**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**
**Государственное образовательное учреждение**
**высшего профессионального образования**
**ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Н. Н. Коннов, В. Б. Механов**

# АНАЛИЗ СЕТЕВЫХ ПРОТОКОЛОВ

**Лабораторный практикум по курсу**
**«Сети ЭВМ и телекоммуникации»**

**Часть 1**

**ПЕНЗА 2010**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение
высшего профессионального образования
«Пензенский государственный университет» (ПГУ)

Н. Н. Коннов, В. Б. Механов

# Анализ сетевых протоколов

Лабораторный практикум по курсу
«Сети ЭВМ и телекоммуникации»

Часть 1

Пенза
Издательство ПГУ
2010

УДК 683.3
К64

Р е ц е н з е н т
доктор технических наук, профессор,
заведующий кафедрой «Вычислительные системы и моделирование»
Пензенского государственного педагогического университета
имени В. Г. Белинского
*В. И. Горбаченко*

**Коннов, Н. Н.**

К64 Анализ сетевых протоколов : лаб. практикум по курсу «Сети ЭВМ и телекоммуникации» / Н. Н. Коннов, В. Б. Механов. – Пенза : Изд-во ПГУ, 2010. – Ч. 1. – 68 с.

Настоящее издание является руководством к лабораторному практикуму по курсу «Сети ЭВМ и телекоммуникации», посвященному изучению методов передачи данных в современных сетях ЭВМ. В первой части исследуются методы контроля и мониторинга с помощью стандартных утилит сетей, построенных на операционной системе Windows; изучаются протоколы сетевого и транспортного уровней и прикладные протоколы (HTTP и FTP) стека TCP/IP с помощью программы Network Monitor.

Лабораторный практикум подготовлен на кафедре «Вычислительная техника» и предназначен для студентов, обучающихся по направлению 230100 «Информатика и вычислительная техника».

**УДК 683.3**

# Лабораторная работа № 1

# Диагностические сетевые утилиты и их использование

## 1.1. Цель работы

**Целью работы** является изучение методов контроля и мониторинга сетей, построенных на базе стека протоколов TCP/IP с помощью диагностических утилит операционной системы Windows.

## 1.2. Теоретический материал

### 1.2.1. Адресация в IP-сетях

Сетевая операционная система Windows содержит набор утилит, полезных при диагностике сети, использующей протоколы TCP/IP. Основными задачами этих утилит являются:

- определение параметров и характеристик сети,
- определение работоспособности сети,
- в случае неправильного функционирования сети – локализация сегмента или сервиса, вызывающих неисправность.

Главными параметрами сетевых подключений являются их канальные и сетевые адреса и другие параметры, влияющие на работу сетевого уровня.

Каждый компьютер в сети Internet (их принято называть хостами) имеет адреса двух уровней: канального и сетевого.

Канальный адрес хоста определяется технологией, с помощью которой осуществляется его подключение к Internet. Для машин, входящих в локальные сети Ethernet, это так называемый MAC-адрес (*Media Access Control* – управление доступом к среде) сетевого адаптера, который назначается производителем оборудования и является уникальным. Для существующих технологий локальных сетей MAC-адрес имеет 48-разрядный формат (6 байтов):

- первый бит указывает: для одиночного (0) или группового (1) адресата предназначен кадр;
- следующий бит указывает, является ли MAC-адрес глобально (0) или локально (1) администрируемым;

- следующие 22 бита являются идентификатором фирмы производителя;
- младшие 3 байта назначаются уникальным образом самим производителем.

MAC-адреса обычно представляются в 16-разрядной системе, например, 00-E0-4C-78-23-FD. Адрес FF-FF-FF-FF-FF-FF является широковещательным.

В качестве сетевого адрес хоста Internet используется IP-адрес (*Internet Protocol Address*), который характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение. При связи через сеть Internet требуется глобальная уникальность адреса, что обеспечивается рекомендациями специального подразделения Internet InterNIC (*Network Information Center*). Провайдеры услуг Internet получают диапазоны адресов у подразделений InterNIC, а затем распределяют их между своими абонентами. В случае изолированной от Internet локальной сети уникальность сетевого адреса требуется лишь в ее пределах, при этом IP-адреса должны выбираться администратором из специально зарезервированных для таких сетей блоков «закрытых» адресов.

В наиболее распространенной четвертой версии протоколов Internet (IP.v4) IP-адрес представляет собой 32-битовое двоичное число, записываемое в виде четырех десятичных чисел (значения от 0 до 255), разделенных точками (например, 192.168.0.1). Адрес состоит из двух логических частей – номера сети и номера хоста в сети.

При классовой модели форматирования адресов значения первых битов адреса определяют, какая его часть относится к номеру сети, а какая – к номеру хоста, как показано в табл. 1.1.

*Таблица 1.1*

**Классовая модель форматирования адресов**

<table>
<tr><td rowspan="2">Класс</td><td colspan="10">IP адрес</td><td>Диапазон адресов</td></tr>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>25</td><td>24</td><td>23 16</td><td>15 8</td><td>7 0</td><td></td></tr>
<tr><td>A</td><td>0</td><td colspan="5">№ сети</td><td colspan="3">№ хоста</td><td></td><td></td><td>0.1.0.0–126.0.0.0</td></tr>
<tr><td>B</td><td>1</td><td>0</td><td colspan="4">№ сети</td><td colspan="2">№ хоста</td><td></td><td></td><td>128.0.0.0–191.255.0.0</td></tr>
<tr><td>C</td><td>1</td><td>1</td><td>0</td><td colspan="4">№ сети</td><td></td><td>№ хоста</td><td></td><td>192.0.1.0–223.255.255.0</td></tr>
<tr><td>D</td><td>1</td><td>1</td><td>1</td><td>0</td><td colspan="6">адрес группы multicast</td><td></td><td>224.0.0.0–239.255.255.255</td></tr>
<tr><td>E</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td colspan="5">зарезервировано</td><td></td><td>240.0.0.0–247.255.255.255</td></tr>
</table>

Ряд адресов сетей и подсетей являются особыми:

- если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того хоста, который сгенерировал этот пакет;
- если все двоичные разряды IP-адреса хоста равны 1, то пакет с таким адресом назначения является широковещательным, т.е. должен рассылаться всем хостам, находящимся в той же сети, что и источник этого пакета;
- если все двоичные разряды IP-адреса хоста равны 0, то этот адрес обозначает не отдельный адрес, а всю сеть;
- адрес 127.0.0.1 означает пересылку в пределах одного и того же хоста (используется для автономной отладки сетевого ПО);
- адреса закрытых сетей (частная сеть, сеть интранет) лежат в диапазонах 10.0.0.0–10.255.255.255, 172.16.0.0–172.31.255.255, 192.168.0.0–192.168.255.255.

В целях более экономного распределения IP-адресов между пользователями классовая модель вытесняется бесклассовой, при которой выделение разрядов в адресе, отводимых для нумерации сети, задается специальным четырехбайтовым кодом – маской подсети. Разряды маски, используемые для нумерации сетей, имеют единичные значения. Например, маска 255.255.255.240 (код 11111111.11111111.11111111.11110000 в двоичной системе) указывает, что для нумерации сети используется 28 старших разрядов, а для нумерации хоста – только 4 младших разряда соответствующего IP-адреса. Часто применяется запись IP-адресов вида 192.96.10.0/28. Число после косой черты означает количество единичных разрядов в маске подсети.

IP-адреса для конкретных компьютеров могут устанавливаться администратором сети вручную, что весьма трудоемко. Для автоматизации процесса назначения IP-адресов хостам сети локальной сети применяется специальный протокол DHCP (*Dynamic Host Configuration Protocol*), который обеспечивает статическое или динамическое назначение IP-адресов. Назначаемые адреса формирует DHCP-сервер по запросам DHCP-клиентских программ, устанавливаемых на отдельных хостах.

При автоматическом статическом способе DHCP-сервер без вмешательства оператора присваивает IP-адрес и другие параметры конфигурации клиента из пула (набора) наличных IP-адресов. Границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера. Между идентификатором клиента и его IP-адресом

по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первичного назначения сервером DHCP IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер назначает адрес клиенту на ограниченное время, что дает возможность впоследствии повторно использовать IP-адреса другими компьютерами.

### 1.2.2. Отображение символьных адресов на IP-адреса: служба DNS

Компьютеры используют для взаимодействия числовые IP-адреса, тогда как людям удобнее работать со словесными именами. Чтобы в сетевых приложениях можно было применять словесные имена, требуется механизм преобразования имен в IP-адреса, реализуемый службой доменных имен DNS (*Domain Name System*) распределенной базой данных, поддерживающей иерархическую систему имен для идентификации хостов в сети Internet.

Служба DNS предназначена для автоматического поиска IP-адреса по известному символьному имени хоста. DNS-серверы хранят часть базы данных о соответствии символьных имен и IP-адресов. Эта база данных распределена по административным доменам сети Internet. Клиенты сервера DNS знают IP-адрес сервера DNS своего административного домена и по протоколу IP передают запрос, в котором сообщают известное символьное имя и просят вернуть соответствующий ему IP-адрес.

Если данные о запрошенном соответствии хранятся в базе данного DNS-сервера, то он сразу посылает ответ клиенту, если же нет, то он посылает запрос DNS-серверу другого домена, который либо сам обрабатывает запрос, либо передает его другому DNS-серверу. Все DNS-серверы соединены иерархически, в соответствии с иерархией доменов сети Internet.

База данных DNS имеет структуру дерева, называемого доменным пространством имен, в котором каждый домен (узел дерева) имеет имя и может содержать поддомены. Имя домена идентифицирует его положение в этой базе данных по отношению к родительскому домену, причем точки в имени отделяют части, соответствующие хостам домена.

Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Доменное имя строится из слов, разде-

ленных точками и содержащих латинские буквы, цифры и значок «минус» (–). Доменные имена могут содержать до 63 символов и нечувствительны к регистру букв, т.е. заглавные и строчные буквы считаются одинаковыми.

Организация InterNIC, управляющая всем адресным пространством Internet, а также всем пространством имен, делегирует некоторым организациям право ведения доменов первого уровня, к которым относятся следующие «организационные» зоны (***com*** – коммерческие, ***edu*** – образовательные, ***gov*** – правительственные, ***int*** – международные, ***mil*** – военные, ***net*** – организации, обеспечивающие работу сети, ***org*** – некоммерческие организации, ***biz*** – то же самое, что и ***com***, ***info*** – информационные ресурсы), а также более двухсот «географических» доменов (***ru*** и ***su*** – Россия, ***uk*** – Великобритания, ***de*** – Германия, ***fr*** – Франция, ***ua*** – Украина и т.д.).

Владелец доменной зоны может организовывать в ней любые поддомены и делегировать функции администрирования этих поддоменов другим организациям. Поддомен создается путем дописывания к имени домена еще одного отделенного точкой слова слева. Каждый домен имеет уникальное имя, а каждый из поддоменов имеет уникальное имя внутри своего домена. Каждый хост в сети Internet однозначно определяется своим полным доменным именем, которое включает имена всех доменов по направлению от хоста к корню. Пример полного DNS-имени: ***alice.pnzgu.ru***.

## 1.2.3. Системные утилиты сетевой диагностики

### 1.2.3.1. 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.

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 .

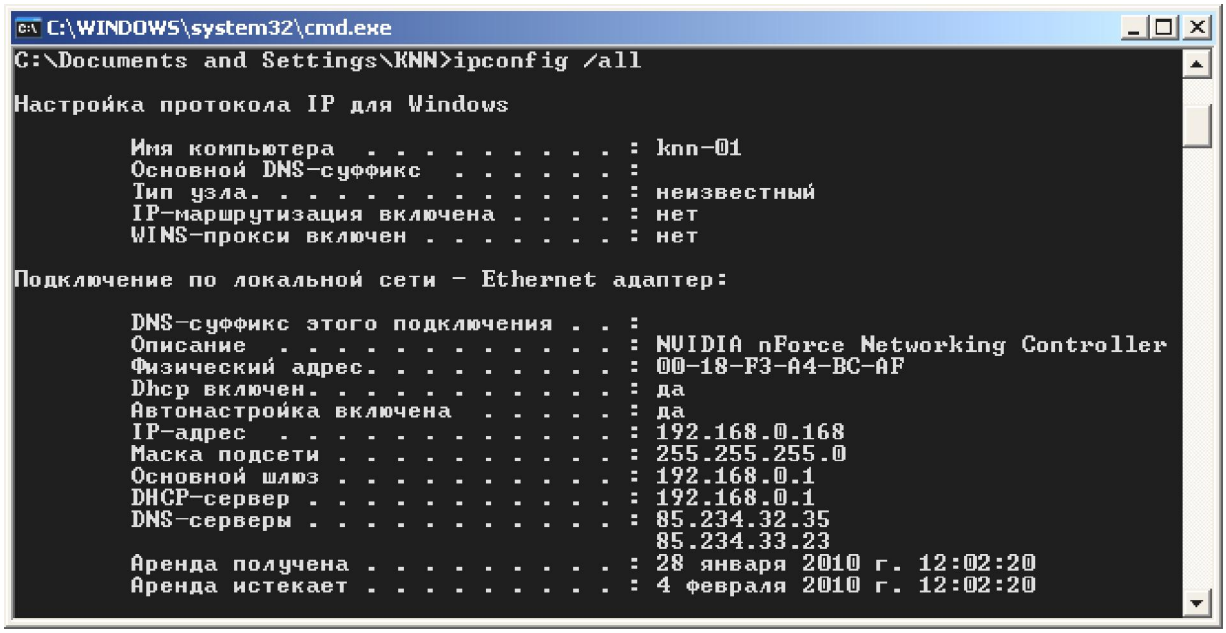


Рис. 1.1. Отображение установленных на компьютере сетевых конфигураций утилитой ***ipconfig***

### 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***] ***destination-list.*** 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 параметром ***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.

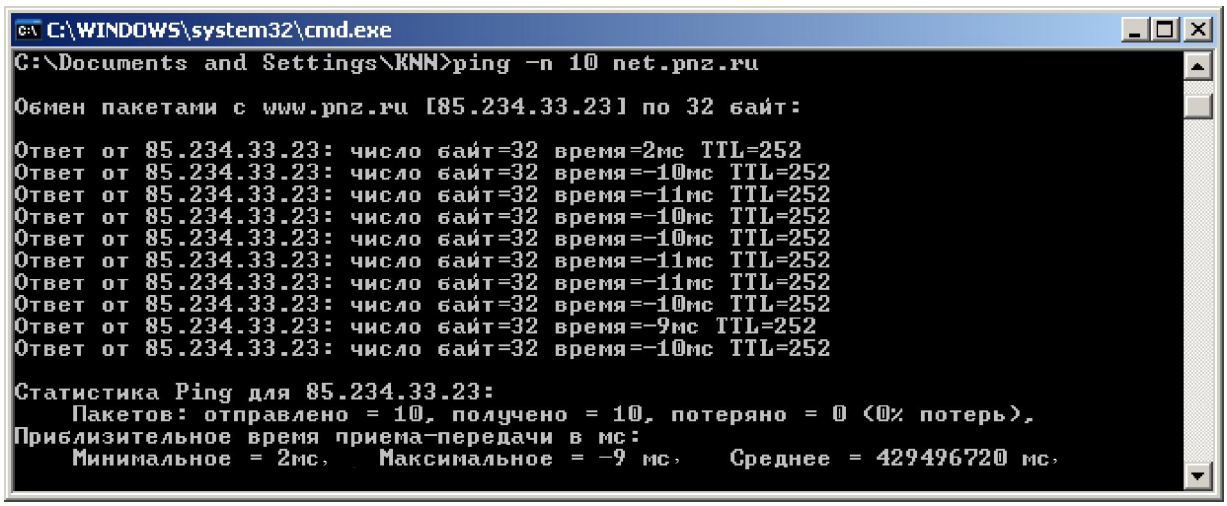


Рис. 1.2. Пример использования утилиты ***ping***

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.

