Системные утилиты сетевой диагностики

Утилита ipconfig

Утилита *ipconfig* предназначена для проверки правильности конфигурации TCP/IP для операционной системы Windows. Выводит значения для текущей конфигурации стека TCP/IP: MAC- и IP-адрес, маску подсети, адрес шлюза по умолчанию, адреса серверов WINS (*Windows Internet Naming Service*) и DNS, использование DHCP (Dynamic Host Configuration Protocol).

При устранении неисправностей в сети TCP/IP следует сначала проверить правильность конфигурации с помощью утилиты *ipconfig.*

Синтаксис утилиты: *ipconfig* [*/all*] [*/renew*[*adapter*]] [*/release* [*adapter*]]. Параметры (здесь и далее в квадратных скобках указаны необязательные параметры):

- *all* выдает весь список параметров, без этого ключа отображается только IP-адрес, маска и шлюз по умолчанию;
- *renew* [*adapter*] обновляет параметры конфигурации DHCP для указанного сетевого адаптера именем *adapter* ;
- *release* [*adapter*] освобождает выделенный DHCP IP-адрес.

Таким образом, утилита *ipconfig* (рис. 1.1) позволяет выяснить, инициализирована ли конфигурация и не дублируются ли IP-адреса:

- если конфигурация инициализирована, то появляются IP-адрес, маска, шлюз;
- если IP-адреса дублируются, то маска сети будет 0.0.0.0;
- если при использовании DHCP компьютер не смог получить IP-адрес, то он будет равен 0.0.0.0 .

