

## Сети ЭВМ и телекоммуникации

### Лабораторная работа №1. «Использование сетевых утилит ОС Windows для проверки и настройки локальной сети»

#### Цель работы:

Приобретение практических знаний и навыков по настройке локальных сетей и получения информации о сетевых интерфейсах с использованием стандартных утилит ОС Windows.

#### Теоретические сведения

##### 1. Утилита ping

Утилита ping тестирует сетевое соединение путем отправки 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. Общий формат использования утилиты (находящиеся в квадратных скобках параметры опциональны):

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

Для получения такой подсказки достаточно запустить ping без параметров, для вывода подсказки в файл ping\_test.txt следует использовать ping > ping\_test.txt (то же относится и к большинству иных утилит).

- **Параметр ‘-t’** включает постоянную проверку связи до нажатия Ctrl+C. При нажатии Ctrl+Break выводится статистически накопленная информация и работа продолжается (обычно этот параметр используют, чтобы как можно быстрее узнать о наличии связи с заданным узлом).
- **Параметр ‘-a’** требует определение IP-адреса по имени узла (по умолчанию не выполняется).
- **Параметр ‘-n <число>’** позволяет задать количество запросов (по умолчанию четыре запроса).
- **Параметр ‘-l <число>’** дает возможность задать размер пакета (по умолчанию размер пакета 64 байта).
- **Параметр ‘-f’** позволяет установить в запросах флаг ‘не фрагментировать’. Используется в сочетании с параметром ‘-l’ для обнаружения сетей с малым размером кадра, для передачи через которые IP-пакеты приходится фрагментировать.
- **Параметр ‘-i <число>’** задает время жизни пакета (TTL), по умолчанию у ICMP-пакетов время жизни равно 255.
- **Параметр ‘-r <число>’** дает возможность получить маршрут, по которому передавались запрос и ответ (показать маршрутизацию). Числовой параметр может

быть от 1 до 9 и определяет максимальное количество узлов, которые будут показаны в маршруте.

- Параметр **'-w <число>'** позволяет задать время ожидания каждого пакета (в миллисекундах), по умолчанию это 1'000 миллисекунд.
- **КонечноеИмя** задает IP-адрес или имя узла

## 2. Утилита **tracert**

Утилита **tracert** дает возможность проследить маршрут пакетов до заданного узла и получить временные характеристики для каждого промежуточного маршрутизатора на этом пути.

Эта утилита, как и ранее описанная **ping**, отправляет серию пакетов ICMP типа 8, но с разными значениями TTL:

1. сначала отправляется три пакета с TTL=1 (на эти пакеты ближайший маршрутизатор ответит пакетами ICMP типа 11 (истекло время передачи), из которых будет извлечен его адрес),
2. затем с TTL=2 (на эти пакеты ответит второй маршрутизатор) и так далее до тех пор, пока не будет достигнут заданный узел или значение TTL не превысит порог.

Для каждого TTL утилита выводит по одной строке с адресом маршрутизатора (и, возможно, с его доменным именем - если удалось его разрешить) и тремя значениями времени, которое понадобилось для передачи пакета.

Формальный синтаксис:

**tracert [-d] [-h максЧисло] [-j списокУзлов] [-w интервал] имя**

- Параметр **'-d'** позволяет (принудительно) не выполнять разрешение IP адресов маршрутизаторов в доменные имена, это позволяет ускорить работу утилиты за счет отмены обращения к службе DNS.
- Параметр **'-h '** дает возможность задать порог, до которого будет расти TTL (по умолчанию - 30).
- Параметр **'-w '** позволяет задать время ожидания каждого пакета (в миллисекундах), по умолчанию 1'000 миллисекунд.

## 3. 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.

#### 4. Утилита **arp**

Утилита **arp** дает возможность просматривать и изменять ARP-таблицу, в которой хранятся пары 'MAC-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\_addr]**

или

**ARP -a [inet\_addr] [-N if\_addr]**

- Параметр **if\_addr** задает номер интерфейса.
- Параметр **'-a'** позволяет вывести всю ARP-таблицу на экран.
- Параметр **'-a <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.

#### 5. 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 MAC-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 адаптеров).
- Параметр **'/registerdns'** запрашивает обновление аренды по DHCP всех адресов и повторную их регистрацию в DNS.

#### 6. Утилита **route**

Утилита **route** отображает таблицу маршрутов и позволяет ее изменять. Формальный синтаксис:

**route [-f] [-p] [команда [узел]] [MASK маска] [шлюз] [METRIC метрика] [IF-интерфейс]**

При использовании route параметр 'метрика' определяет качество данного маршрута (в хопх – количестве промежуточных маршрутизаторов, времени прохождения пакета по линиям связи, характеристикой надежности линии связи на данном маршруте и т.п.) в соответствии с заданным в сетевом пакете критерием (т.н. классом сервиса).

- команда 'PRINT' выводит таблицу маршрутов: сетевой адрес; маска сети; адрес шлюза; интерфейс; метрика,
- команда 'ADD' позволяет добавить новый маршрут,
- команда 'DELETE' – удалить маршрут,
- команда 'CHANGE' – изменить (существующий) маршрут).