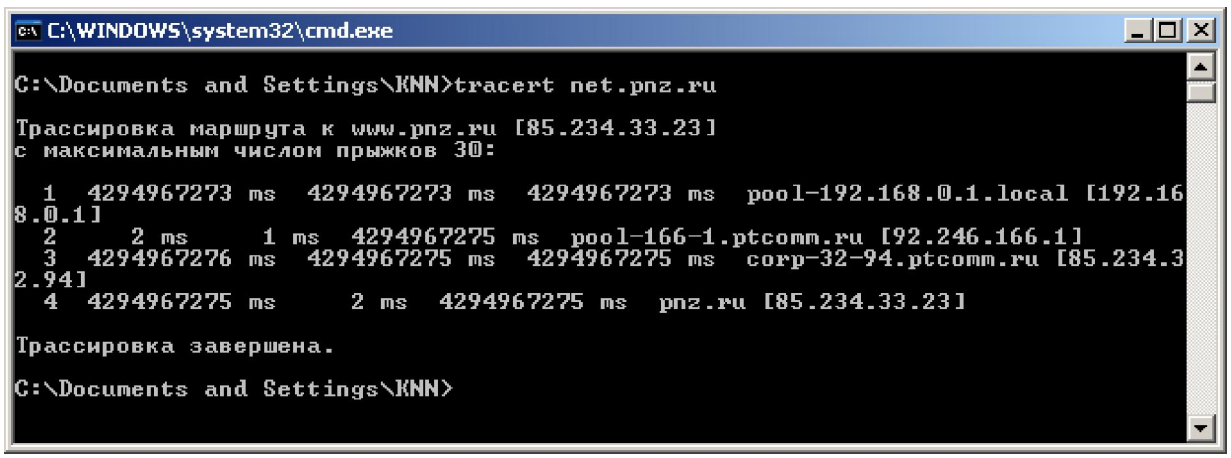


Рис. 1.3. Пример использования утилиты ***tracert***

### 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.

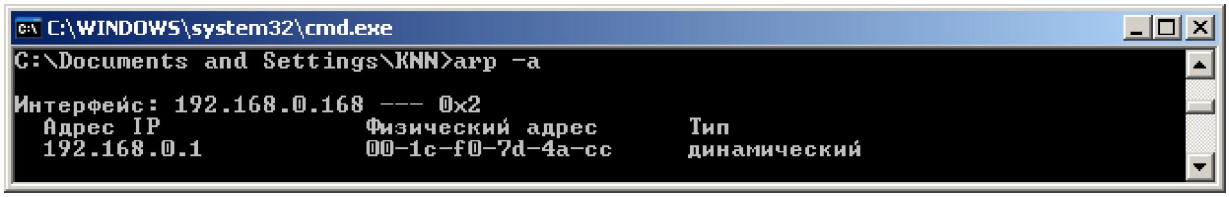


Рис. 1.4. Пример использования утилиты ***arp***

### 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).

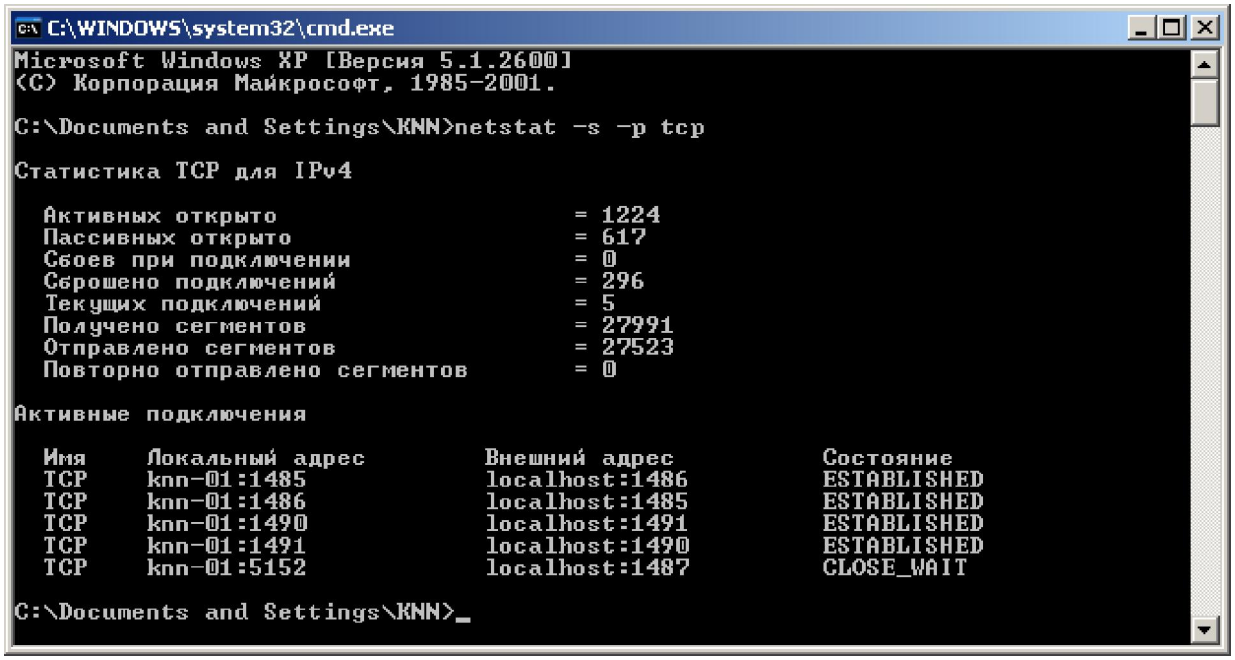


Рис. 1.5. Пример отображения утилитой ***netstat*** установленных на компьютере TCP-соединений

Открытые TCP-порты обозначаются в колонке «Состояние» строкой ***LISTENING*** – пассивно открытые соединения («слушающие» сокеты) или ***ESTABLISHED*** – установленные соединения, т.е. уже используемые сетевыми сервисами. Содержание состояний протокола TCP (всего имеется 11 состояний) раскрыто в лабораторной работе № 2 настоящего практикума.

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.

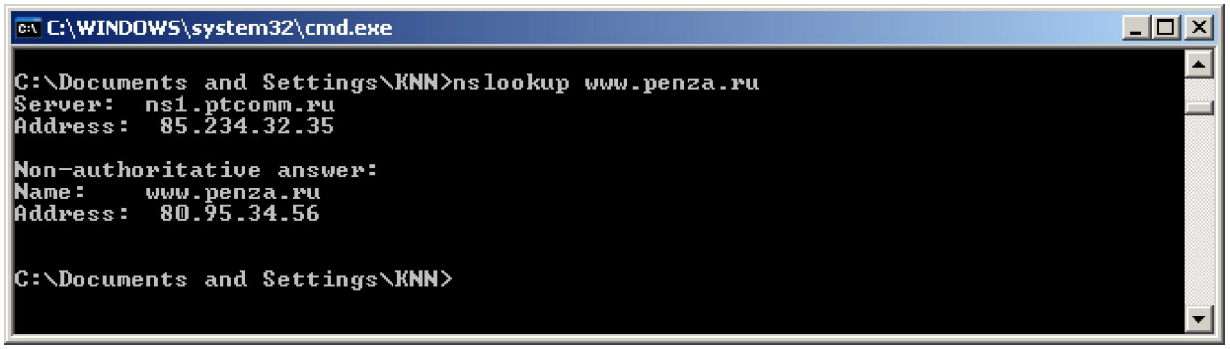


Рис. 1.6. Пример отображения утилитой ***nslookup*** запроса к DNS

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 ***Non-authoritative answer***, означающее, что ответ получен не с DNS-сервера, ответственного за зону ***penza.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.

## 1.3. 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*

**Доменные имена удаленных серверов**

| № | Адрес | № | Адрес |
|---|---|---|---|
| 1 | *net.pnz.ru* | 7 | *penza.ertelecom.ru* |
| 2 | *penza.vt.ru* | 8 | *mypenza.ru* |
| 3 | *www.trans-link.ru* | 9 | *penza.com.ru* |
| 4 | *penzartc.ru* | 10 | *www.penza-gsm.ru* |
| 5 | *www.penza.ru* | 11 | *www.zato.ru* |
| 6 | *www.ptcomm.ru* | 12 | *penza.citydom.ru* |

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. Провести пингование какого либо хоста локальной сети, адрес которого не был отражен в кэше. Повторно открыть ARP-кэш и проконтролировать модификацию его содержимого. 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 необхо-

димо 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 и контактные данные администратора (тел., e-mail). 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 (по указанию преподавателя);
- 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.

## 1.4. Вопросы для самопроверки

1. Каковы назначение и форматы MAC- и IP-адресов? С какой целью применяется «маска подсети»?

2. Как по IP-адресу и маске одной из рабочих станций определить адрес, принадлежащий всей локальной сети?

3. Как определить MAC-адрес сетевого адаптера, установленного в компьютере?

4. Что такое «основной шлюз»?

5. Каким образом утилита ***ping*** проверяет соединение с удаленным хостом?

6. Сколько промежуточных маршрутизаторов сможет пройти IP-пакет, если его время жизни равно 30?

7. Как работает утилита ***tracert***?

8. Каково назначение утилиты ***arp***?

9. С помощью каких утилит можно определить по доменному имени хоста его IP-адрес?

10. Как утилита ***ping*** разрешает имена хостов в IP-адреса?

11. Какие могут быть причины неудачного завершения ***ping*** и ***tracert***?

12. Какая служба позволяет узнать символьное имя хоста по его IP-адресу?

13. Какие операции можно выполнить с помощью утилиты ***netstat***?

# Лабораторная работа № 2

# Анализ протоколов сетевого и транспортного уровней

## 2.1. Цель работы

**Целью работы** является изучение протоколов сетевого и транспортного уровней стека TCP/IP и приобретение практических навыков в использовании программных средств, позволяющих контролировать сетевой трафик, на примере программы *Network Monitor*.

## 2.2. Теоретический материал

### 2.2.1. Описание стека протоколов TCP/IP

#### 2.2.1.1. Архитектура семейства протоколов TCP/IP

Семейство протоколов TCP/IP представляет собой промышленный стандарт стека протоколов, разработанный для локальных и глобальных сетей. Лидирующая роль стека TCP/IP объясняется следующими причинами:

- это наиболее завершенный стандартный, имеющий многолетнюю историю;
- стек TCP/IP поддерживают все современные операционные системы;
- это гибкая технология для соединения разнородных систем как на уровне транспортных подсистем, так и на уровне прикладных сервисов;
- это устойчивая масштабируемая межплатформенная среда для сетевых клиент-серверных приложений.

Так как стек TCP/IP был разработан до появления модели взаимодействия открытых систем OSI, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI приведено в табл. 2.1.

Протоколы TCP/IP делятся на четыре уровня – в стеке TCP/IP верхние три уровня (прикладной, представительский и сеансовый) модели OSI объединяют в один – прикладной. Нижний уровень (уровень доступа к сети) в стеке TCP/IP специально не регламентируется, но поддерживает все популярные стандарты физического и канального уровней.

*Таблица 2.1*

**Соответствие уровней стека TCP/IP уровням модели OSI**

| **Модель TCP/IP** | | Протоколы | **Модель OSI** | |
|---|---|---|---|---|
| № | Наименование уровня | | Наименование уровня | № |
| 1 | Прикладной | HTTP, FTP, DNS, SMTP, SNMP, Telnet | Прикладной | 7 |
| | | | Представительский | 6 |
| | | | Сеансовый | 5 |
| 2 | Транспортный | TCP, UDP, RTP, SCTP, DCCP | Транспортный | 4 |
| 3 | Межсетевой | IP, ICMP, IGMP | Сетевой | 3 |
| 4 | Уровень доступа к сети | Специально не определен (Ethernet, PPP, Frame Relay и др.) | Канальный | 2 |
| | | | Физический | 1 |

Важно отметить, что в состав стека входят протоколы маршрутизации, которые функционально принадлежат сетевому уровню, но работают поверх его (например, RIP работает поверх UDP, OSPF – поверх IP, BGP – поверх TCP), поэтому их невозможно вписать в модель.

К основным протоколам стека TCP/IP относятся:

- ARP – отвечает за получение MAC-адреса хоста, размещенного в текущей сети, по его IP-адресу;
- ICMP – обеспечивает посылку сообщений об ошибках, обнаруженных в процессе передачи пакетов;
- IP – обеспечивает маршрутизацию пакетов;
- TCP – обеспечивает соединение между двумя хостами, с гарантируемой доставкой пакетов;
- UDP – обеспечивает соединение между двумя хостами, при котором не гарантируется доставка пакетов.

Протоколы передают данные вниз по стеку протоколов при отправке в сеть и вверх по стеку при получении из сети, используя принцип инкапсуляции (вложенности) (рис. 2.1).

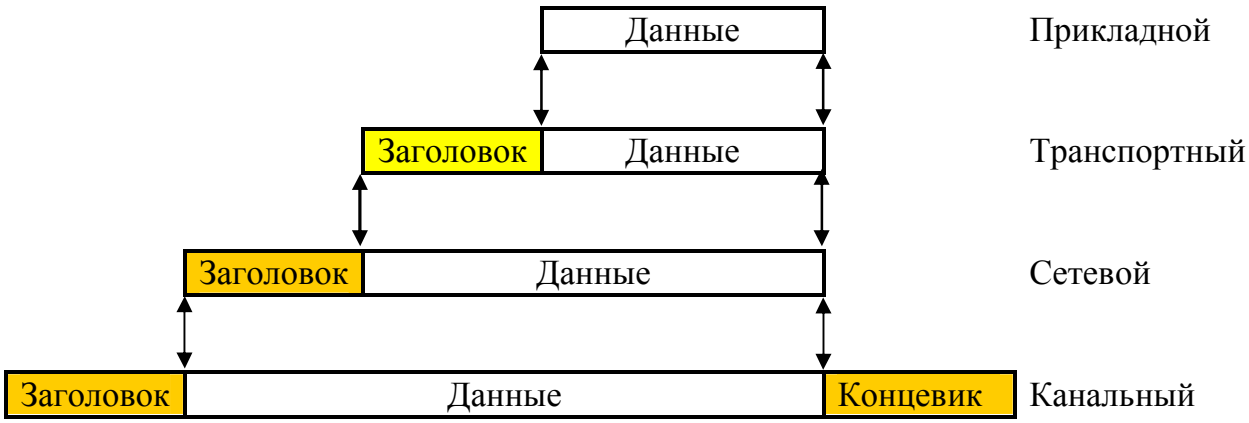


Рис. 2.1. Принцип инкапсуляции в стеке протоколов

При передаче данных каждый уровень формирует сообщение стандартного формата, в которое помещает служебную информацию (заголовок) и, возможно, передаваемые данные. Затем это сообщение направляется нижестоящему уровню, который, в свою очередь, добавляет свой заголовок и т.д. Протоколы канального уровня обычно предусматривают наличие в сообщении не только заголовка, но и концевика (обычно контрольная сумма). Наконец, сообщение достигает самого низкого, физического уровня, который действительно передает его по линиям связи.

Когда сообщение по сети поступает на другую машину, оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует, обрабатывает и удаляет заголовок своего уровня, выполняет соответствующие данному уровню функции и передает сообщение вышележащему уровню.

Каждому уровню TCP/IP соответствует собственная структура данных и специальная терминология ее описания:

- на прикладном уровне направляемые приложением в протокол TCP данные считают потоком, а приложения UPD – сообщением;
- на транспортном уровне данные TCP хранятся в сегментах, а данные UPD – в пакетах;
- сетевой уровень рассматривает данные в качестве блоков, называемых дейтаграммами или пакетами;
- в многочисленных типах канальных протоколов, поверх которых работает TCP/IP, блок данных именуется кадром.

### 2.2.1.2. Протокол **ARP**

Протокол определения адреса ARP (*Address Resolution Protocol*) устанавливает соответствие между сетевым и канальным адресами узла назначения в пределах локальной сети. Каждый раз, когда хост-отправитель должен переслать сформированный IP-пакет, он запускает протокол ARP для определения по заданному сетевому адресу MAC-адреса хоста назначения.

Работа протокола ARP начинается с просмотра ARP-таблицы, в которой каждая строка указывает соответствие между IP-адресом и MAC-адресом. Если искомый адрес в ARP-таблице отсутствует, то протокол ARP посылает широковещательный кадр, в который вложен ARP-запрос, смысл которого «если Вы владелец этого IP-адреса, сообщите мне Ваш MAC-адрес». Структура ARP-пакета приведена в табл. 2.2.