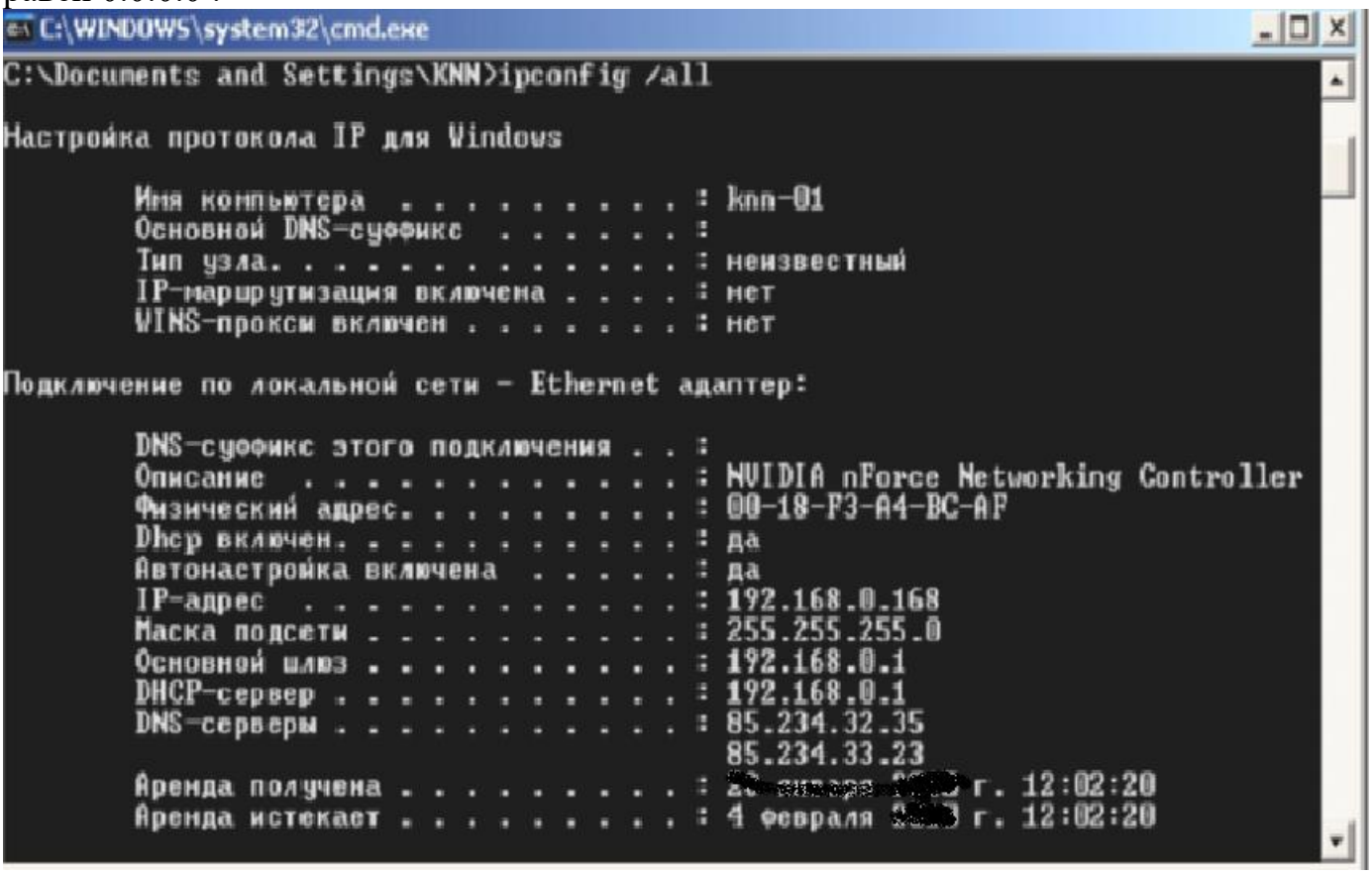

Рис. 1.1. Отображение установленных на компьютере сетевых конфигураций утилитой *ipconfig*

1.2.3.2. Утилита *ping*

Утилита *ping* (*Packet Internet Grouper*) используется для проверки конфигурирования TCP/IP и диагностики ошибок соединения. Она определяет доступность и функционирование конкретного хоста – любого сетевого устройства, обменивающегося информацией с другими сетевыми устройствами по TCP/IP. Использование *ping* есть лучший способ проверки существования маршрута между локальным компьютером и сетевым хостом.

Команда *ping* проверяет соединение с удаленным хостом путем посылки к нему эхо-пакетов протокола ICMP (*Internet Control Message Protocol*) и прослушивания эхо-ответов. *Ping* выводит количество переданных и принятых пакетов. Каждый принятый пакет проверяется в соответствии с переданным сообщением. Если связь между хостами плохая, из сообщений *ping* станет ясно, сколько пакетов потеряно.

По умолчанию передаются четыре эхо-пакета длиной 32 байта, представляющих собой последовательность символов алфавита в верхнем регистре. *Ping* позволяет изменить размер и количество пакетов, указать, следует ли записывать маршрут, который она использует, какую величину времени жизни устанавливать, можно ли фрагментировать пакет и т.д. При получении ответа в поле определяется, за какое время (в миллисекундах) посланный пакет доходит до удаленного хоста и возвращается назад. Так как значение по умолчанию для ожидания отклика равно 1 с, то все значения данного поля будут меньше 1000 мс. Если получается сообщение «Превышен интервал ожидания», то, возможно, увеличение времени ожидания отклика позволит пакету дойди до удаленного хоста.

При пользовании утилитой *ping* следует помнить:

 задержка, определенная утилитой, вызвана не только пропускной способностью канала передачи данных до проверяемой машины, но и загруженностью этой машины;

 некоторые серверы в целях безопасности могут не посылать эхо-ответы, так как с утилиты *ping* может начинаться хакерская атака.

Ping можно использовать для тестирования как с доменным именем хоста, так и с его IP-адресом. Если *ping* с IP-адресом выполнилась успешно, а с именем –неудачно, это значит, что проблема заключается в распознавании соответствия адреса и имени, а не в сетевом соединении.

Синтаксис: *ping* [*-t*] [*-a*] [*-n count*] [*-l length*] [*-f*] [*-i ttl*] [*-v tos*] [*-r count*] [*-s count*] [[*-j host-list*| [*-k host-list*]] [*-w timeout*] *destinationlist.*

Параметры:

 -t выполняет команду *ping* до прерывания (*Ctrl-Break* – посмотреть статистику и продолжить, *Ctrl-C* – прервать выполнение команды);

 -a позволяет определить доменное имя удаленного компьютера по его IP-адресу;

 -n count посылает количество пакетов *Echo*, указанное пара метром *count* (по умолчанию передается четыре запроса);

 -l length посылает пакеты длиной *length* байт (максимальная длина 8192 байта);

 -f посылает пакет с установленным флагом «не фрагментировать», запрещающим фрагментирование пакета на транзитных маршрутизаторах;

