр ‘/registerdns’ запрашивает обновление аренды по DHCP всех адресов и повторную их регистрацию в DNS.

**3) Порядок выполнения работы:**

1. Программа должна быть оформлена в виде удобной утилиты с интерактивным интерфейсом пользователя
2. Текст программы оформляется прилично (удобочитаемо, с описанием ВСЕХ функций, переменных и критических мест).
3. В процессе работы программа ОБЯЗАТЕЛЬНО выдает информацию о состоянии процесса
4. Интерфейс программы может быть произвольным, но удобным и понятным (разрешается использование библиотек GUI)
5. Среда разработки и язык программирования могут быть произвольными

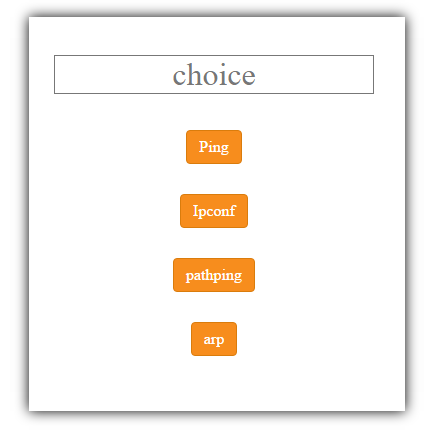
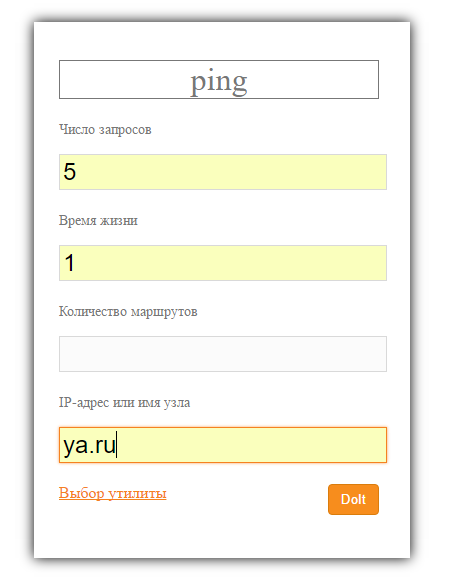