*Таблица 2.2*

**Структутра ARP-пакета**

| Название поля | Размер (байт) | Описание |
|---|---|---|
| *Hardware Type* | 2 | Идентификатор типа протокола передачи данных канального уровня (1 для Ethernet) |
| *Protocol Type* | 2 | Идентификатор типа сетевого протокола (800h для IP) |
| *Hardware Address Length* | 1 | Длина MAC-адреса в байтах (6 для Ethernet и Token Ring) |
| *Protocol Address Length* | 1 | Длина адреса для используемого протокола (для IP составляет 4 байта) |
| *Operation* | 2 | Код выполняемой операции (1 в случае запроса и 2 в случае ответа) |
| *Sender Hardware Address* | 6 | MAC-адрес отправителя |
| *Sender IP Address* | 4 | IP-адрес отправителя |
| *Target Hardware Address* | 6 | MAC-адрес получателя (при запросе поле пусто) |
| *Target IP Address* | 4 | IP-адрес получателя |

Все узлы локальной сети получают ARP-запрос и сравнивают указанный в нем IP-адрес с собственным адресом. Хост, фиксирующий их совпадение, формирует ARP-ответ, в котором указывает свои IP- и MAC-адреса, и посылает его персонально отправителю ARP-запроса.

После получения ARP-ответа хост, инициировавший ARP-запрос, добавляет запись в свою ARP-таблицу, и IP-дейтаграмма, из-за которой начался обмен ARP-пакетами, может быть отослана.

В случае отсутствия ARP-ответа считается, что хост назначения не доступен в локальной сети, и поэтому и исходная IP-дейтаграмма должна быть направлена в шлюз, MAC-адрес которого определяется также с помощью протокола ARP.

### 2.2.1.3. Протокол **IP**

Основными функциями протокола межсетевого взаимодействия IP (*Internet Protocol*) являются:

- определение формата дейтаграммы – базовой единицы передачи в сетях Internet;
- перенос между сетями различных типов адресной информации в унифицированной форме;

• обмен данными между уровнем доступа к сети и транспортным уровнем;

• маршрутизация дейтаграмм, адресованных удаленным узлам;

• разборка (фрагментация) и сборка пакетов при передаче их между сетями с различным максимальным значением длины пакета.

Пакет IP состоит из заголовка и поля данных. Заголовок пакета имеет поля, показанные в табл. 2.3.

*Таблица 2.3*

**Поля заголовка пакета**

| Название поля | Размер (бит) | Описание |
|---|---|---|
| *Version* | 4 | Версия протокола. Текущая версия 4 |
| *Header Length* | 8 | Количество 32-битных слов в заголовке пакета. Минимальный размер заголовка 20 байт (5 слов). Наличие поля *Options* может увеличить размер заголовка максимум на 4 байта |
| *Type of Service* | 8 | Желаемое качество обслуживания пакета при его доставке (задает приоритетность пакета и вид критерия выбора маршрута) |
| *Total Length* | 16 | Общая длина IP пакета |
| *Identifier* | 16 | Идентификатор пакета. Если пакет фрагментирован, то все фрагменты имеют одинаковый идентификатор. Это необходимо для восстановления исходного пакета |
| *Fragmentation Flags* | 3 | Флаги управления фрагментацией (один всегда равен 0) |
| *Fragment Offset* | 13 | Смещение – позиция фрагмента внутри пакета. Если пакет не фрагментирован, то смещение равно 0 |
| *Time to Live* | 8 | Время жизни пакета, в течение которого он может находиться в сети. Маршрутизаторы уменьшают значение этого поля на то время, которое пакет находится на нем (обычно на 1). По истечении TTL пакет уничтожается |
| *Protocol* | 8 | Идентификатор интернет-протокола следующего уровня (например, 1-ICMP, 4-IP, 6-TCP, 17-UDP) |
| *Header Checksum* | 16 | Контрольная сумма заголовка |
| *Source Address* | 32 | IP-адрес отправителя пакета |
| *Destination Address* | 32 | IP-адрес получателя пакета |
| *Options* | | Необязательное поле для дополнительной информации. Переменное число 32-битных слов; если это поле заполнено не полностью, то незаполненная часть дополняется нулями |

IP работает без создания логических соединений между хостами: для передачи IP-пакетов получателям он использует адреса, помещенные в заголовок пакетов, не требуя при этом подтверждения получения данных, т.е. отправитель и получатель не информируются о пропаже пакета или неправильной последовательности получения пакетов.

Маршрутизаторы и коммутаторы третьего уровня считывают записанную в IP-пакетах информацию и используют ее совместно с таблицами маршрутизации и некоторыми другими интеллектуальными средствами поддержки работы сети, пересылая данные по сетям TCP/IP любого масштаба. Механизм IP- маршрутизации:

- маршрутизатор проверяет IP-адрес входящего пакета и просматривает таблицу, определяя, не является ли пунктом назначения локальная сеть;
- если IP-адрес назначения локальный, то маршрутизатор находит во внутреннем хранилище IP- и MAC-адресов локальных устройств MAC-адрес места назначения, помещает его в заголовок пакета и направляет пакет получателю;
- если MAC-адрес получателя не обнаруживается, маршрутизатор должен послать запрос о нем по IP-адресу получателя. Если после просмотра таблицы выясняется, что пакет не предназначен для локальной сети, маршрутизатор переправляет его маршрутизатору следующего сетевого сегмента, используя MAC-адрес последнего.

Процесс построения и обновления таблиц маршрутизации практически непрерывен. Он осуществляется либо вручную администратором (статическая маршрутизация), либо средствами, использующими интеллектуальные протоколы, например RIP, OSPF или др. (динамическая маршрутизация).

В таблице каждого маршрутизатора указан оптимальный маршрут до адреса назначения или до маршрутизатора следующего сегмента сети. Последовательно просматривая собственные таблицы маршрутизации, соответствующие устройства передают пакет дальше, запрашивая, при необходимости, MAC-адрес конечной станции. Этот процесс продолжается до тех пор, пока пакет не дойдет до пункта назначения.

При пересылке пакета через множество сетевых сегментов существует опасность образования «петель»: неправильно сконфигурированный маршрутизатор постоянно возвращает пакет тому маршрутизатору, через который данный пакет уже проходил. Во избежание этого в IP предусмотрена TTL-функция (*Time-to-live*), позволяющая

задать предел времени движения пакета по сети. Значение TTL устанавливается заранее и уменьшается на единицу при каждом прохождении любого маршрутизатора. Если величина TTL становится равной нулю, пакет удаляется, а маршрутизатор отсылает отправителю специальное сообщение ICMP.

Важной функцией IP-протокола является управление фрагментацией. Физический сетевой уровень обычно накладывает ограничение на размер кадра, который может быть передан. Маршрутизатор получает IP-пакет, который необходимо отправить, определяет, на какой локальный интерфейс отправляется (маршрутизируется) дейтаграмма, и запрашивает интерфейс, чтобы тот сообщил размер установленной максимальной единицей передачи MTU (*Maximum Transmission Unit*). IP сравнивает MTU с размером дейтаграммы и осуществляет, если необходимо, фрагментацию.

Когда IP-дейтаграмма фрагментируется, каждый фрагмент становится пакетом с собственным IP-заголовком и маршрутизируется независимо от других пакетов. Поэтому возможно, что дейтаграммы прибудут в конечный пункт назначения в другом порядке, нежели они были исходно отправлены и фрагментированы. Однако в IP-заголовке хранится информация для того, чтобы дейтаграмма была собрана корректно.

Фрагментированная дейтаграмма не собирается вновь до тех пор, пока не достигнет конечного пункта назначения, на котором осуществляется сборка, поэтому существует вероятность, что фрагмент дейтаграммы будет снова фрагментирован (возможно, даже несколько раз).

В процессе фрагментации используются следующие поля IP-заголовка:

- поле идентификации, которое содержит уникальное для каждой отправленной IP-дейтаграммы значение, копируемое в каждый фрагмент конкретной дейтаграммы;
- поле флагов (три бита), в котором установленный бит DF (*Do not Fragment*) запрещает маршрутизатору фрагментировать данный пакет, установленный бит MF (*More Fragments*) говорит о том, что пакет переносит промежуточный фрагмент (устанавливается в единицу для каждого фрагмента, кроме последнего);
- поле смещения фрагмента (*Fragment Offset*) содержит смещение этого фрагмента от начала исходной дейтаграммы. Когда дейтаграмма фрагментируется, поле полной длины каждого фрагмента изменяется так, чтобы соответствовать размеру фрагмента.

### 2.2.1.4. Протокол **ICMP**

Протокол ICMP (*Internet Control Message Protocol*) относится к сетевому уровню модели TCP/IP и предназначается для передачи управляющих и диагностических сообщений. Сообщения ICMP генерируются и обрабатываются протоколами сетевого (IP) и более высоких уровней (TCP или UDP) и выполняют следующие информативные, управляющие и связанные с ошибками функции:

- управление потоками данных;
- обнаружение недостижимых адресатов;
- перенаправление маршрутов;
- проверка состояния удаленных узлов.

ICMP-сообщения передаются внутри IP-пакетов и имеют формат, приведенный в табл. 2.4. Заголовок ICMP включает восемь байт, но только первые четыре байта одинаковы для всех сообщений, остальные поля заголовка и тела сообщения определяются типом сообщения. Поле контрольной суммы охватывает ICMP-сообщения целиком.

*Таблица 2.4*

**Формат ICMP-сообщений**

| Название поля | Размер(бит) | Описание |
|---|---|---|
| *Type* | 8 | Поле, содержащее тип ICMP-пакета |
| *Code* | 8 | Поле, содержащее код (номер) функции соответствующего типа сообщения. Если тип имеет только одну функцию, то значение поля равно 0 |
| *Checksum* | 16 | Контрольная сумма |
| *Type Specific Data* | ... | Дополнительные данные, индивидуальные для каждого типа пакета |

Типы и коды ICMP-сообщений приведены в табл. 2.5.

*Таблица 2.5*

**Типы и коды**

| Типы | Коды | Описание |
|---|---|---|
| 1 | 2 | 3 |
| 0 | | Эхо-ответ (ping-отклик) |
| 3 | | Адресат недостижим |
| | 0 | * Сеть недостижима |
| | 1 | * ЭВМ недостижима |
| | 2 | * Протокол недоступен |
| | 3 | * Порт недоступен |
| | 4 | * Необходима фрагментация сообщения |

*Окончание табл. 2.5*

| 1 | 2 | 3 |
|---|---|---|
| | 5 | * Исходный маршрут вышел из строя |
| | 6 | * Сеть места назначения неизвестна |
| | 7 | * ЭВМ места назначения неизвестна |
| | 8 | * Исходная ЭВМ изолирована |
| | 9 | * Связь с сетью места назначения административно запрещена |
| | 10 | * Связь с ЭВМ места назначения административно запрещена |
| | 11 | * Сеть недоступна для данного вида сервиса |
| | 12 | * ЭВМ недоступна для данного вида сервиса |
| | 13 | * Связь административно запрещена с помощью фильтра |
| | 14 | * Нарушение старшинства ЭВМ |
| | 15 | * Дискриминация по старшинству |
| 4 | 0 | * Отключение источника при переполнении очереди |
| 5 | | Переадресовать (изменить маршрут) |
| | 0 | Переадресовать дейтаграмму в сеть (устарело) |
| | 1 | Переадресовать дейтаграмму на ЭВМ |
| | 2 | Переадресовать дейтаграмму для типа сервиса (tos) и сети |
| | 3 | Переадресовать дейтаграмму для типа сервиса и ЭВМ |
| 8 | 0 | Эхо запроса (ping-запрос) |
| 9 | 0 | Объявление маршрутизатора |
| 10 | 0 | Запрос маршрутизатора |
| 11 | | Для дейтаграммы время жизни истекло (ttl=0): |
| | 0 | *при передаче |
| | 1 | * при сборке (случай фрагментации) |
| 12 | | * Проблема с параметрами дейтаграммы |
| | 0 | * Ошибка в ip-заголовке |
| | 1 | * Отсутствует необходимая опция |
| 13 | | Запрос временной метки |
| 14 | | Временная метка-отклик |
| 15 | | Запрос информации (устарел) |
| 16 | | Информационный отклик (устарел) |
| 17 | | Запрос адресной маски |
| 18 | | Отклик на запрос адресной маски |

Примечание. «*» отмечены сообщения об ошибках, остальные сообщения являются запросами.

Сообщения об ошибках ICMP обрабатываются специальным образом. Например, ICMP-сообщение об ошибке никогда не генерируется в ответ на ICMP-сообщение об ошибке. Кроме того, ICMP-сообщение об ошибке всегда содержит IP-заголовок и первые восемь байт

IP-дейтаграммы, вызвавшей генерацию этого сообщения, что позволяет принимающему приложению с помощью номера порта, который содержится в первых байтах заголовков TCP и UDP, установить соответствие между полученным сообщением ICMP и конкретным пользовательским процессом. Например, формат IP-пакета с сообщения о недоступности порта UDP имеет вид, показанный на рис. 2.2.

| Заголовок IP-пакета | Заголовок ICMP Тип = 3 Код = 3 | Заголовок IP-пакета, сгенерировавшего ошибку | Заголовок UDP |
|---|---|---|---|
| 20 байт | 8 байт | 20 байт | 8 байт |

Рис. 2.2. ICMP-сообщение «порт UDP недоступен»

### 2.2.1.5. Протокол **TCP**

Протокол управления передачей TCP (*Transmission Control Protocol*) обеспечивает надежную транспортировку данных между прикладными процессами, запущенными на хостах, путем установления логического соединения. Протокол TCP находится на транспортном уровне стека TCP/IP, между протоколом IP и собственно приложением и гарантирует, что приложение получит данные без потерь и точно в такой же последовательности, в какой они были отправлены.

Протокол транспортного уровня инкапсулирует данные, полученные от протоколов прикладного уровня, добавляя к ним свой заголовок. Если протоколы прикладного уровня передают TCP больше данных, чем вмещает отдельный пакет, то TCP разбивает данные на несколько сегментов, совокупность которых, пересылаемая за одно соединение (транзакцию), называется последовательностью. К каждому сегменту добавляется собственный заголовок TCP (табл. 2.6), после чего он передается на сетевой уровень для передачи в отдельной IP-дейтаграмме. Когда сегменты достигают целевого компьютера, TCP восстанавливает из них исходную последовательность, передаваемую прикладному процессу.

Протокол TCP обеспечивает работу одновременно нескольких приложений. Приложение (процесс), использующее TCP, однозначно определяется числом – номером порта. Заголовок TCP-сегмента содержит номера портов процесса-отправителя и процесса-получателя. При получении сегмента модуль TCP анализирует номер порта получателя и отправляет данные соответствующему прикладному процессу.

*Таблица 2.6*

**TCP-заголовок**

| Название поля | Размер (бит) | Описание |
|---|---|---|
| *TCP Source Port* | 16 | Порт хоста, отправившего пакет |
| *TCP Destination Port* | 16 | Порт хоста, получающего пакет |
| *Sequence Number* | 32 | Номер последовательности, соответствующий данному сегменту (порядковый номер первого байта в сегменте) |
| *Acknowledgment Number* | 32 | Номер последовательности подтверждения, указывает на байт, который хост хочет получить следующим |
| *Data Length* | 4 | Количество 32-битных слов в TCP заголовке |
| *Reserved* | 6 | Зарезервировано (заполняется нулями) |
| *Flags* | 6 | Флаги:<br>**URG –** срочности,<br>**ACK** – подтверждения,<br>**PSH** – функции проталкивания,<br>**RST** – перезагрузки данного соединения,<br>**SYN –** синхронизации номеров очереди,<br>**FIN –** завершения отправки данных |
| *Window* | 16 | Размер окна, указывает на число байт, которое готов принять отправитель сегмента TCP |
| *Checksum* | 162 | Контрольная сумма |
| *Urgent Pointer* | 16 | Указатель срочности. Если отправляется срочный пакет (это определяется флагом **URG**), то здесь хранится указатель на положение срочных данных в сегменте |
| *Options* | | Необязательное поле для дополнительной информации. Переменное число (до 10) 32-битных слов (если это поле заполнено не полностью, то незаполненная часть дополняется нулями) |

Для облегчения взаимодействия между различными программами прикладного уровня приняты соглашения о номерах портов, закрепленных за определенными службами Internet. Номера портов наиболее известных служб сети приведены в табл. 2.7.