 -i ttl устанавливает время жизни пакета в величину *ttl* (каждый маршрутизатор уменьшает *ttl* на единицу, т.е. время жизни является счетчиком пройденных маршрутизаторов (хопов));
 -v tos устанавливает значение поля «сервис», задающее приоритет обработки пакета;
 -r count записывает путь выходящего пакета и возвращающегося пакета в поле записи пути, *count* – от 1 до 9 хостов;
 -s count задает максимально возможное количество переходов из одной подсети в другую (хопов);
 -j host-list направляет пакеты с помощью списка хостов, определенного параметром *host-list.*), максимальное количество хостов равно 9;
 -k host-list направляет пакеты через список хостов, определенный в *host-list*, причем указанные хосты не могут быть разделены промежуточными маршрутизаторами (жесткая статическая маршрутизация);
 -w timeout указывает время ожидания *timeout* ответа от удаленного хоста в миллисекундах (по умолчанию – 1с);
 -destination-list указывает удаленный узел, к которому надо направить пакеты *ping*, может быть именем хоста или IP-адресом машины.

На практике в формате команды чаще всего используются опции *-t* и *-n.*

Пример работы утилиты *ping* приведен на рис. 1.2.

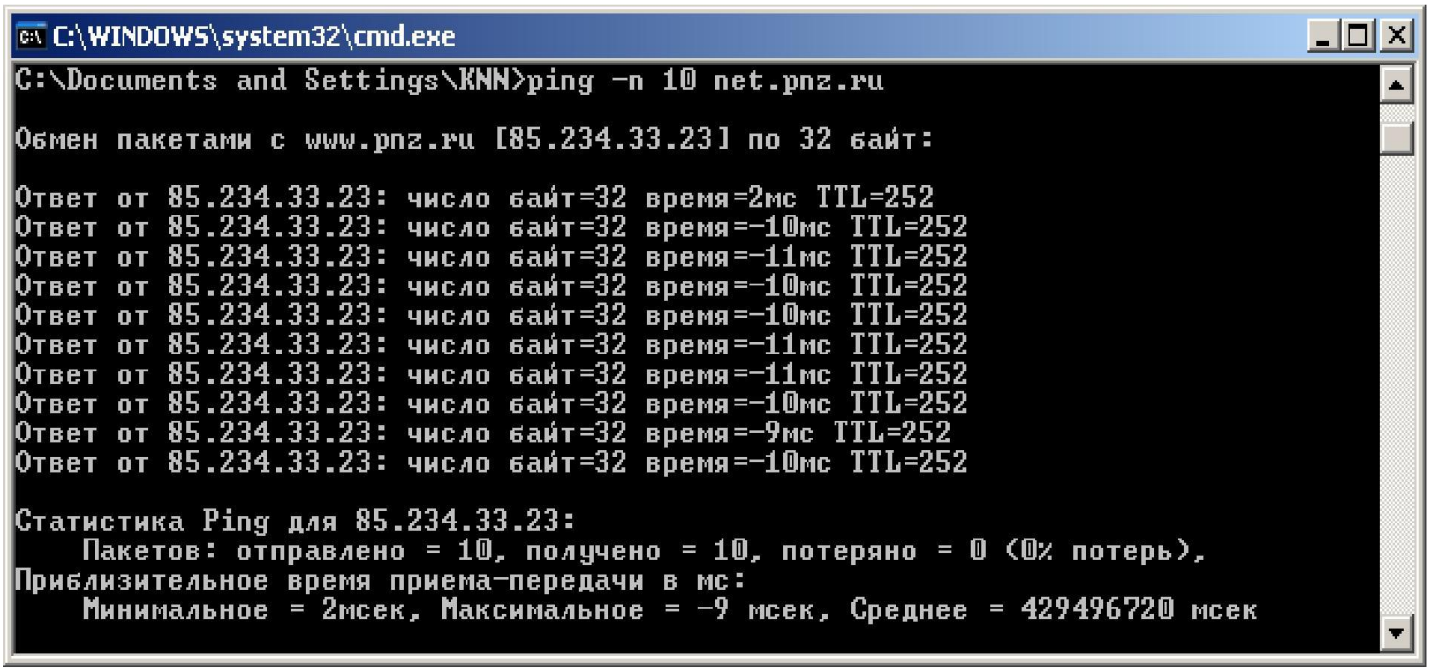

Рис. 1.2. Пример использования утилиты *ping*

Утилита *ping* может использоваться следующими способами:

1. Для проверки того, что TCP/IP установлен и правильно сконфигурирован на локальном компьютере, в команде ping задается адрес петли обратной связи : *ping 127.0.0.1*