Рис1. Форма для выбора стандартной Windows сетевой утилиты



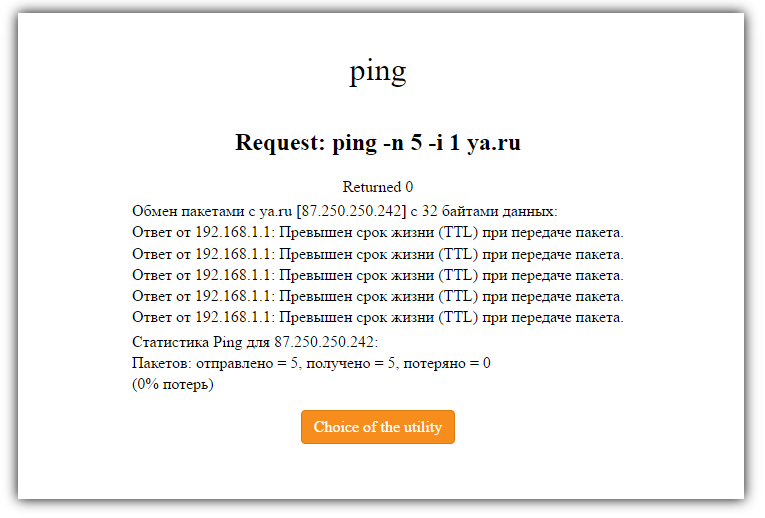


Рис2. Утилита Ping

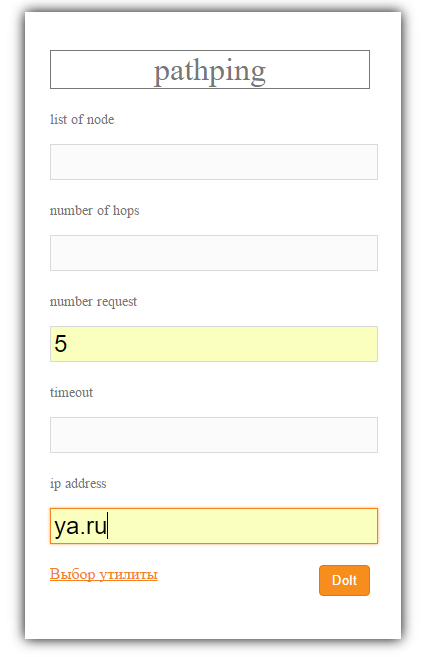


Рис3. Утилита Pathping

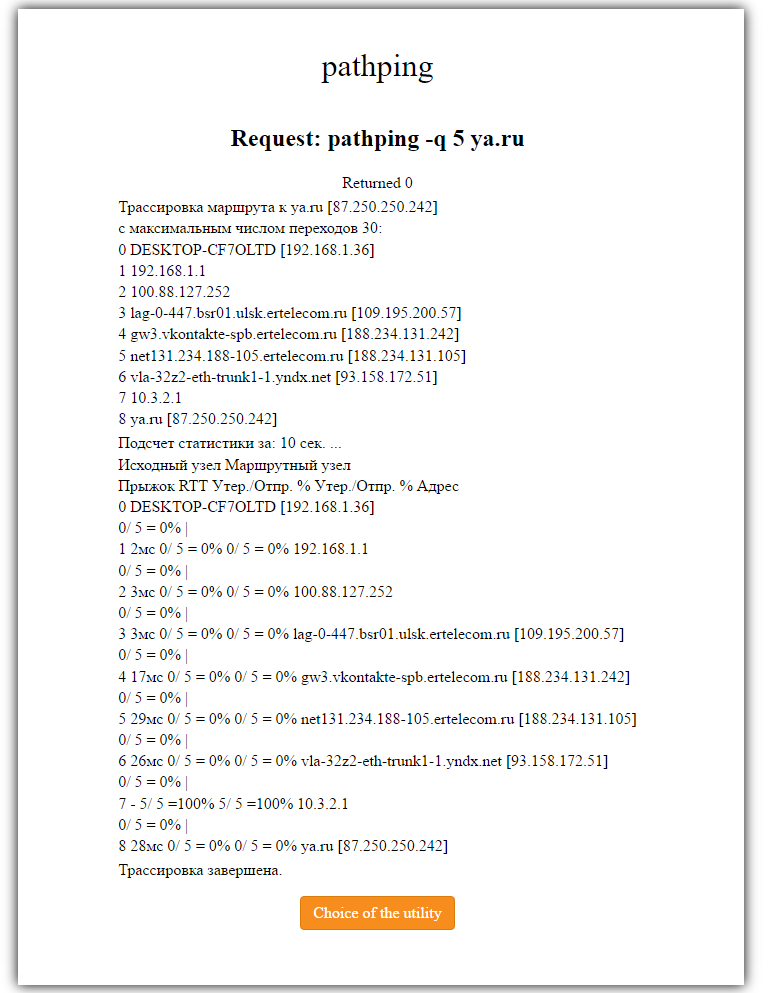
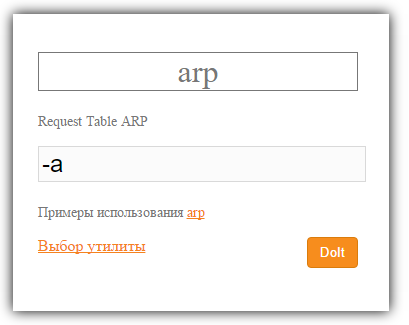