Номера портов от 0 до 255 зарезервированы под системные нужды, их не допускается использовать в прикладных программах. В ин-

тервале от 256 до 1023 многие порты также используются сетевыми службами, поэтому и их не рекомендуется применять для прикладных нужд. Как правило, большинство приложений, построенных на основе TCP/IP, использует номера портов в диапазоне от 1024 до 5000 (максимально возможным номером порта является число 65535). Рекомендуется использовать номера от 3000 до 5000, номера выше 5000 используются чаще всего для краткосрочного применения.

*Таблица 2.7*

**Номер портов служб сети**

| Номер порта | Служба сети | Описание |
|---|---|---|
| 0 | | Зарезервирован |
| 7 | *echo* | Эхо-ответ на входящие сообщения |
| 9 | *discard* | Сброс (поглощение) всех входящих сообщений |
| 11 | *users* | Активные пользователи |
| 13 | *daytime* | Отклик, содержащий время дня |
| 19 | *chargen* | Генератор символов |
| 20 | *ftp data* | Передача данных по протоколу FTP |
| 21 | *ftp* | Передача управляющих команд по протоколу FTP |
| 23 | *telnet* | Порт подключения по протоколу TELNET |
| 25 | *smtp* | Протокол передачи почтовых сообщений SMTP |
| 37 | *time* | Отклик, содержащий время |
| 42 | *name* | Сервер имен |
| 43 | *whois* | Кто это |
| 53 | *domain* | Сервер имен доменов |
| 67 | *boots* | Протокол удаленной загрузки сервера |
| 68 | *bootc* | Протокол удаленной загрузки клиента |
| 69 | *tftp* | Упрощенный протокол передачи файлов TFTP |
| 80 | *http* | Протокол передачи гипертекста HTTP |
| 109 | *pop2* | Протокол почтового ящика POP2 |
| 110 | *pop3* | Протокол почтового ящика POP3 |
| 111 | *rpc* | Протокол удаленного вызова процедур RPC |

Совокупность IP-адреса и номера порта называется сокетом. При соединении любая ЭВМ однозначно определена IP-адресом, а каждый процесс – портом, поэтому соединение между двумя процессами однозначно определяется сокетом. Например, сокет сервера электронной почты на хосте 194.84.124.4 обозначается как 194.84.124.4:25.

Взаимодействие приложений с использованием протокола TCP включает три этапа:

- установление логического соединения;
- обмен данными;
- закрытие соединения.

Протокол TCP ориентирован на соединение. Это означает, что до начала обмена данными прикладного уровня две системы должны установить связь между собой, что гарантирует существование обоих компьютеров, работу без сбоев и готовность к обмену данными. Соединение TCP сохраняется на протяжении всего обмена данными, а затем закрывается установленным образом. Фактически соединение TCP представляет собой два отдельных канала передачи данных, работающих в противоположных направлениях, т.е. TCP является полнодуплексным протоколом.

Рис. 2.3 иллюстрирует этап установления соединения, реализуемый как «трехшаговое рукопожатие» (*three-way handshake*).

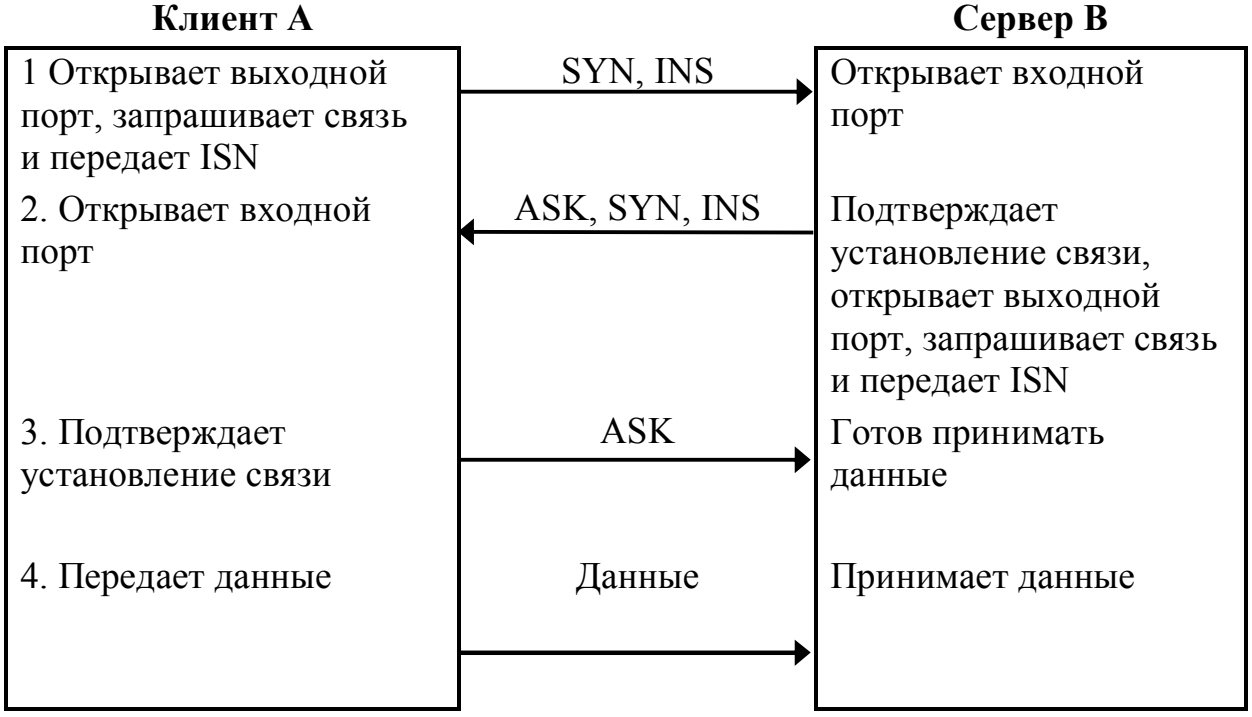


Рис. 2.3. Установка TCP-соединения

На первом шаге клиентская система А посылает серверу В пакет с установленным флагом SYN и начальным значением номера в последовательности ISN (*initial sequence number*). Сервер В, будучи готов установить соединение, отвечает TCP-пакетом, подтверждающим пра-

вильный прием запроса и информирующим о готовности установить соединение. Для этого поле «Номер последовательности подтверждения» устанавливается на 1 больше начального номера в поле последовательности TCP-заголовка пакета от системы A и устанавливается флаг ACK, а также взводится флаг SYN и устанавливается начальный номер в последовательности. На третьем шаге система A подтверждает правильность приема пакета от системы B. Следует отметить, что ни один из пакетов трехшагового рукопожатия не содержит данных прикладного уровня, передача которых начинается на четвертом шаге.

Помимо проверки существования другого компьютера и его готовности к приему данных, на этапе установления логического соединения согласовывается нумерация передаваемых сообщений, действующая на время соединения. Каждый компьютер выбирает для первого сегмента TCP начальный номер последовательности. Затем с каждым последующим сообщением системы увеличивают этот номер на 1. Для выбора ISN компьютеры используют специальный алгоритм, который минимизирует вероятность того, что для соединения между одной и той же парой сокетов одновременно будут использованы одинаковые номера последовательности.

Кроме того, с помощью сообщений SYN компьютеры также информируют друг друга о максимальном размере сегмента MSS (*maximum segment size*). Значение MSS зависит от того, какой протокол канального уровня используется в сети, где находится соответствующая система.По значению MSS для другой системы передающий компьютер определяет количество данных, которое можно включать в последующие сообщения. Величина MSS передается в виде четырех байтов, записанных в полях *Options* заголовков TCP двух пакетов SYN.

Завершение соединения сопровождается обменом пакетов с флагами FIN и ACK. Так как TCP-соединение полнодуплексное, каждое направление должно быть закрыто независимо от другого. При этом каждая сторона должна послать пакет с флагом FIN, когда передача данных завершена. Приняв FIN, протокол должен уведомить свое приложение, что удаленная сторона прекращает передачу данных в этом направлении (FIN обычно отправляется в результате того, что приложение было закрыто), и отправить назад ACK с принятым номером последовательности плюс один. Станция, получившая FIN, может все еще посылать данные. Когда приложение системы, получившей FIN, будет закрыто, TCP-протокол формирует и отсылает свой пакет с флагом FIN, на который должен быть получен ответ ACK. Таким образом, для закрытия соединения требуется четыре шага.

Соединение TCP во время функционирования проходит через серии промежуточных состояний, краткое описание которых содержится в табл. 2.8.

*Таблица 2.8*

**Описание состояний**

| Состояние | Содержание |
|---|---|
| *LISTEN* | Ожидание запроса на соединение со стороны чужих портов и программ TCP |
| *SYN-SENT* | Ожидание парного запроса на установление соединения, когда запрос уже сделан |
| *SYN-RECEIVED* | Ожидание подтверждения после того, как запрос соединения уже принят и отправлен |
| *ESTABLISHED* | Состояние открытого соединения, принимаемые данные можно представить пользователю. Это нормальное состояние соединения в фазе передачи данных |
| *FIN-WAIT-1* | Ожидание запроса от чужой программы TCP или подтверждения ранее отправленного запроса на закрытие соединения |
| *FIN-WAIT-2* | Ожидание запроса на закрытие соединения со стороны чужой программы TCP |
| *CLOSE-WAIT* | Ожидание запроса на закрытие соединения со стороны своего клиента |
| *CLOSING* | Ожидание подтверждения со стороны чужой программы TCP запроса о закрытии соединения |
| *LAST-ACK* | Ожидание запроса на закрытие соединения, ранее отправленного чужой программе TCP (запрос включал также подтверждение получения чужого запроса на закрытие соединения) |
| *TIME-WAIT* | Ожидание, когда истечет достаточное количество времени и можно быть уверенным, что чужая программа TCP получила подтверждение своего запроса на закрытие соединения |
| *CLOSED* | Состояние полного отсутствия соединения |

Протокол TCP обеспечивает защиту от повреждения, потери, дублирования и нарушения очередности получения данных. Для этого все байты в потоке данных сквозным образом пронумерованы в возрастающем порядке. Заголовок каждого сегмента содержит число байт данных в сегменте и порядковый номер первого байта той части потока, которая пересылается в данном сегменте. Например, если в сегменте пересылаются байты с номерами от 2001-го до 3000-го, то номер первого байта в данном сегменте равен 2001, а их количество равно 1000.

Номер первого байта в потоке определяется на этапе установления соединения и обозначается ISN+1. Например, ISN+1=1. Также для каждого сегмента вычисляется контрольная сумма, позволяющая обнаружить повреждение данных.

При удачном приеме сегмента данных получатель посылает отправителю подтверждение о приеме – номер удачно принятого байта плюс 1. Если в течение некоторого времени отправитель не получит подтверждения, считается, что сегмент не дошел или был поврежден, и он посылается снова. Этот механизм контроля надежности называется PAR (*Positive Acknowledgment with Retransmission*). Нумерация байтов используется также для упорядочения данных в порядке очередности и обнаружения дубликатов (которые могут быть посланы из-за большой задержки при передаче подтверждения или потери подтверждения).

Для ускорения и оптимизации процесса передачи больших объемов данных протокол TCP определяет метод управления потоком, называемый методом «скользящего окна», который позволяет отправителю посылать очередной сегмент, не дожидаясь подтверждения о получении в пункте назначения предшествующего сегмента.

Протокол TCP формирует подтверждения не для каждого конкретного успешно полученного пакета, а для всех данных от начала посылки до некоторого порядкового номера ACK SN (*Acknowledge Sequence Number*) исключительно. В качестве подтверждения успешного приема, например, первых 2000 байт, высылается ACK SN = 2001. Это означает, что все данные в байтовом потоке под номерами от ISN+1=1 до данного ACK SN-1 (2000) успешно получены.

Вместе с посылкой отправителю ACK SN получатель объявляет также размер окна, определяющий объем неподтвержденных данных, который отправителю разрешено передавать без квитанции от получателя.

Если объявлен размер окна 6000, то отправитель может посылать данные с порядковыми номерами от текущего ACK SN = 2001 до (ACK SN + размер окна – 1) = 8000, не дожидаясь подтверждения со стороны получателя. Размер окна может динамически изменяться получателем.

Для временной остановки посылки данных достаточно объявить нулевое окно. Но даже и в этом случае через определенные промежутки времени будут отправляться сегменты с одним байтом данных. Это делается для того, чтобы отправитель гарантированно узнал о том,

что получатель вновь объявил ненулевое окно: он обязан подтвердить получение пробных сегментов, а в этих подтверждениях должен указать также и текущий размер своего окна.

Когда протокол TCP передает в сеть сегмент, он «на всякий случай» помещает его копию в очередь повторной передачи и запускает таймер. Когда приходит квитанция на этот сегмент, соответствующая копия удаляется из очереди. Если же квитанция не приходит до истечения срока, то сегмент посылается повторно. Может случиться так, что повторный сегмент придет тогда, когда исходный сегмент уже окажется на месте, тогда дубликат будет попросту отброшен.

Если никакие данные приложениями не передаются, а соединение открыто, модуль TCP может периодически посылать сегменты-зонды для выяснения того, не отключилась ли другая сторона без уведомления партнера (например, в результате обрыва линии или другим некорректным образом).

#### 2.2.1.6. Протокол **UDP**

Протокол UDP (*User Datagram Protocol*) является более простым транспортным протоколом, чем протокол TCP. Протокол UDP обеспечивает доставку дейтограмм, но не требует подтверждения их получения. Поэтому он не требует установления соединения между передающим и принимающим процессами. Такая связь в принципе ненадежна, так как отправителю не сообщается, правильно ли принято его сообщение и получено ли оно вообще. Для проверки возникновения ошибок может использоваться контрольная сумма пакета, но ошибки никак не обрабатываются: они либо игнорируются, либо их обработка выполняется уже на более высоком, прикладном уровне.

В основном сообщения UDP применяются протоколами прикладного уровня DNS и DHCP. В определенных ситуациях UDP можно использовать и для передачи больших объемов данных, например, в аудио и видеоинформации. В данном случае использование UDP допустимо, поскольку мультимедийные приложения не критичны к потере определенной доли пакетов.

Заголовок UDP имеет длину 8 байт, т.е. значительно короче заголовка TCP, его структура приведена в табл. 2.9.

*Таблица 2.9*

**Структура заголовка UDP**

| Название поля | Размер (бит) | Описание |
|---|---|---|
| *UDP Source Port* | 16 | Порт хоста, отправившего пакет |
| *UDP Destination Port* | 16 | Порт хоста, получающего пакет |
| Length | 16 | Длина заголовка и данных UDP в байтах |
| Checksum | 16 | Контрольная сумма (не является обязательной) |

Сравнивая два протокола транспортного уровня, следует указать, что на TCP возложена сложная и очень важная задача обеспечения надежной передачи данных через ненадежную сеть, с другой стороны, функциональная простота протокола UDP обусловливает простоту алгоритма его работы, компактность и высокое быстродействие. Поэтому те приложения, в которых реализован собственный надежный механизм обмена сообщениями, основанный на установлении соединения, предпочитают для передачи данных использовать менее надежный, но более быстрый протокол UDP. Заметим также, что, поскольку протокол TCP основан на логических соединениях, он, в отличие от протокола UDP, не годится для широковещательной и групповой рассылки.

## 2.2.2. Работа с сетевым монитором

### 2.2.2.1. Общая характеристика сетевого монитора

Программа Сетевой монитор *Network Monitor* предназначена для сбора и анализа данных, передаваемых по вычислительной сети. Потребность в этом может возникнуть при наличии сбоев в работе сети, когда анализ передаваемых пакетов может помочь локализовать и устранить источник ошибок. Кроме того, утилита оказывается полезной при изучении сетевых протоколов, так как позволяет просмотреть полное содержимое всех передаваемых по сети пакетов с разбором формата их заголовков.

Программа *Network Monitor* фирмы Microsoft входит в стандартную поставку *Windows NT Server*, начиная с версии 4.0 (версия с усеченными функциями), и является частью управления сервера.