Если тест успешно пройден, то вы получите следующий ответ:

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

2. Чтобы убедиться в том, что компьютер правильно добавлен в сеть и IP-адрес не дублируется, используется IP-адрес локального компьютера: *ping IP-адрес_локального_хоста.*
3. Чтобы проверить, что шлюз по умолчанию функционирует и можно установить соединение с любым хостом в локальной сети, задается IP-адрес шлюза по умолчанию: *ping IP-адрес_шлюза.*
4. Для проверки возможности установления соединения через маршрутизатор в команде *ping* задается IP-адрес удаленного хоста: *ping IP-адрес_удаленного хоста.*

1.2.3.3. Утилита *tracert*

Утилита *tracert* (*trace route*) позволяет выявлять последовательность маршрутизаторов, через которые проходит IP-пакет на пути к пункту своего назначения путем изучения сообщений ICMP, которые присылаются обратно промежуточными маршрутизаторами.

Утилита *tracert* работает следующим образом: посылается по три пробных эхо-пакета протокола ICMP с TTL=1 на узел назначения, первый маршрутизатор пошлет в компьютер-источник сообщение ICMP «Время истекло». Затем TTL увеличивается на 1 в каждой последующей посылке до тех пор, пока пакет не достигнет хоста назначения либо не будет достигнута максимально возможная величина TTL (по умолчанию 30).

Имя машины может быть именем хоста или IP-адресом машины. Выходная информация представляет собой список хостов, начиная с первого шлюза и заканчивая пунктом назначения. На экран при этом выводится время ожидания ответа на каждый пакет.

В тех случаях, когда удаленный узел не достижим, применение утилиты *tracert* более удобно, чем *ping,* так как с ее помощью можно локализовать район сети, в которой имеются проблемы со связью. Если возникли проблемы, то утилита выводит на экран звездочки (*) либо сообщения типа «Заданная сеть недоступна», «Время истекло».

Следует помнить, что некоторые маршрутизаторы просто уничтожают пакеты с истекшим TTL и не будут видны утилите *tracert*.

Синтаксис утилиты: *tracert* [*-d*] [*-h maximum_hops*] [*-j host-list*] [*-w timeout*] *destination-list*. Параметры:

- *-d* указывает, что не нужно распознавать адреса для имен хостов;
- *-h maximum_hops* указывает максимальное число хопов (по умолчанию – 30);
- *-j host-list* указывает нежесткую статическую маршрутизацию в соответствии с *host-list*;
- *-w timeout* указывает, что нужно ожидать ответ на каждый эхо-пакет заданное число мс;
- *-destination-list* указывает удаленный узел, к которому надо направить пакеты *ping*.

Пример работы утилиты *tracert* приведен на рис. 1.3.