Рис4. Утилита Pathping



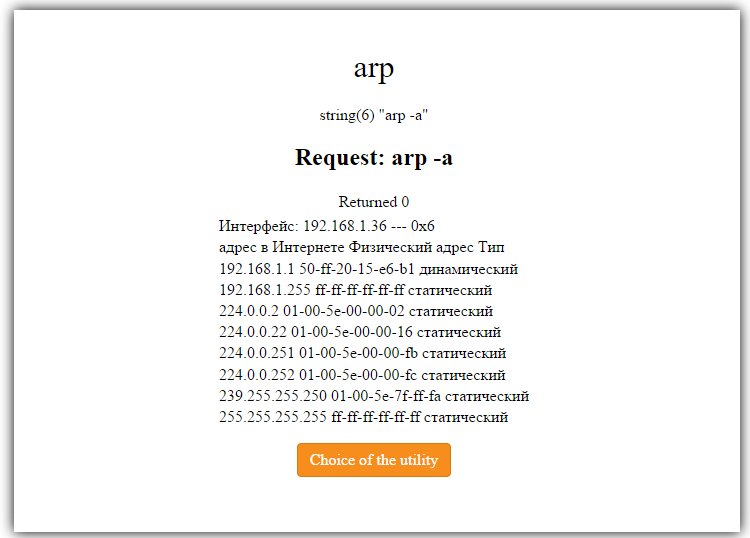
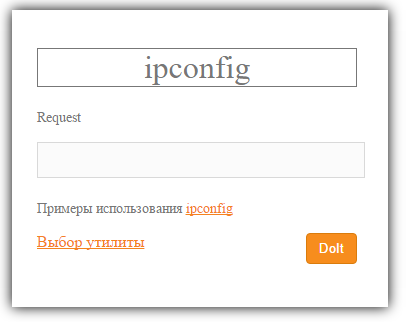


Рис5. Утилита Arp



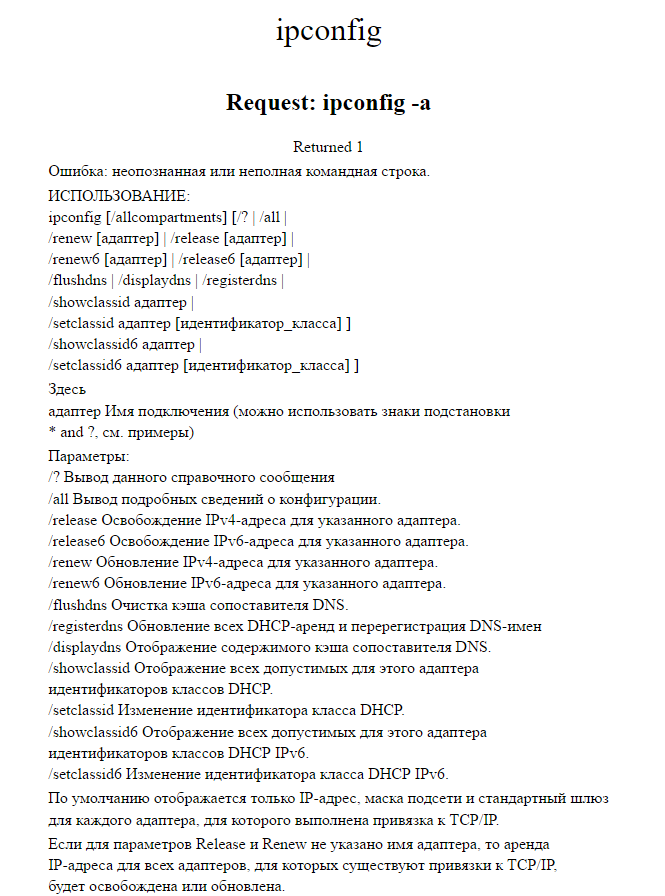


Рис5. Утилита Ipconfig

**Вывод:**

В ходе выполнения данной лабораторной работы, мне удалось приобрести практические знания и навыки по настройке локальных сетей и получения информации о сетевых интерфейсах с использованием стандартных утилит ОС Windows.