Сетевой монитор состоит из ядра, поддерживающего работу сетевого адаптера и программного обеспечения, декодирующего протокол канального уровня, с которым работает сетевой адаптер, а также наиболее распространенные протоколы верхних уровней, например IP, TCP, FTP, Telnet, HTTP, IPX, NCP, NetBEUI и др.

Сетевой монитор позволяет собирать данные, отправленные или полученные локальным компьютером из сетевого потока, и копировать их во временный файл записи. Сетевой монитор отображает статистику для собранных кадров динамически в окне сбора данных и позволяет создавать фильтр записи для кадров, удовлетворяющих определенным условиям. Сетевой монитор позволяет отображать, фильтровать, сохранять и печатать собранные данные.

### 2.2.2.2. Основное окно сетевого монитора

При захвате кадров сетевым монитором сведения о кадрах отображаются в окне сбора данных, разделенном на четыре области (панели) (рис. 2.4).

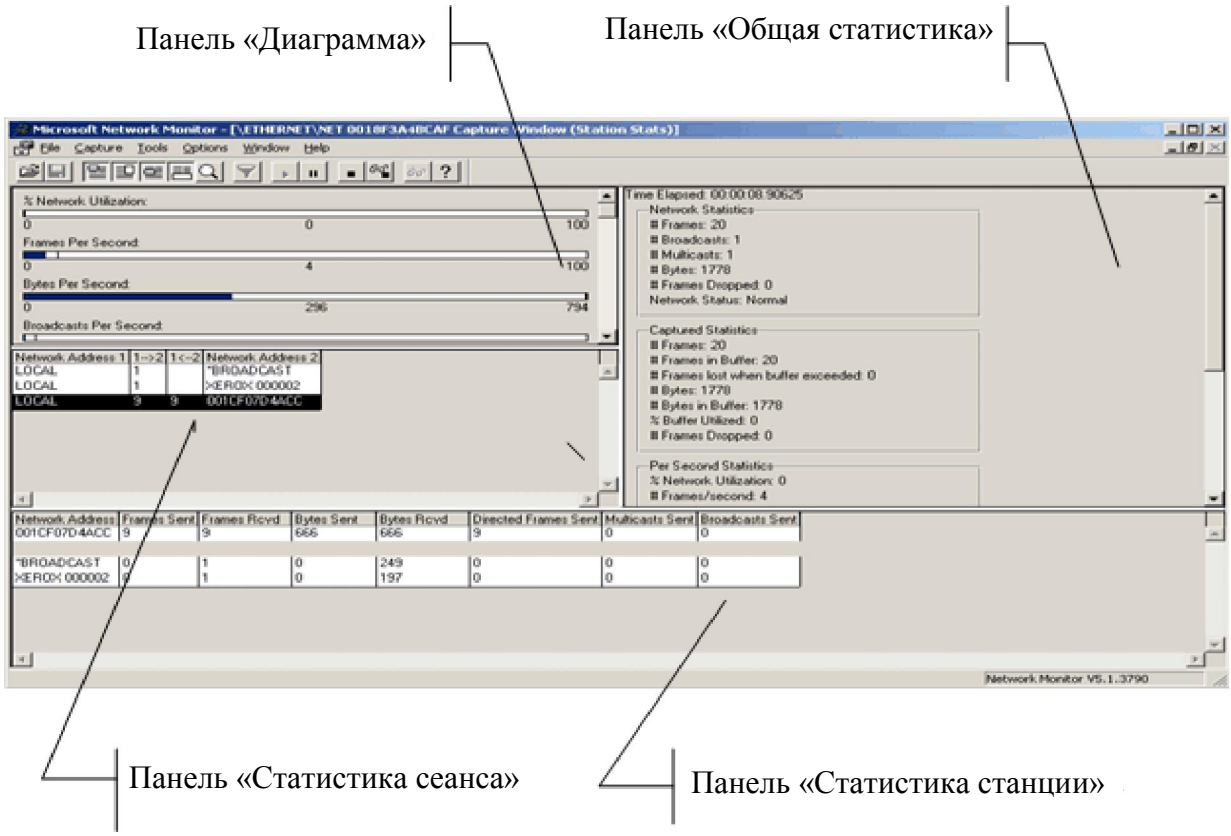


Рис. 2.4. Окно сбора данных

На панели «Диаграмма» представляются в реальном времени следующие сведения о собранных данных:

- использование сети в процентах (% *Network Utilization*);
- число кадров, передаваемых сетью каждую секунду (*Frames Per Second*);

- число байт, передаваемых сетью каждую секунду (*Bytes Per Second*);
- число широковещательных кадров, передаваемых сетью каждую секунду (*Broadcasts Per Second*);
- число многоадресных кадров, передаваемых сетью каждую секунду (*Multicasts Per Second*).

На панели «Статистика сеанса» отображается статистика для каждого сеанса (периода времени, в течение которого два компьютера соединяются и обмениваются данными): сетевые адреса участников каждого сеанса и количество данных, отправленных в любом направлении между ними.

На панели «Статистика станции» содержится статистика, отображается активность рабочей станции с определенным сетевым адресом в сети:

- число отправленных кадров, адресом;
- число полученных кадров;
- число отправленных байт;
- число полученных байт;
- число не широковещательных и не многоадресных кадров, переданных по сети соответствующим адресом;
- число отправленных многоадресных кадров;
- число отправленных широковещательных кадров.

На панели «Общая статистика» отображаются сведения об активности сети:

- ***Network Statistics*** – статистика сети, т.е. общий трафик сети с момента начала текущей записи, а также параметр ее состояния (***network status***), который для сети Ethernet всегда имеет значение ***Normal***;
- ***Captured Statistics*** **–** статистические сведения, описывающие процесс сбора (захвата) данных: общее число кадров и байт, захваченных анализатором, общее число кадров и байт, записанных в буфер, число кадров, пропущенных при переполнении буфера, процент используемого выделенного пространства буфера, число кадров, пропущенных сетевым монитором;
- ***Per Second Statistics*** – «статистика за секунду», т.е. показатели, рассчитываемые с момента начала сбора данных и постоянно обновляющиеся для отражения активности в данную секунду: среднее число кадров и байт в секунду, среднее число широковещательных и многоадресных сообщений в секунду, процент использования сети;

• ***Network Card*** (***MAC***) ***Statistics*** – статистика, которая отражает активность, обнаруженную сетевым адаптером с момента начала текущего сбора данных: общее число кадров и байт, обнаруженных сетевым адаптером, общее число широковещательных и многоадресных кадров, обнаруженных сетевым адаптером;

• ***Network Card*** (***MAC***) ***Error Statistics*** – статистика, которая отражает ошибки сетевого адаптера, произошедшие с момента начала сбора данных: число ошибок, вызванных несовпадением контрольной суммы (CRC) со значением CRC, вычисленным по полученным байтам данных, число кадров, обнаруженных сетевым адаптером, но пропущенных сетевым монитором вследствие нехватки места в буфере, число кадров, обнаруженных сетевым адаптером, но потерянных из-за аппаратных ограничений.

Примечание. Не все сетевые карты поддерживают ведение статистики, представленной последними двумя группами. Если данные о каких-либо величинах этих групп недоступны, в соответствующем пункте окна статистики появляются сообщения *Unsupported* или *Unknown Stat*.

### 2.2.2.3. Настройка Сетевого монитора

Сначала необходимо выбрать сетевой интерфейс, трафик которого должен контролироваться. Это делается с помощью меню ***Capture*** (пункт ***Networks***). В появившейся панели отображаются все сетевые интерфейсы, которые имеются у компьютера, – сетевые адаптеры (если их несколько) и COM-порты службы RAS, если она установлена на компьютере (рис. 2.5). Интерфейс RAS отличается от сетевых адаптеров тем, что он имеет MAC-адрес, равный 000000000000.

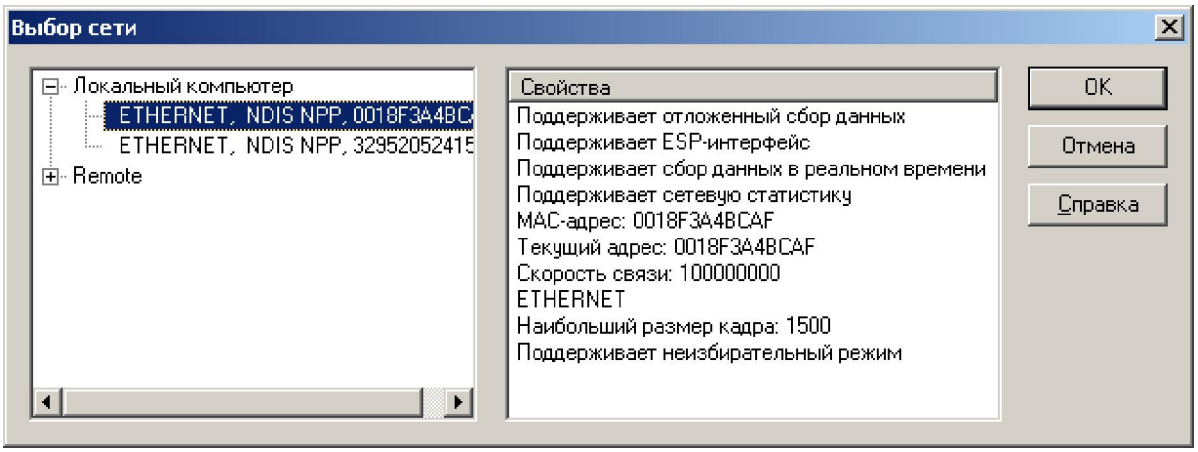


Рис. 2.5. Выбор сетевого интерфейса

Перед началом сбора статистики следует установить объем буфера для собранных пакетов. Объем буфера определяет количество пакетов, которые утилита сможет сохранить для последующего отображения. Для установки объема буфера следует выбрать пункт ***Buffer Settings***... в меню ***Capture***. При этом на экран будет выведено окно, вид которого представлен на рис. 2.6.

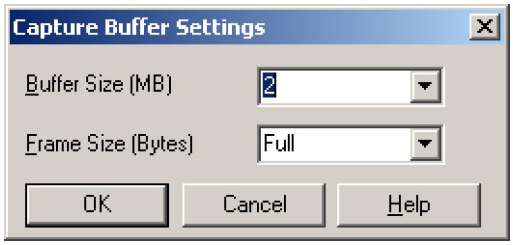


Рис. 2.6. Установка объема буфера

Значение ***Frame Size*** определяет размер, до которого будут обрезаться пакеты, если их длина больше этого значения. В этом поле следует оставить значение ***Full***. Поле ***Buffer Size*** собственно и определяет объем буфера в мегабайтах, обычно достаточно установить величину в 2–3 MB.

Для отображения мнемонических имен имена компьютеров, с которых были захвачены кадры, в адреса сетевого монитора используется база данных адресов. С помощью пункта **Address**… в меню ***Capture*** открывается окно ***Address Database***, позволяющее выполнять добавление, удаление или редактирование информации в базе данных адресов (рис. 2.7).

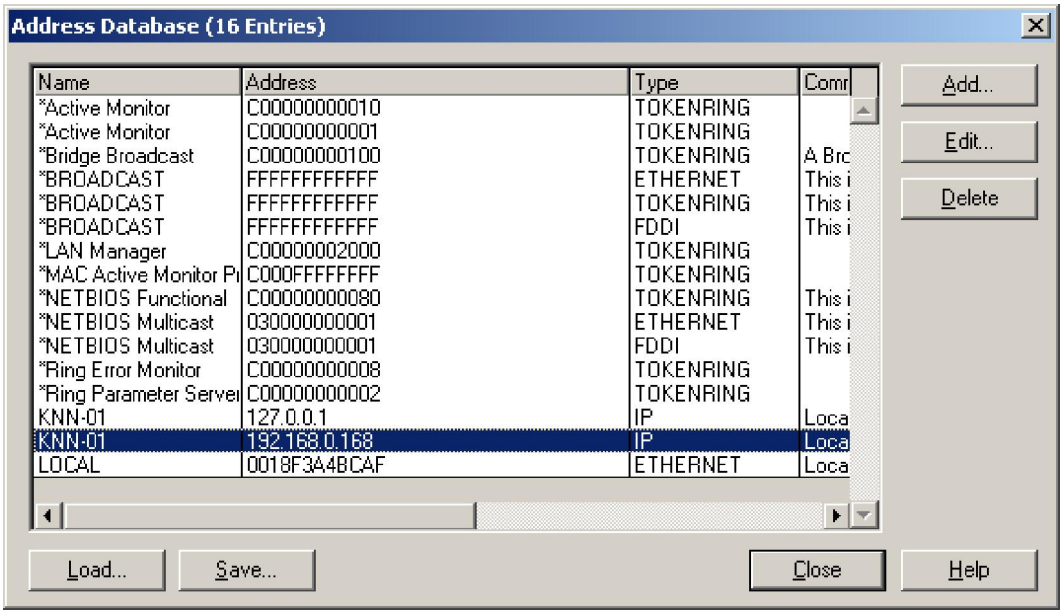


Рис. 2.7. Окно работы с базой данных адресов

Важным этапом настройки сетевого анализатора является создание фильтров записи. Чтобы создать фильтр записи, следует выбрать критерии фильтрации в диалоговом окне фильтра захвата, которое открывается из пункта ***Filter***... в меню ***Capture***.

В этом диалоговом окне (рис. 2.8) отображается дерево критериев, которое является графическим представлением логики фильтра. При каждом добавлении или исключении компонентов спецификации записи эти изменения отражаются в дереве критериев.

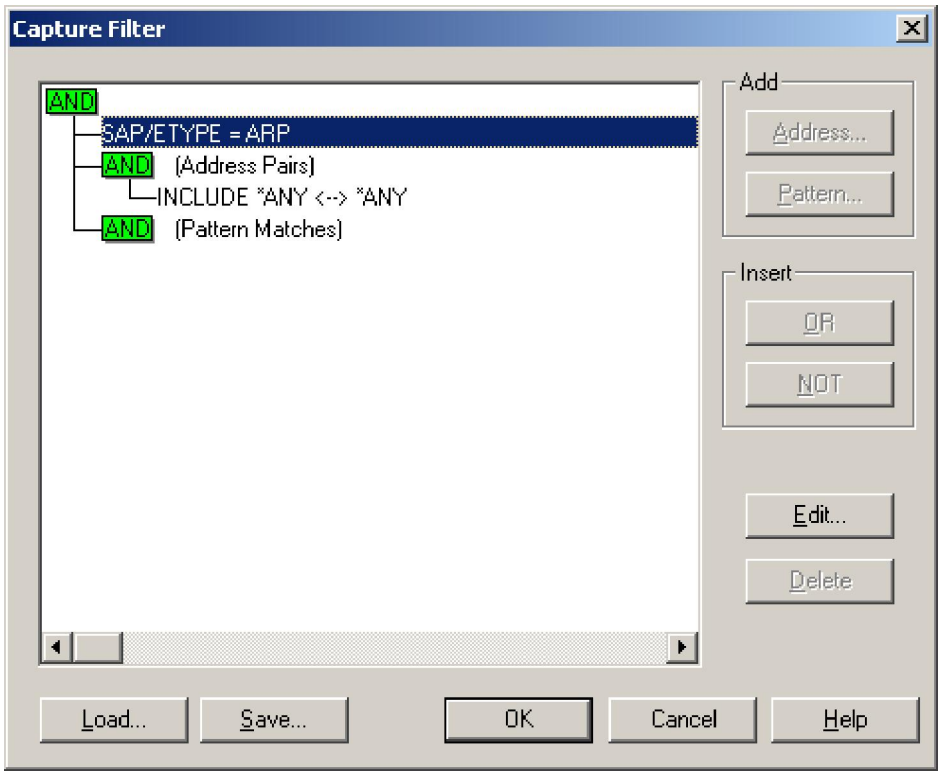


Рис. 2.8. Окно выбора критериев фильтрации при захвате

Допускается использовать три типа критериев фильтрации: на основе протоколов, адресов и на основе соответствия шаблону.

Для фильтрации записываемых пакетов с использованием конкретного протокола необходимо указать этот протокол. Для этого:

- выбрать строку ***SAP/ETYPE***= дерева критериев и нажать кнопку ***Edit***... или дважды кликнуть по этой строке;
- в открывшемся диалоговом окне ***Capture Filter SAPs and ETYPES*** (рис. 2.9) задать нужные протоколы, причем сначала следует запретить все протоколы, а после этого разрешить нужные.