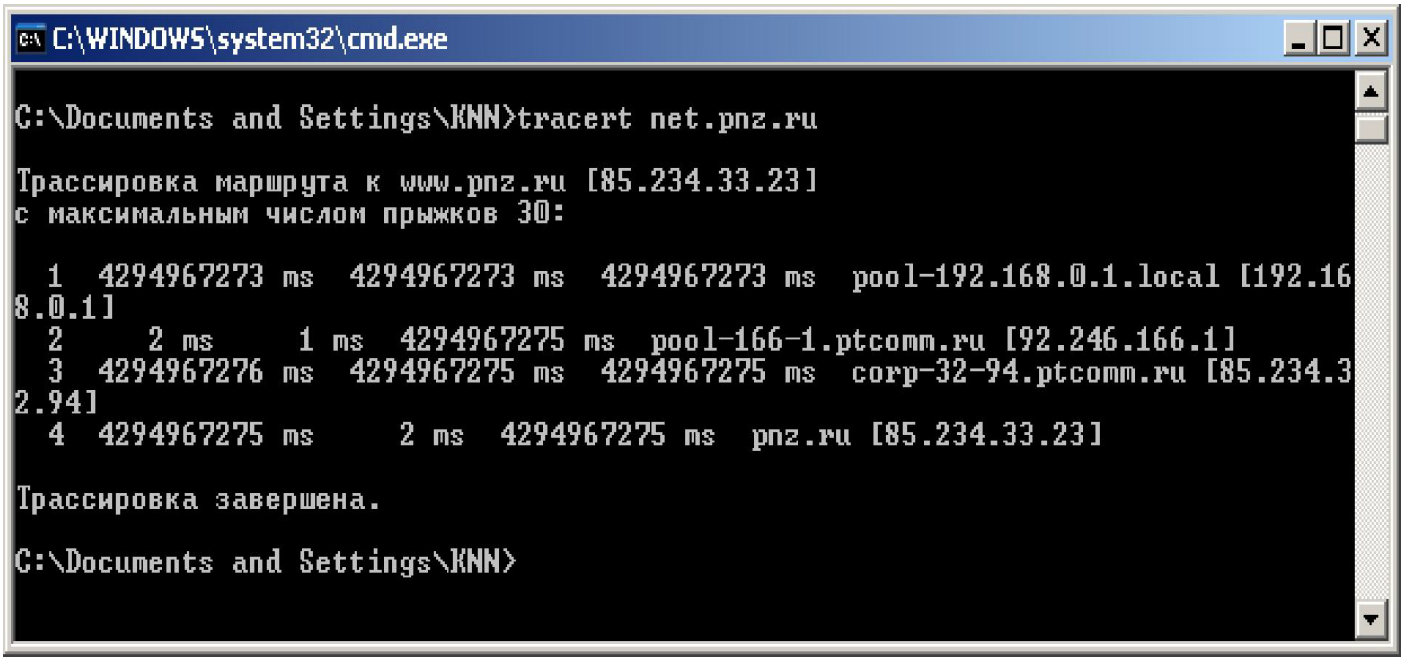

Рис. 1.3. Пример использования утилиты *tracert*

1.2.3.4. Утилита arp

Утилита *arp* (*Address Resolution Protocol* – протокол разрешения адресов) позволяет управлять так называемым ARP-кэшем – таблицей, используемой для трансляции IP-адресов в соответствующие локальные адреса. Записи в ARP-кэше формирует протокол ARP. Если необходимая запись в таблице не найдена, то протокол ARP отправляет широковещательный запрос ко всем компьютерам локальной подсети, пытаясь найти владельца данного IP-адреса.

В кэше могут содержаться два типа записей: статические и динамические. Статические записи вводятся вручную и хранятся в кэше постоянно. Динамические записи помещаются в кэш в результате выполнения широковещательных запросов. Для них существует понятие времени жизни. Если в течение определенного времени (по умолчанию 2 мин) запись не была востребована, то она удаляется из ARP-кэша.

Синтаксис утилиты: *arp* [*-s inet_addr eth_addr*] [*-d inet_addr*] [*-a*].

Параметры:

- *-s inet_addr eth_addr* заносит в кэш статическую запись с указанными IP-адресом и MAC-адресом;
- *-d inet_addr* удаляет из кэша запись для определенного IP-адреса;
- *-a* просматривает содержимое кэша для всех сетевых адаптеров локального компьютера, как показано на рис. 1.4.

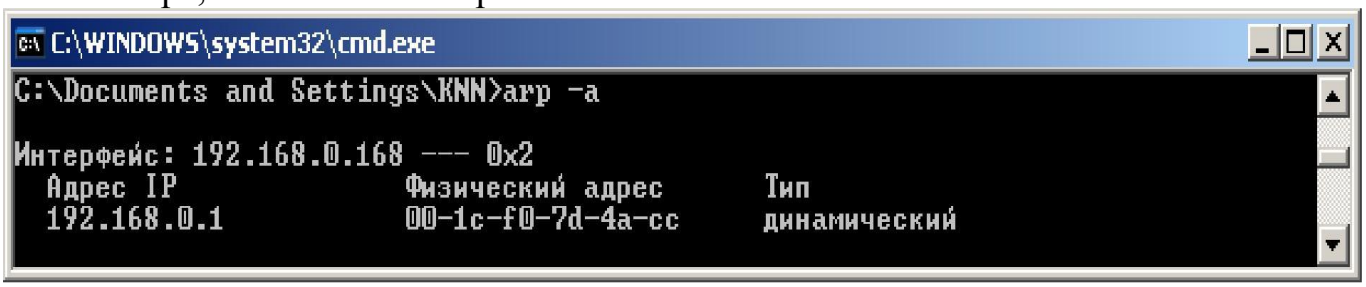

Рис. 1.4. Пример использования утилиты *arp*

1.2.3.5. Утилита *netstat*

Утилита *netstat* выводит статистику протоколов и текущих TCP/IP соединений и имеет следующий синтаксис: *netstat* [*-a*][*-e*][*-n*][*-s*][*-p name*][*-r*][*interval*]. Параметры:

- *-a* отображает полную информацию по всем соединениям и портам, на которых компьютер ожидает соединения;
- *-e* отображает статистику Ethernet (этот ключ может применяться вместе с ключом *–s*);
- *-n* отображает адреса и номера портов в числовом формате, без их преобразования в символьные имена DNS и в название сетевых служб, что делается по умолчанию *t*;
- *-p name* задает отображение информации для протокола *name* (допустимые значения *name*: *tcp, udp* или *ip*) и используется вместе с ключом *s*;
- *-r* отображает содержимое таблицы маршрутов (таблица маршрутизации);
- *-s* отображает подробную статистику по протоколам. По умолчанию выводятся данные для TCP, UDP и IP. Ключ *p* позволяет задать вывод данных по определенному протоколу, ключ *interval* инициирует повторный вывод статистических данных через указанный в секундах интервал (в этом случае для прекращения вывода данных надо нажать клавиши *Ctrl+C*).

Результатом выполнения команды является список активных подключений, в который входят установленные соединения и открытые порты (рис. 1.5).

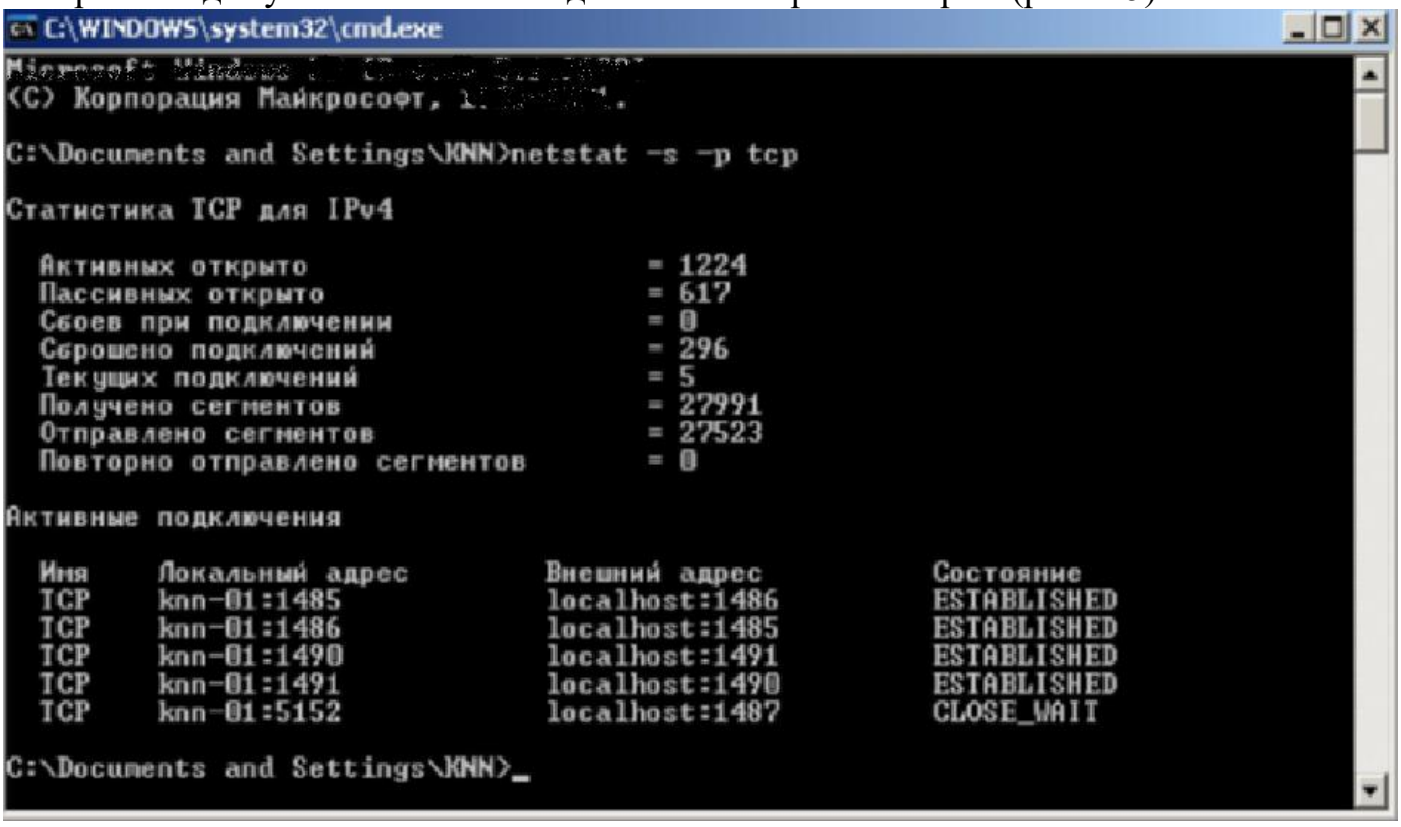

Рис. 1.5. Пример отображения утилитой *netstat* установленных на компьютере TCP-соединений

Открытые TCP-порты обозначаются в колонке «Состояние» строкой *LISTENING* – пассивно открытые соединения («слушающие» сокеты) или *ESTABLISHED* – установленные соединения, т.е. уже используемые сетевыми сервисами.