**Список литературы:**

* Руководство по PHP https://www.php.net/manual/ru/index.php (Дата обращения 2.12.19).
* Лабораторная работа Мартынов Антон Иванович - Сети и телекоммуникации.
* Советы по использованию IPconfig в Windows https://ip-calculator.ru/blog/ask/sovety-po-ispolzovaniyu-ipconfig-v-windows/ (Дата обращения 8.12.19).
* Работа с сетью в командной строке Windows (CMD). https://ab57.ru/netcmd.html#id01 (Дата обращения 8.12.19).

**Приложение 1**

<?php

header("Content-Type: text/html; charset=cp866");

function ping($numberRequest, $lifeTime, $numberRoutes, $ipAddress) {

$numberRequest ? $numberRequest = '-n ' . $numberRequest : '';

$lifeTime ? $lifeTime = '-i ' . $lifeTime : '';

$numberRoutes ? $numberRoutes = '-r ' . $numberRoutes : '';

$request = "ping $numberRequest $lifeTime $numberRoutes $ipAddress";

exec($request, $output\_array, $return\_val);

echo "<h2> Request: $request </h2>";

echo "<p> Returned $return\_val </p> <div class=\"output\">";

foreach ($output\_array as $o) {

if(isset($o))

echo "<p> $o </p>";

}

echo "</div>";

}

function ipconfig($Request) {

$request = "ipconfig $Request";

exec($request, $output\_array, $return\_val);

echo "<h2> Request: $request </h2>";

echo "<p> Returned $return\_val </p> <div class=\"output\">";

foreach ($output\_array as $o) {

if(isset($o))

echo "<p> $o </p>";

}

echo "</div>";

}

function pathping($listNode, $numberHops, $numberRequest, $timeout, $ipAddress) {

$listNode ? $listNode = '-g ' . $listNode : '';

$numberHops ? $numberHops = '-h ' . $numberHops : '';

$numberRequest ? $numberRequest = '-q ' . $numberRequest : '';

$timeout ? $timeout = '-w ' . $timeout : '';

$ipAddress ? $ipAddress = $ipAddress : '';

$request = "pathping $listNode $numberHops $numberRequest $timeout $ipAddress";

exec($request, $output\_array, $return\_val);

echo "<h2> Request: $request </h2>";

echo "<p> Returned $return\_val </p> <div class=\"output\">";

foreach ($output\_array as $o) {

if(isset($o))

echo "<p> $o </p>";

}

echo "</div>";

}

function arp($Request) {

$request = "arp $Request";

var\_dump($request);

exec($request, $output\_array, $return\_val);

echo "<h2> Request: $request </h2>";

echo "<p> Returned $return\_val </p> <div class=\"output\">";

foreach ($output\_array as $o) {

if(isset($o))

echo "<p> $o </p>";

}

echo "</div>";

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link href="css/custom.css" rel="stylesheet"> <link href="css/options.css" rel="stylesheet"><title>lab1</title>

</head>

<body>

<div class="container mlogin">

<div id="result">

<h1><?php echo $\_POST['command-name']?></h1>

<div class="choice">

<?php

switch($\_POST['command-name']) {

case 'ping':

ping($\_POST['number-request'], $\_POST['life-time'], $\_POST['number-routes'], $\_POST['ip-address']);

break;

case 'ipconfig':

ipconfig($\_POST['request']);break;

case 'pathping':

pathping($\_POST['list-node'], $\_POST['number-hops'], $\_POST['number-request'], $\_POST['timeout'], $\_POST['ip-address']); break;

case 'arp':

arp($\_POST['request']);break;

default:

break;

}

?>

<a href="choice.php" name="btn" class="button">Choice of the utility</a>

</div>

</div>

</div>

</body>

</html>