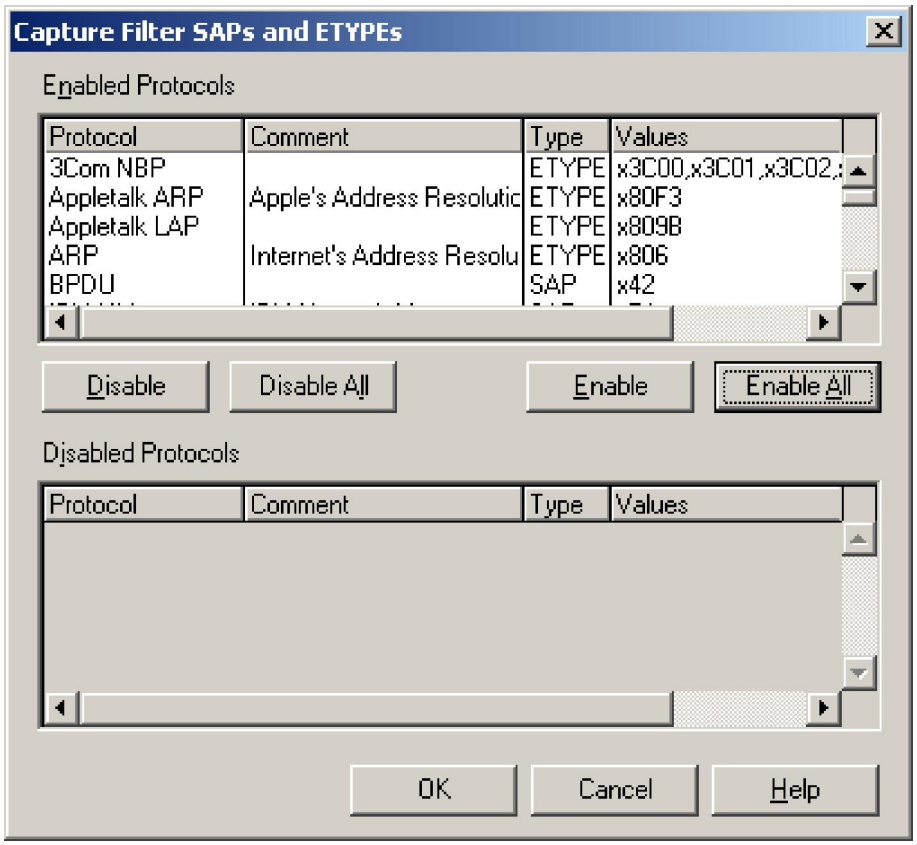


Рис. 2.9. Окно выбора протокола фильтрации при захвате

Для захвата на основе адресов пакетов, относящихся к определенным компьютерам сети, следует указать от одной до четырех адресных пар в фильтре записи (рис. 2.10). Допускается одновременное наблюдение за четырьмя определенными адресными парами, каждая из которых состоит из:

- адресов двух компьютеров, за трафиком между которыми ведется наблюдение;
- стрелок, указывающих интересующее направление передачи данных;
- ключевого слова ***INCLUDE*** или ***EXCLUDE***, которое определяет, будет ли сетевой монитор захватывать или игнорировать соответствующие пакеты данных.

Независимо от того, в каком порядке располагаются адресные пары в диалоговом окне ***Capture Filter***, инструкции ***EXCLUDE*** обрабатываются первыми. Поэтому, если пакет удовлетворяет критериям, указанным в инструкции ***EXCLUDE***, он будет игнорирован независимо от того, присутствует ли в спецификации фильтра инструкции ***INCLUDE*** и ***EXCLUDE***. Сетевой монитор не проверяет, удовлетворяет

ли этот пакет условиям инструкций ***INCLUDE***. Если в списке отсутствуют инструкции ***INCLUDE***, по умолчанию будут записываться пакеты, передаваемые между локальным компьютером и любыми другими узлами сети (т.е. фильтр использует адресную пару ***INCLUDE локальный_компьютер*** <--> ***ANY***).

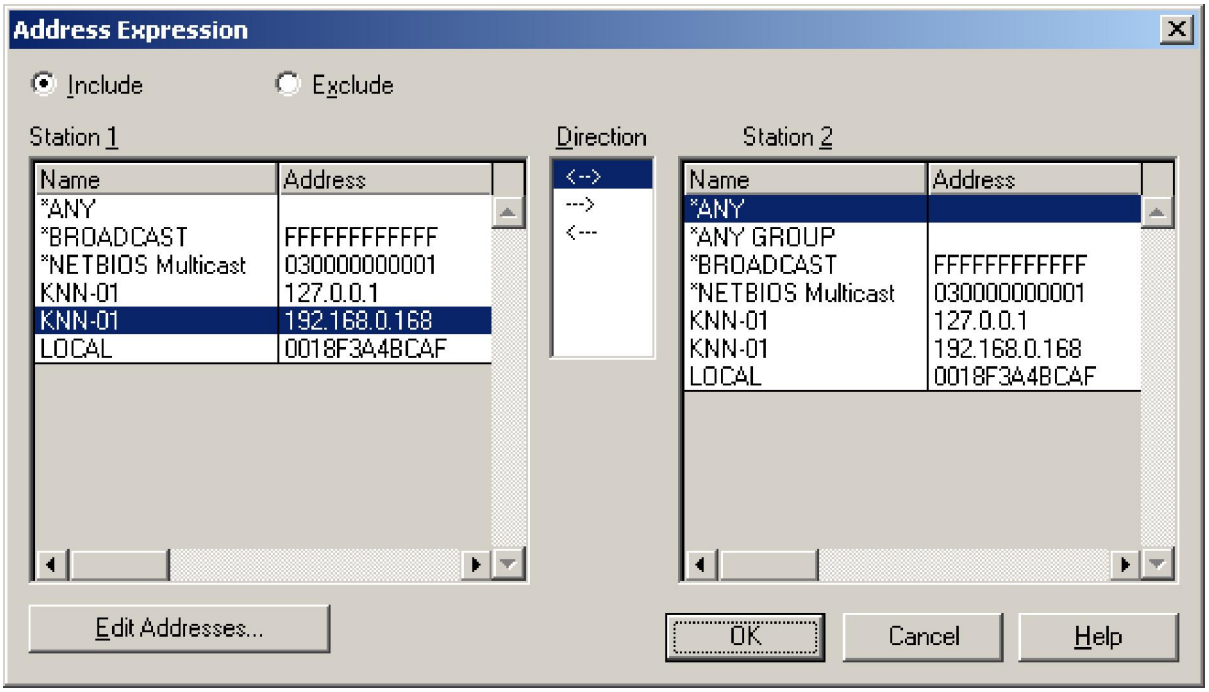


Рис. 2.10. Окно задания адресных пар фильтрации при захвате

Задание адресных пар выполняется в диалоговом окне ***Address Expression***, которое вызывается аналогично окну ***Capture Filter SAPs and ETYPES***.

Фильтрация на основе соответствия шаблону позволяет ограничить запись подмножеством пакетов, которые содержат указанный шаблон – некоторый код, задаваемый в текстовом или шестнадцатеричном виде.

При задании фильтра соответствия шаблону необходимо указать значение и местоположение шаблона в пакете (смещение – число байтов от начала или конца). Если пакеты не имеют постоянной длины (например, пакеты протокола MAC в сетях Ethernet или Token Ring), следует указывать смещение от конца заголовка пакета соответствующей топологии.

Шаблон задается в диалоговом окне ***Pattern Match***, вызываемом аналогично окну ***Address Expression*** (рис. 2.11).

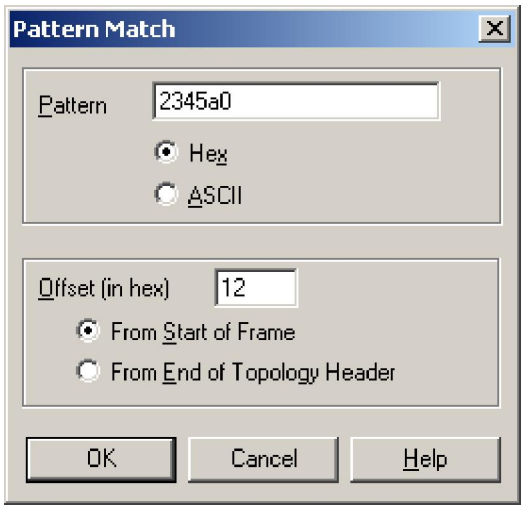


Рис. 2.11. Окно задания шаблона фильтрации при захвате

Единожды созданные критерии могут быть сохранены в отдельном файле и затем многократно использоваться.

### 2.2.2.4. Сбор и расшифровка результатов сбора данных

Запуск сетевого монитора на захват выполняется после настройки его параметров из пункта ***Start*** в меню ***Capture***. Сетевой монитор начинает собирать данные, отправленные или полученные локальным компьютером из сетевого потока, и копировать их во временный файл записи, при этом Сетевой монитор динамически отображает статистику собранных кадров в панелях окна сбора данных.

Для остановки сбора надо выполнить пункт ***Stop and View*** в меню ***Capture***. Захваченные кадры из буфера отображаются в окне записи данных, основные элементы которого показаны на рис. 2.12.

В панели «Сводка» приводятся общие сведения о записанных кадрах в порядке их поступления:

- ***Frame*** – номер кадра, определяемый порядком записи кадров (нумерация кадров начинается с 1);
- ***Time*** – время записи кадра в форме, заданной в параметрах отображения (абсолютное время записи кадра, время записи кадра относительно начала записи в миллисекундах или время записи кадра относительно времени записи предыдущего кадра в миллисекундах);
- ***Src MAC Addr*** – MAC-адрес компьютера, который отправил кадр;

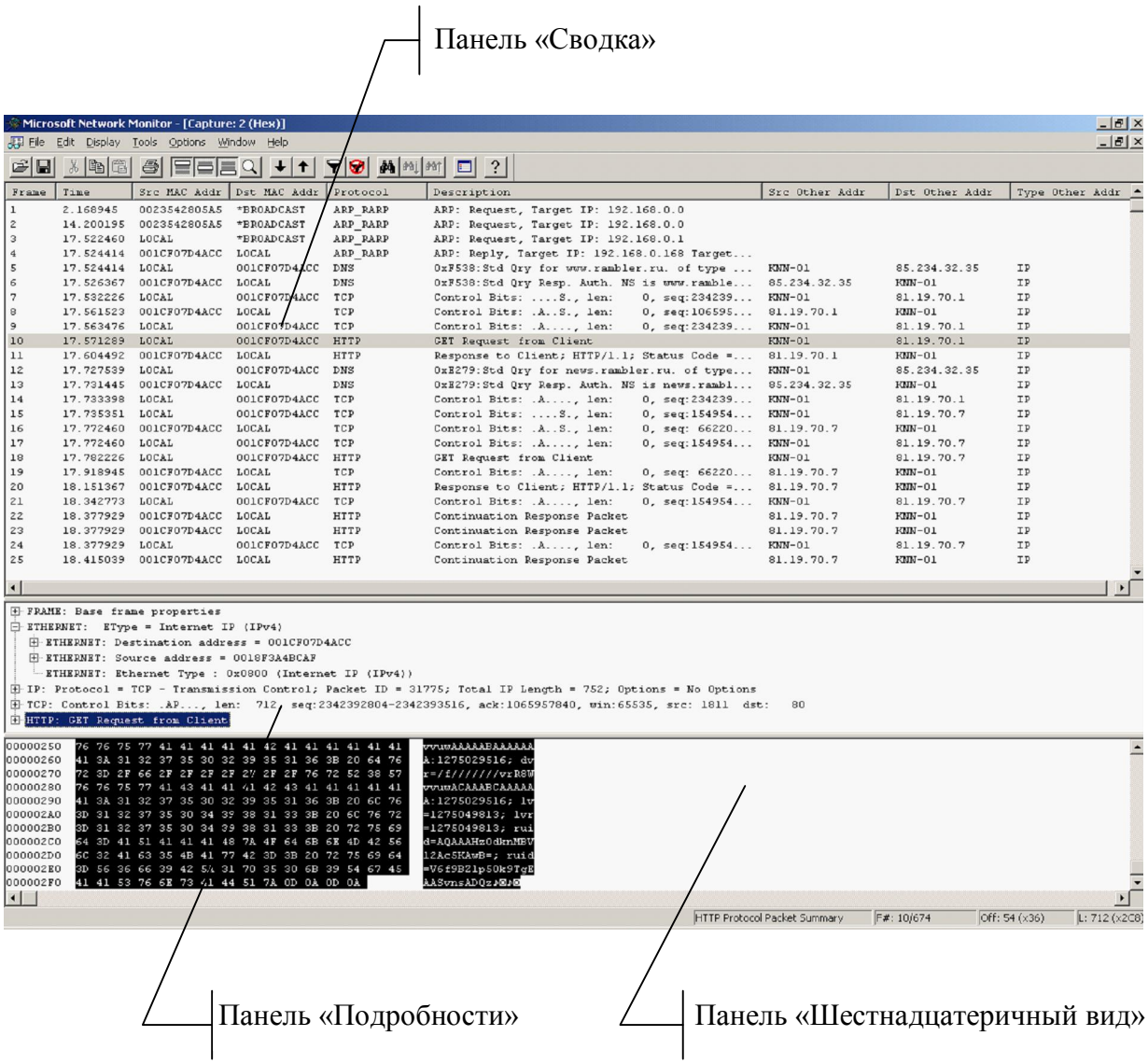


Рис. 2.12. Окно записи данных

• ***Drc MAC Addr*** – MAC-адрес компьютера, на который отправлен кадр;

• ***Protocol*** – основной протокол, используемый для отправки данных;

• ***Dtscription*** – краткое описание содержимого кадра;

• ***Src Other Addr*** – IP- или IPX/XNS-адрес, соответствующий адресу в столбце ***Src MAC Addr***;

• ***Drc Other Addr*** – IP- или IPX/XNS-адрес, соответствующий адресу в столбце ***Drc MAC Addr***;

• ***Type Other Addr*** – типы адресов в столбцах ***Src Other Addr*** и ***Drc Other Addr***.

В панели **Подробности** приводятся сведения о кадре, выделенном в панели **Сводка**, с указанием полей протоколов все уровней,

используемых для его передачи. Чтобы развернуть список полей кадра, щелкните значок «плюс» (+) в панели подробностей. Все остальные строки, использующие данный протокол, автоматически разворачиваются при выделении. Чтобы свернуть список полей кадра, щелкните значок «минус» (–) в панели подробностей.

В панели «Шестнадцатеричный вид» приводятся данные выделенного в панели **Сводка** кадра в шестнадцатеричном и текстовом виде (рис. 2.13).