Часть портов связана с системными службами Windows и отображается не по номеру, а по названию – *epmap, microsoft-ds, netbios-ss* и др. Порты, не относящиеся к

стандартным службам, отображаются по номерам. UDP-порты не могут находиться в разных состояниях, поэтому специальная пометка *LISTENING* в их отношении не используется. Как и TCP-порты, они могут отображаться по именам или по номерам.

1.2.3.6. Утилита *nslookup*

Утилита *nslookup* предназначена для выполнения запросов к DNS-серверам на разрешение имен в IP-адреса и в простейшем случае имеет следующий синтаксис: *nslookup* [*host* [*server*]]. Параметры:

- *host* – доменное имя хоста, которое должно быть преобразовано в IP-адрес;
- *server* – адрес DNS-сервера, который будет использоваться для разрешения имени. Если этот параметр опущен, то будут использованы адреса DNS-серверов из параметров настройки протокола TCP/IP (отображаются утилитой *ipconfig*).

Результаты выполнения команды *nslookup* приведены на рис. 1.6.

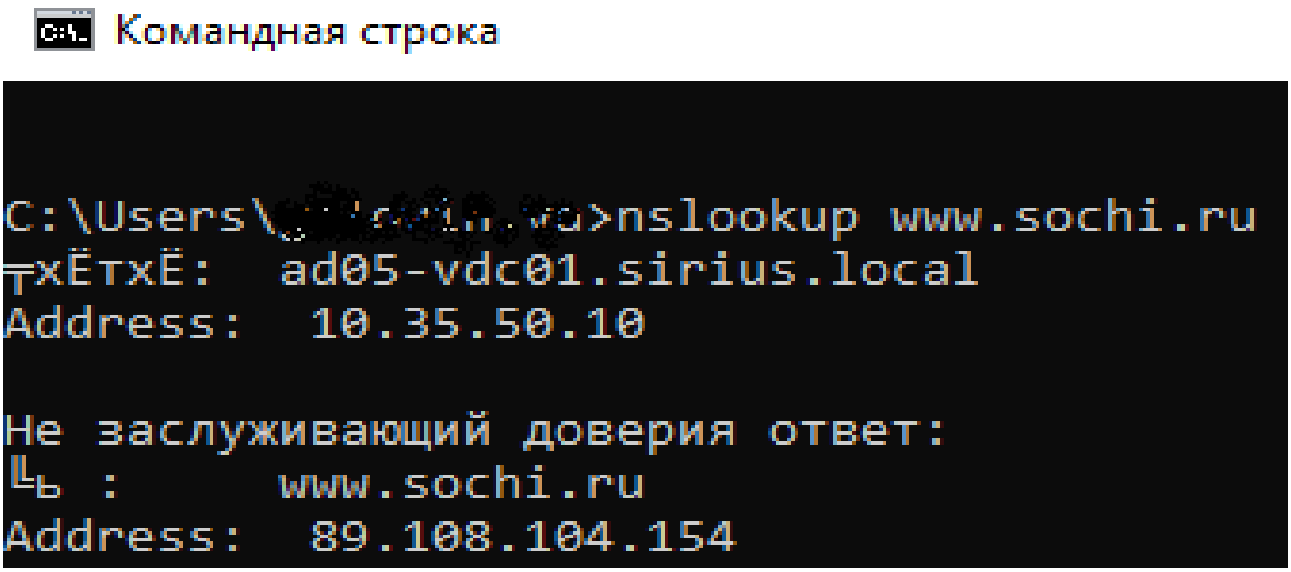

Рис. 1.6. Пример отображения утилитой *nslookup* запроса к DNS

Первые две строки ответа содержат имя и IP-адрес DNS-сервера, который был использован для разрешения имени. Следующие строки содержат реальное доменное имя хоста и его IP-адрес и указание *Nonauthoritativeanswer*, означающее, что ответ получен не с DNS-сервера, ответственного за зону *sochi.ru*. Также может присутствовать строка *Aliase*, которая содержит альтернативные имена искомого сервера.