## 7. Утилита netstat

Утилита netstat отображает текущие соединения, порты, ожидающие соединения и статистические данные по протоколам TCP/IP. Без дополнительных параметров выводится список текущих соединений (протокол: TCP или UDP; локальный адрес и порт; внешний адрес и порт; состояние соединения).

Формальный синтаксис:

**netstat [-a] [-e] [-n] [-s] [-r имя] [-r] [интервал]**

- **Параметр '-a'** дополнительно отображает порты, ожидающие соединения; ожидающие TCP-порты обозначены состоянием 'LISTENING', а UDP-порты - внешним адресом '\*.\*'.
- **Параметр '-n'** требует выводить все адреса и номера портов в числовом формате, поскольку по умолчанию netstat пытается разрешить IP-адреса и имена и заменить номер порта на его имя.
- **Параметр '-r'** выводит таблицу маршрутов (сетевой адрес; маска сети; адрес шлюза; интерфейс; метрика). Подобную информацию можно получить с помощью утилиты route.
- **Параметр '-e'** позволяет получить статистику Ethernet.
- **Параметр '-s'** выводит статистику по протоколам TCP, UDP и IP.
- **Параметр '-e <протокол>'** применяется совместно с параметром '-s' для ограничения выдаваемой статистики заданным протоколом (TCP, UDP или IP).

**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 Proxu-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 MAC- 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).

**Примечание:**

1. Задание является дифференцированным.

- На оценку «Удовлетворительно» достаточно уметь пользоваться всеми вышеперечисленными командами в режиме командной строки
- На оценку «Хорошо» необходимо реализовать приложение с GUI, которое позволяет выполнять команды, указанные в варианте и выводить результаты в удобном для пользователя виде:
  - Параметры вызова всех команд должны быть доступны через настройки пользовательского интерфейса
  - Каждая результирующая строка команды ping или pathping должна сохраняться в отдельной строке таблицы с разделением по столбцам
  - Результаты команды ipconfig по каждому интерфейсу сохраняются на отдельной вкладке
  - Для команд arp, netstat и route необходим удобный интерфейс, который позволяет добавлять, удалять, изменять параметры
- На оценку «Отлично» необходимо дополнительно к GUI интерфейсу реализовать графический интерфейс, позволяет:
  - Для команд arp, netstat и route показывать проанализированную структуру сети со связями в виде графа, где узлами являются объекты сети, а ребрами связи между ними
  - Для команд ping или pathping визуально показывать перемещение пакетов между текущим узлом и проверяемым на построенном графе

**Требования для сдачи лабораторной работы:**

1. Демонстрация работы реализованной вами системы.
2. АВТОРСТВО
3. Оформление и представление письменного отчета по лабораторной работе, который содержит:
  - Титульный лист
  - Задание на лабораторную работу
  - Сам отчет, который содержит:
    - При использовании утилит ping, tracert или pathping необходимо отметить время передачи пакетов (обычно среднее), число и долю

потерянных пакетов (по ней сделать вывод о корректности работы каждого промежуточного маршрутизатора).

- При использовании утилиты netstat необходимо зафиксировать назначение TCP- и UDP-портов имени ПЭВМ, локальному адресу и их состояние (параметр '-a'). Общую статистику обменов можно получить при использовании параметра '-e', с использованием параметра '-s' следует зафиксировать и проанализировать статистику по всем протоколам стека TCP/IP (включая ICMP, UDP).
- При использовании утилиты route (с параметром PRINT для вывода на экран) просматривается как список интерфейсов, так и список активных маршрутов. Удаление маршрута достигается вводом route DELETE узел (где узел - IP-адрес удаляемого из маршрута узла), добавление - route ADD узел MASK маска шлюз METRIC метрика IF интерфейс (где узел – IP-адрес добавляемого узла, маска – значение маски, шлюз – IP-адрес шлюза, метрика – значение метрики добавляемого маршрута, интерфейс – номер сетевого интерфейса; может быть опущен, тогда выбирается наиболее подходящий для указанного шлюза).

#### Варианты заданий:

№	На оценку «Удовлетворительно»	Список команд, которые нужно реализовать на оценку «хорошо» или «отлично»
1.	Уметь пользоваться всеми вышеперечисленными командами в режиме командной строки.	ping, ipconfig, tracert, route
2.		ping, ipconfig, pathping, netstat
3.		ping, ipconfig, tracert, arp
4.		ping, ipconfig, pathping, route
5.		ping, ipconfig, tracert, netstat
6.		ping, ipconfig, pathping, arp
7.		ping, ipconfig, tracert, route
8.		ping, ipconfig, pathping, netstat
9.		ping, ipconfig, tracert, arp
10.		ping, ipconfig, pathping, route
11.		ping, ipconfig, tracert, netstat
12.		ping, ipconfig, pathping, arp
13.		ping, ipconfig, tracert, route
14.		ping, ipconfig, pathping, netstat
15.		ping, ipconfig, tracert, arp