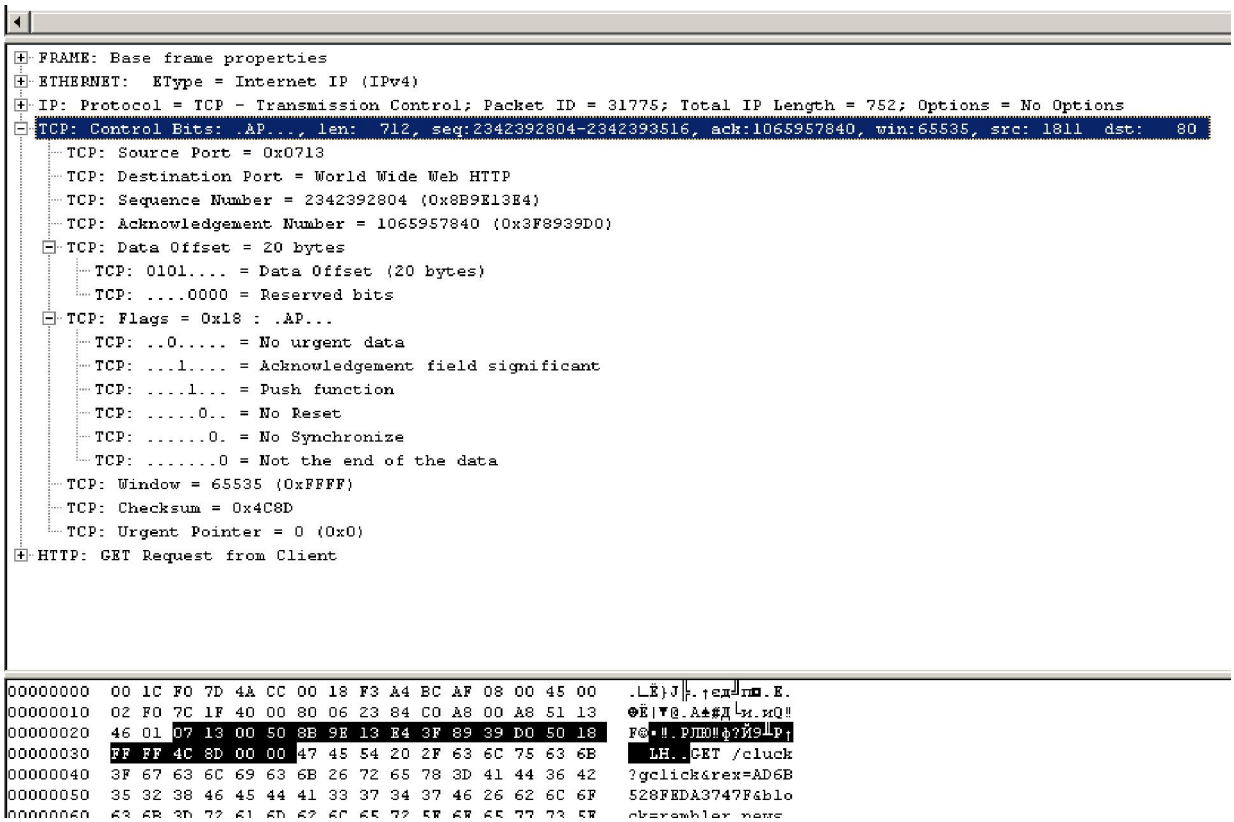


Рис. 2.13. Развернутое отображение полей протокола TCP анализируемого кадра

С целью упрощения анализа записанных данных можно выполнить их фильтрацию на основе выбранных критериев. Для этого надо создать фильтр отображения, который работает как запрос базы данных, Фильтр создается с помощью диалогового окна (рис. 2.14), которое вызывается из меню ***Display*** окна записи данных командой ***Filter***.

В этом диалоговом окне можно указать параметры для ограничения числа отображаемых кадров. Например, можно указывать протоколы, адреса компьютеров и свойства протоколов для кадров, которые надо отобразить в окне записи данных.

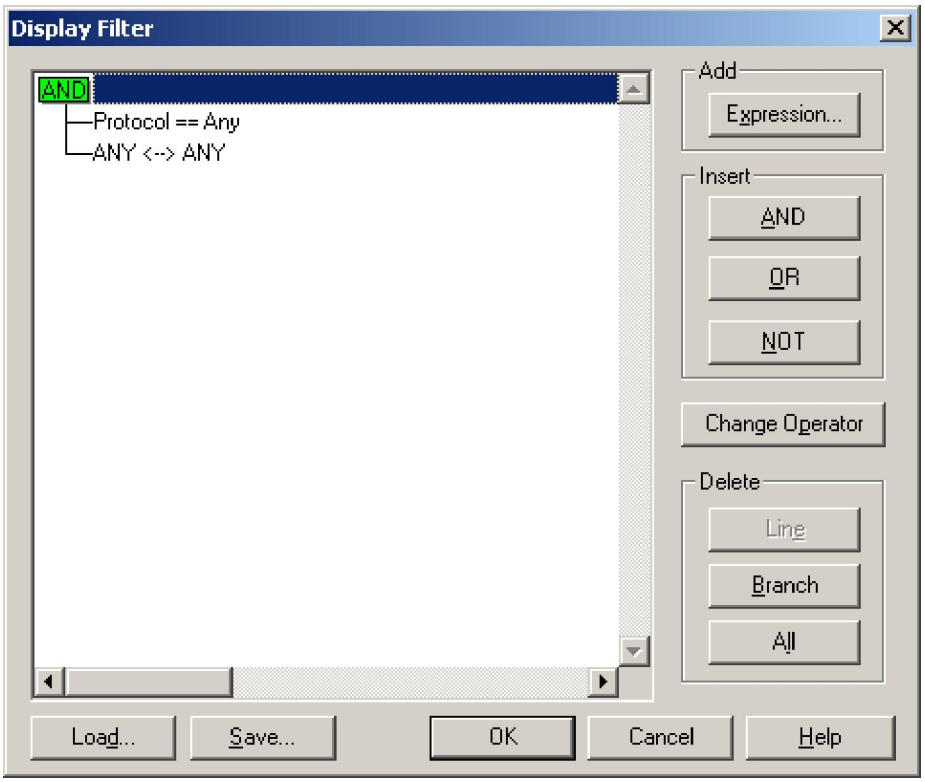


Рис. 2.14. Диалоговое окно Фильтр отображения

Следующие средства управления позволяют внести изменения в дерево критериев фильтра отображения:

• ***Protocol*** = = ... – ветвь дерева критериев, в которой перечислены отображаемые протоколы. По умолчанию в данной ветви установлено значение ***Protocol = = Any***;

• **адрес <–> адрес –** ветвь дерева критериев, в которой перечислены отображаемые пары адресов компьютера. По умолчанию значение, установленное в данной ветви, ***Any <–> Any***.

Чтобы изменить список протоколов либо список пар адресов, надо дважды щелкнуть соответствующую ветвь для открытия диалогового окна Выражение, в котором можно указать отображаемые пары адресов компьютеров, протоколы или свойства протоколов (рис. 2.15), или нажать кнопку ***Expression***.

С помощью кнопок ***AND***, ***OR***, ***NOT*** можно добавить соответствующие ветви в дерево критериев.

С помощью кнопки ***Change Operator*** / ***Change Expression*** можно изменить выбранный в дереве критериев оператор или изменить данные в выбранном выражении.

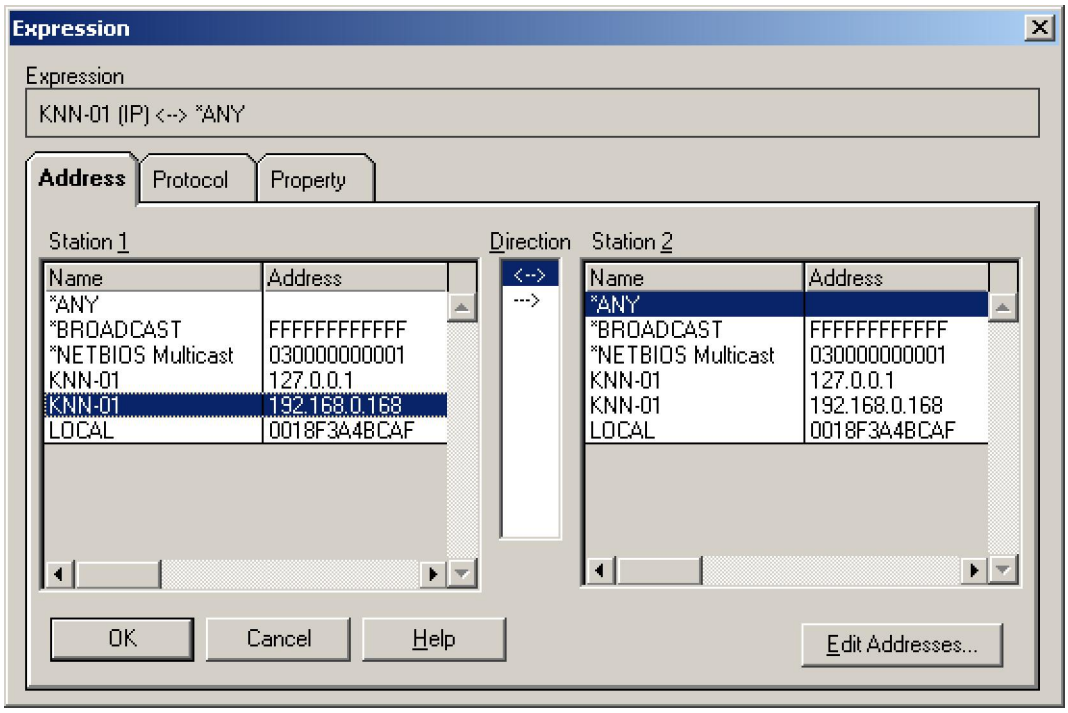


Рис. 2.15 Диалоговое окно для задания адресов, протоколов или свойств протоколов

Кнопки ***Delete*** позволяют удалять все дерево критериев, критерии в дереве либо выбранную строку или выбранную ветвь из дерева критериев.

Кнопка ***Save*** позволяет открыть диалоговое окно сохранения фильтра отображения в котором можно сохранить текущий фильтр в файл с расширением (*.df*). Чтобы позднее открыть фильтр отображения, надо нажать кнопку ***Load***.

Чтобы сохранить собранные кадры в файле, необходимо выполнить команду ***Save as*** в меню ***File*** Окна записи данных, указав расширение .cap в имени сохраняемого файла, что позволит позднее загрузить этот файл в сетевом мониторе с помощью пункта ***Open***.

## 2.3. Задание на лабораторную работу

2.3.1. Настроить Сетевой монитор, установив фильтры по протоколам «все», по адресам – MAC-адрес своего компьютера и любой сторонний. Запустить захват, контролировать трафик сети с помощью Окна сбора данных не менее 2–3 мин, обратившись с браузера на любой веб-сайт. Прекратить захват и просмотреть захваченные кадры с помощью Окна записи данных, определив их тип. Разобрать 1–2 кадра по полям. Найти и разобрать кадры широковещательной рассылки.

В отчете привести собранную статистику мониторинга сети, назначение, формат и содержание полей разобранных кадров.

2.3.2. С помощью программы *Network Monitor* проанализировать работу протоколов ARP, IP и ICMP. Для этого:

- настроить Сетевой монитор, установив фильтры по протоколам ARP и IP, по адресам – MAC-адрес своего компьютера и любой сторонний;
- с помощью утилиты **arp** просмотреть ARP-кэш;
- запустить захват сетевого монитора;
- пропинговать ЭВМ, IP-адрес которой отсутствует в ARP-кэш;
- остановить сбор данных и отобразить захваченные кадры;
- отфильтровать ARP-кадры и выполнить их разбор;
- отфильтровать ICMP-пакеты и выполнить их разбор.

В отчете привести содержание кадров с ARP-запросами и ответами, содержание пакетов с ping-откликом и ping-запросом.

2.3.3. С помощью программы *Network Monitor* проанализировать работу протоколов IP и TCP. Для этого:

- настроить Сетевой монитор, установив фильтры по протоколам ARP и IP, по адресам – MAC-адрес своего компьютера и любой сторонний;
- запустить захват сетевого монитора;
- запустить браузер, загрузить с его помощью страницу с веб-сайта по указанию преподавателя, закрыть браузер;
- остановить сбор данных и отобразить захваченные кадры;
- проанализировать захваченные пакеты и найти и разобрать последовательности пакетов при установлении и закрытии TCP-соединения;
- проанализировать захваченные последовательности пакетов при передаче данных с открытой страницы веб-сайта, определить работу механизмов обеспечения надежности передачи данных (нумерация байтов, квитирование, управление скользящим окном).

В отчете привести содержание пакетов при установлении и разрыве TCP-соединения, изменение значений полей «Номер последовательности» и «Номер последовательности подтверждения», а также размеров окна при передаче данных.

## 2.4. Вопросы для самопроверки

1. Какую роль выполняет транспортный протокол TCP в сети Internet?

2. Что такое скользящее окно?

3. Что такое сокет?

4. В чем отличие протокола передачи сообщений UDP от протокола TCP?

5. Что протокол ARP позволяет определить передающему узлу?

6. В чем отличие форматов ARP-запроса и ARP-ответа?

7. Что такое фрагментация? Какие средства протокола IP управляют фрагментацией?

8. Каково назначение полей «Номер последовательности» и «Номер последовательности подтверждения» заголовка TCP?

9. Для чего предназначен протокол ICMP? Перечислите основные типы сообщений протокола ICMP.

10. Каково назначение сетевого монитора?

11. Какая информация отображается в Окне сбора данных и Окне записи данных сетевого монитора?

12. Какие способы фильтрации пакетов при захвате реализованы в сетевом мониторе?

13. Какие параметры протокола TCP позволяют однозначно идентифицировать каждое соединение?

# Лабораторная работа № 3

# Анализ протоколов прикладного уровня

## 3.1. Цель работы

**Целью работы** является изучение прикладных протоколов стека TCP/IP и приобретение практических навыков анализа протоколов с помощью программы *Network Monitor*.

## 3.2. Теоретический материал

### 3.2.1. Описание протокола HTTP

#### 3.2.1.1. Назначение и принцип работы протокола HTTP

Протокол прикладного уровня для передачи гипертекста HTTP (*HyperText Transfer Protocol*) был разработан как основа Всемирной паутины WWW (*World Wide Web*) – распределенной системы, предоставляющей доступ к связанным между собой документам, расположенным на веб-серверах Интернета. Большинство ресурсов Всемирной паутины представляет собой гипертекст. Гипертекстовые документы, размещаемые во Всемирной паутине, называются веб-страницами. Несколько web-страниц, объединенных общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом. Для загрузки и просмотра веб-страниц используются специальные программы – браузеры.

Основой HTTP является технология «клиент–сервер», т.е. программа-клиент инициирует соединение и выдает ему HTTP-запрос. Сервер обрабатывает этот запрос, производит необходимые действия и возвращает клиенту HTTP-ответ с результатом.

Протокол HTTP никак не связан со способом визуализации веб-страниц и определяет только метод обмена информацией между клиентом и сервером. Служба WWW, помимо протокола HTTP, базируется еще на трех стандартах:

- универсальном способе адресации ресурсов в сети URL (*Universal Resource Locator*);
- языке гипертекстовой разметки документов HTML (*HyperText Markup Language*);

• универсальном интерфейсе шлюзов CGI (*Common Gateway Interface*), обеспечивающем взаимодействия HTTP-сервера с другими программами (например, СУБД).

Версии протокола HTTP используют TCP в качестве протокола транспортного уровня. HTTP-клиент сначала устанавливает TCP-соединение с сервером, после чего клиент и сервер начинают взаимодействовать с протоколом TCP через интерфейс сокетов (стандартный номер порта–80). После завершения обслуживания клиентов сервер не сохраняет о них никакой информации, т.е. протокол HTTP является протоколом без запоминания состояния соединения.

Протокол HTTP поддерживает постоянные и непостоянные соединения. Непостоянное соединение состоит из единственного сообщения-запроса и сообщения-ответа. Для получения веб-страницы требуется многократное установление и завершение соединения. При этом необходимо учитывать, что каждое соединение требует от протокола TCP выделения буфера, а также ряда служебных переменных как на стороне клиента, так и на стороне сервера. Так как многие веб-серверы параллельно обслуживают сотни клиентов, подобная схема серьезно затрудняет процесс взаимодействия между клиентами и сервером. Кроме того, установление соединения для каждого объекта из-за времени оборота приводит к дополнительным временным затратам.

При постоянном соединении сервер не закрывает TCP-соединение после обслуживания запроса, что позволяет обслужить несколько запросов в одном соединении. Передача веб-страниц через одно соединение возможна в случаях, если все объекты находятся на одном и том же сервере. Обычно закрытие TCP-соединения происходит в случае, когда оно не используется в течение некоторого установленного времени (интервала ожидания).

По умолчанию протокол HTTP 1.1 настроен на использование постоянных соединений с конвейеризацией. Это позволяет новому запросу направляться к серверу, не дожидаясь окончания обслуживания других запросов. Аналогично сервер, получая новые запросы, начинает их немедленное обслуживание. Таким образом, уменьшается время установления соединения и значительно сокращается время простоя сервера.

### 3.2.1.2. Структура HTTP-запроса

В HTTP существуют два типа сообщений: запросы и ответы, которые представляет собой совокупность текстовых символов в кодировке ASCII.

HTTP-запрос состоит из заголовка запроса и тела запроса, разделенных пустой строкой. Тело запроса может отсутствовать.

Заголовок запроса состоит из главной (первой) строки запроса и последующих строк, уточняющих запрос в главной строке. Последующие строки могут отсутствовать.