1.2.3.7. Сервис *Whois*

При трассировке маршрутов или проверке доступности хоста в Internet часто возникает необходимость определить по IP-адресу хоста его юридического владельца и контактные данные его администратора.

В отношении доменов второго уровня эта информация становится свободно доступной для любого пользователя сети Internet через сервис *Whois*. On-line сервиса *Whois* можно получить через форму на странице сайта http://www.nic.ru/whois.

Задание на лабораторную работу

1.3.1. С помощью утилиты ipconfig, запущенной из командной строки, определить имя, IP-адрес и физический адрес основного сетевого интерфейса компьютера, IP-адрес шлюза, IP-адреса DNS-серверов и использование DHCP. Результаты представить в виде таблицы.

1.3.2. С помощью утилиты *nslookup* определить IP-адрес одного из удаленных серверов, доменные имена которых указаны в табл. 1.2.

1.3.3. С помощью утилиты *ping* проверить состояние связи с любыми компьютером и шлюзом локальной сети, а также с одним из удаленных серверов, доменные имена которых указаны в табл. 1.2.

Таблица 1.2

№	Адрес	№	Адрес
1.	*Sochi.com*	*12*	*adler.flagma.ru/*
2.	*krasnodar.rt.ru*	*13*	*sochi.zoon.ru*
3.	*sutr.ru*	*14.*	*adlernet.ru*
4.	*sochi.edu.ru*	*15.*	*sochi.wifire.ru*
5.	*sochisirius.ru*	*16.*	*rudn-sochi.ru*
6.	*sochipark.ru*	*17.*	*dendrarium.ru*
7.	*sochi.spravker.ru*	*18.*	*art-of-sochi.krd.muzkult.ru*
8.	*sochi.meldana.com*	*19*	*krasnodar.ucheba.ru*
9.	*Sports.ru*	*20*	*privetsochi.ru/*
10.	*sochi24.tv/*	*21*	*cisco.com/*
11.	*dns-shop.ru*	*22*	*twitter.com*

Число отправляемых запросов должно составлять не менее 10. Для каждого из исследуемых хостов отразить в виде таблицы IP-адрес хоста назначения, среднее время приема-передачи, процент потерянных пакетов.

1.3.4. С помощью утилиты *arp* проверить состояние ARP-кэша. Представить полученные значения ARP-кэша в отчете.

1.3.5. Провести трассировку одного из удаленных хостов в соответствии с вариантом, выбранным в п. 1.3.2. Если есть потери пакетов, то для соответствующих хостов среднее время прохождения необходимо определять с помощью утилиты *ping* по 10 пакетам. В отчете привести копию окна с результатами работы утилиты *tracert*. Определить участок сети между двумя соседними маршрутизаторами, который характеризуется наибольшей задержкой при пересылке пакетов. Для найденных маршрутизаторов с помощью сервиса *Whois* определить название организаций. Полученную информацию привести в отчете.

1.3.6. С помощью утилиты *netstat* посмотреть активные текущие сетевые соединения и их состояние на вашем компьютере, для чего:

 запустить несколько экземпляров веб-браузера, загрузив в них различные страницы с разных веб-сайтов (5-7);

 закрыть браузеры и с помощью *netstat* проверить изменение списка сетевых подключений.

Проконтролировать сетевые соединения в реальном масштабе времени, для чего:

 закрыть ранее открытые сетевые приложения;

 запустить из командной строки утилиту *netstat*, задав числовой формат отображения адресов и номеров портов и повторный вывод с периодом 20–30 с;
 в отдельном окне командной строки запустить утилиту *ping* в режиме «до прерывания»;
 наблюдать отображение *netstat*, текущей статистики сетевых приложений;
 с помощью клавиш *Ctrl+C* последовательно закрыть утилиты *ping* и *netstat*.

В отчете привести копии окон с результатами работы утилиты *netstat* с пояснением отображаемой информации.