Запрос в главной строке состоит из трех частей, разделенных пробелами:

**Метод Унифицированный идентификатор ресурса HTTP/Версия**.

Метод (иначе говоря, команда HTTP) может иметь следующие значения:

- **GET** – запрос документа, употребляется наиболее часто (в HTTP/0.9 был единственным);
- **HEAD** – запрос заголовка документа, отличается от **GET** тем, что выдается только заголовок запроса с информацией о документе, а сам документ не выдается;
- **POST** – применяется для передачи данных CGI-скриптам, а сами данные – в последующих строках запроса в виде параметров;
- **PUT** – запрос на размещение документа на сервере, используется редко.

Унифицированный идентификатор ресурса **URI** (*Uniform Resource Identifier*) представляет собой путь к запрашиваемому документу (файлу) на сервере. Если запрашивается корневой файл из корневой директории веб-сервера, то **URI** может отсутствовать.

Если ресурс – просто какой-либо файл для считывания, сервер должен по этому запросу выдать его в теле ответа. Если же это путь к какому-либо CGI-скрипту, то сервер запускает скрипт и возвращает результат его выполнения. Благодаря такой унификации ресурсов для клиента практически безразлично, как ресурс представлен на сервере.

HTTP/Версия указывает на версию протокола HTTP, с которой работает клиентская программа (наиболее распространена версия HTTP 1.1).

Строки после главной строки запроса задают параметры запроса и имеют формат **Параметр**: **значение**. Наличие параметров необяза-

тельно, все строки после главной строки запроса могут отсутствовать; в этом случае сервер принимает их значение по умолчанию или по результатам предыдущего запроса. Некоторые наиболее употребительные параметры HTTP-запроса:

- ***Connection*** – соединение. Может принимать значения ***Keep-Alive*** и ***close***. ***Keep-Alive*** («оставить в живых») означает, что после выдачи данного документа соединение с сервером не разрывается, и можно выдавать еще запросы. Большинство современных браузеров работают именно в режиме ***Keep-Alive***, так как он позволяет за одно соединение с сервером получить как html-документ, так и рисунки веб-страницы. Единожды установленный режим ***Keep-Alive*** сохраняется до первой ошибки или до явного указания в очередном запросе ***Connection***: ***close***, требующего закрытия соединения после ответа на данный запрос;
- ***User-Agent*** – список названий и версий браузера, например, ***User-Agent***: ***Mozilla/4.0*** (***compatible***; ***MSIE 7.0***; ***Windows NT 5.1***; ***MRA 5.0*** (***build 02094***);
- ***Accept*** – список поддерживаемых браузером типов содержимого в порядке их предпочтения данным браузером, например, ***Accept image/gif***, ***image/x-xbitmap***, ***image/jpeg***, ***image/pjpeg***, ***application/vnd.ms***: ***excel***, ***application/msword***, ***application/vnd.ms-powerpoint***,****/****. Это параметр нужен для случая, когда сервер может выдавать один и тот же документ в разных форматах и используется в основном CGI-скриптами для формирования ответа, адаптированного для данного браузера;
- ***Referer*** – ссылка, т.е. URL, с которого перешли на этот ресурс;
- ***Host*** – имя хоста, с которого запрашивается ресурс;
- ***Accept-Language*** – поддерживаемый язык. Имеет значение для сервера, который может выдавать один и тот же документ в разных языковых версиях;
- ***Accept-Encoding*** – перечень поддерживаемых способов кодирования запрашиваемого документа при передаче. Например, большинство современных браузеров включают указание на возможность сжатия контента в каждый отсылаемый запрос: ***Accept-encoding***: ***gzip***, ***deflate***.

Пример HTTP-запроса при обращении к главной странице сайта *alice.pnzgu.ru*:

***GET / HTTP/1.1***

***Accept***: ***image/gif***, ***image/x-xbitmap***, ***image/jpeg***, ***image/pjpeg***, ***application/x-shockwave-flash***, ***application/vnd.ms-excel***, ***application/vnd.ms-powerpoint***, ***application/msword***, ***application/x-ms-application***, ***application/x-ms-xbap***, ***application/vnd.ms-xpsdocument***, ***application/xaml+xml***, ****/****

***Accept-Language***: ***ru***

***UA-CPU***: ***x86***

***Accept-Encoding***: ***gzip***, ***deflate***

***User-Agent***: ***Mozilla/4.0*** (***compatible***; ***MSIE 7.0***; ***Windows NT 5.1***; ***MRA 5.0*** (***build 02094***); ***Mozilla/4.0*** (***compatible***; ***MSIE 6.0***; ***Windows NT 5.1***; ***SV1***); ***.NET CLR 2.0.50727***; ***.NET CLR 3.0.4506.2152***; ***.NET CLR 3.5.30729***)

***Host***: ***alice.pnzgu.ru***

***Connection***: ***Keep-Alive***

### 3.2.1.3. Структура HTTP-запроса

Формат ответа очень похож на формат запроса: он также имеет заголовок и тело, разделенное пустой строкой.

Заголовок также состоит из основной строки и строк параметров, но формат основной строки отличается от таковой в заголовке запроса.

Основная строка запроса состоит из трех полей, разделенных пробелами: **HTTP/Версия Код ошибки Описание ошибки**:

- версия протокола – аналогичен соответствующему параметру запроса;
- код ошибки – кодовое обозначение «успешности» выполнения запроса;
- словесное описание ошибки – расшифровка предыдущего кода. Наиболее часто встречающиеся: ***200 OK*** – документ отправлен, ***404 Not Found*** – документ не найден (ошибка URL), клиент должен проверить правильность написания URL, если не помогает, значит, документ уже удален, ***500 Internal Server Error*** – внутренняя ошибка сервера.

Наиболее употребительные параметры HTTP-ответа:

- ***Connection*** – аналогичен соответствующему параметру запроса. Если сервер не поддерживает ***Keep-Alive***, то значение ***Connection*** в ответе всегда ***close***;

- ***Content-Type*** – тип содержимого ответа. В зависимости от значения ***Content-Type*** браузер воспринимает ответ как HTML-страницу (***Content-Type***: ***text/html***), картинку (***Content-Type***: ***image/jpeg***), простой текст (***Content-Type***: ***text/plain***). Значение ***Content-Type*** для браузера аналогично значению расширения файла для Windows;
- ***Content-Length –*** длина содержимого ответа в байтах;
- ***Last-Modified*** – дата последнего изменения документа;
- ***Date*** – дата и время генерации ответа;
- ***Server –*** список названий и версий веб-сервера и его компонентов. ***Server***: ***Apache/2.0.53*** (***Linux/SUSE***).

Пример HTTP-ответа, полученного при обращении к главной странице веб-сайта ***alice.pnzgu.ru***:

***HTTP/1.1 200 OK***

***Date***: ***Thu***, ***09 Sep 2010 07***:***01***:***05 GMT***

***Server***: ***Apache/2.0.53*** (***Linux/SUSE***)

***Accept-Ranges***: ***bytes***

***Content-Length***: ***116***

***Keep-Alive***: ***timeout=15***, ***max=100***

***Connection***: ***Keep-Alive***

***Content-Type***: ***text/html***

***<html> <head> <title> </title> <meta http-equiv="refresh" content="0***; ***url=/cms2"Content-Type> </head> <body> </body> </html>***

Последние две строки представляют тело ответа, которое передается после заголовка.

## 3.2.2. Описание протокола FTP

### 3.2.2.1. Назначение и принцип работы протокола FTP

Протокол прикладного уровня передачи файлов FTP (*File Transfer Protocol*) используется службой передачи файлов в Интернете. Протокол FTP позволяет подключаться к серверам FTP, просматривать содержимое их каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами.

Протокол FTP для передачи данных использует транспортный протокол TCP, причем, в отличие от большинства других протоколов, FTP использует сразу два TCP-соединения: одно для управления, а

другое для собственно передачи данных. Порт 21 используется для передачи команд, а порт 20 для передачи данных (порт для канала данных может назначаться сервером и из нестандартных портов с номерами > 1024).

Служба FTP построена по хорошо известной схеме клиент сервер. FTP-клиент посылает запросы серверу и принимает файлы. FTP-сервер обрабатывает запросы клиента на получение файла. Схема взаимодействия клиента и сервера показана на рис. 3.1.

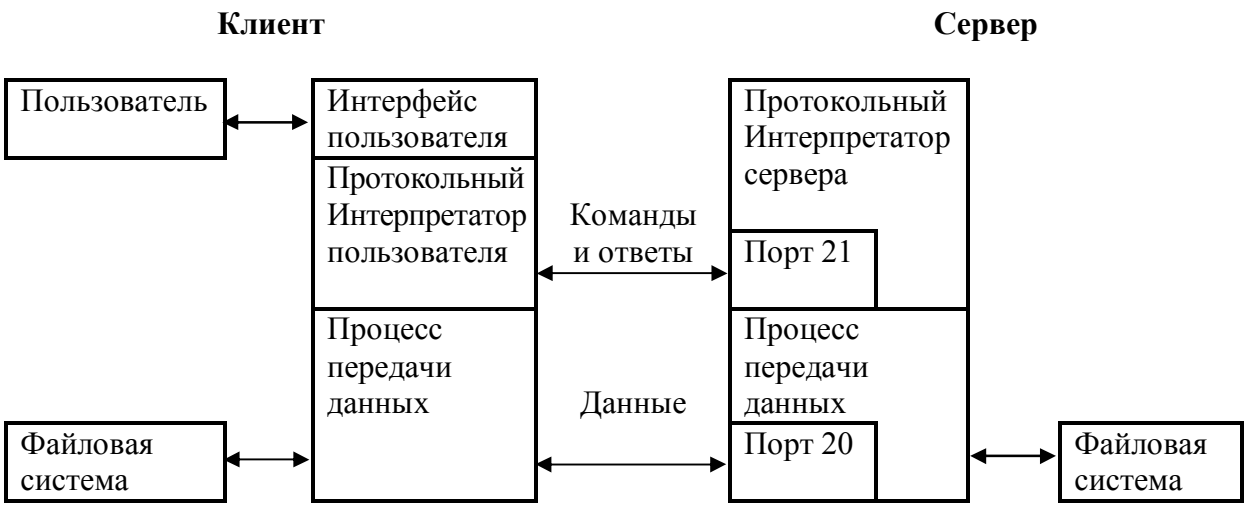


Рис. 3.1. Схема взаимодействия клиента и сервера FTP

Возможны два режима установления соединений в протоколе FTP – активный и пассивный.

Действия сервера и клиента в активном режиме:

- клиент инициирует TCP-соединение с динамического порта N (N > 1024) к порту номер 21 на FTP-сервере;
- сервер посылает ответ на порт N клиента;
- сервер устанавливает связь для передачи данных по порту 20 на порт клиента N+1.

Действия сервера и клиента в пассивном режиме:

- клиент устанавливает связь и посылает запрос, сообщая, что надо работать в пассивном режиме, на 21-й порт сервера с динамического порта N (N > 1024);
- сервер посылает ответ и сообщает номер порта для канала данных P (P > 1024) на порт N клиента;

- клиент устанавливает связь для передачи данных по порту N+1 на порт сервера P.

Главное отличие между активным и пассивным режимами FTP – это та сторона, которая открывает соединение для передачи данных. В активном режиме клиент должен принять соединение от FTP-сервера. В пассивном режиме клиент всегда инициирует соединение.

Хотя активный FTP и удобен для сервера, но попытка соединения со стороны сервера с высокими (по номеру) портами на клиенте будет блокирована брандмауэром на стороне клиента.

Пассивный режим предназначен для соединения через брандмауэр на стороне клиента, инициатором установления всех соединений является FTP-клиент.

### 3.2.2.2. Управление обменом файлов в протоколе FTP

Работа FTP на пользовательском уровне содержит несколько этапов:

- идентификация (ввод имени и пароля);
- выбор каталога;
- определение режима обмена (поблочный, поточный, ASCII или двоичный);
- выполнение команд обмена;
- завершение процедуры.

Протокол FTP определяет запрос-ответный способ взаимодействия между клиентом и сервером, который реализуется обменом командами и ответами.

Команды передаются серверу в текстовом виде. Команда состоит из четырехбуквенного имени, за которым может следовать аргумент (там, где он требуется). Аргумент отделяется от команды пробелом. Например, команда авторизации, передающая на сервер имя учетной записи «anonymous», выглядит так: ***user anonimus***.

Основные команды FTP:

- ***ABOR*** – прервать предыдущую команду FTP и любую передачу данных;
- ***CWD имя директории*** – имя новой рабочей директории;
- ***CDUP*** – перейти на один уровень директории вверх;
- ***LIST список файлов*** – список файлов или директорий;

- ***MODE режим передачи*** – режим передачи данных (S – потоковый, B – блочный, C – сжатый);
- ***PASS пароль*** – пароль на сервере;
- ***PORT n1***, ***n2***, ***n3***, ***n4***, ***n5***, ***n6*** – IP-адрес клиента (n1.n2.n3.n4) и порт (n5*256 + n6);
- ***QUIT*** – закрыть бюджет на сервере;
- ***RETR имя файла*** – получить файл;
- ***STOR имя файла*** – положить файл;
- ***SYST сервер*** – возвращает тип системы;
- ***TYPE тип*** – указать тип файла (A для ASCII, I для двоичного);
- ***USER имя пользователя*** – имя пользователя на сервере.

Ответы сервера представляют собой код результата выполнения команды, состоящий из трех цифр в формате ASCII, за которым следует текст, отделенный от кода пробелом. Код предназначен для анализа FTP-клиентом. По нему можно однозначно определить статус выполнения команды. Текст является комментарием к коду, предназначенным для пользователя. Пример ответа сервера на команду ***user anonimus***: ***331 Guest login ok***, ***send your complete e-mail address as password***.

Каждая из трех цифр в коде отклика имеет собственный смысл. Так, значения первых и вторых цифр в коде отклика означают:

- 1yz – положительный предварительный отклик, т.е. действие началось, однако необходимо дождаться еще одного отклика перед отправкой следующей команды;
- 2yz – положительный отклик о завершении, может быть отправлена новая команда;
- 3yz – положительный промежуточный отклик, когда команда принята, однако необходимо отправить еще одну команду;
- 4yz – временный отрицательный отклик о завершении, требуемое действие не произошло, однако ошибка временная, поэтому команду необходимо повторить позже;
- 5yz – постоянный отрицательный отклик о завершении, когда команда не была воспринята и повторять ее не стоит;
- x0z – синтаксическая ошибка;
- x1z – информация;
- x2z – соединения. Отклики имеют отношение либо к управляющему, либо к соединению данных;

- x3z – отклик имеет отношение к аутентификации или командам, связанным с бюджетом;
- x4z – не определено;
- x5z – состояние файловой системы.

Третья цифра дает дополнительное объяснение сообщению об ошибке. Ниже приведены некоторые типичные отклики с возможными объясняющими строками:

- 125 Соединение данных уже открыто; начало передачи;
- 200 Команда исполнена;
- 214 Сообщение о помощи (для пользователя);
- 331 Имя пользователя принято, требуется пароль;
- 425 Невозможно открыть соединение данных;
- 452 Ошибка записи файла;
- 500 Синтаксическая ошибка (неизвестная команда);
- 501 Синтаксическая ошибка (неверные аргументы).

Перед передачей файла данных по протоколу FTP необходимо определить тип данных. Основные типы данных: ***ASCII*** (***TYPE A***) – передача текстовой информации и ***IMAGE*** (***TYPE I***) – передача бинарных файлов.

Стандарт протокола FTP обеспечивает три режима передачи данных по сети:

- потоковый, когда данные передаются как поток байтов;
- блочный, когда файл передается как последовательность блоков данных; каждый блок данных включает заголовок, содержащий описатель и длину блока;
- сжатый, при котором перед передачей происходит уплотнение информации с целью уменьшения объемов передаваемых данных.

### 3.2.2.3. Использование программы Total Commander как FTP-клиента

Популярный файловый менеджер *Total Commander* имеет встроенного FTP-клиента, которого можно использовать для исследования протокола FTP.

Ядро работы с FTP-протоколом составляет окно соединения с FTP-сервером, которое вызывается либо по нажатию на иконке **FTP** панели инструментов, либо из пункта **Соединится с FTP-сервером** … меню **Сеть**, либо через ***Ctrl+F 99*** (рис. 3.2).

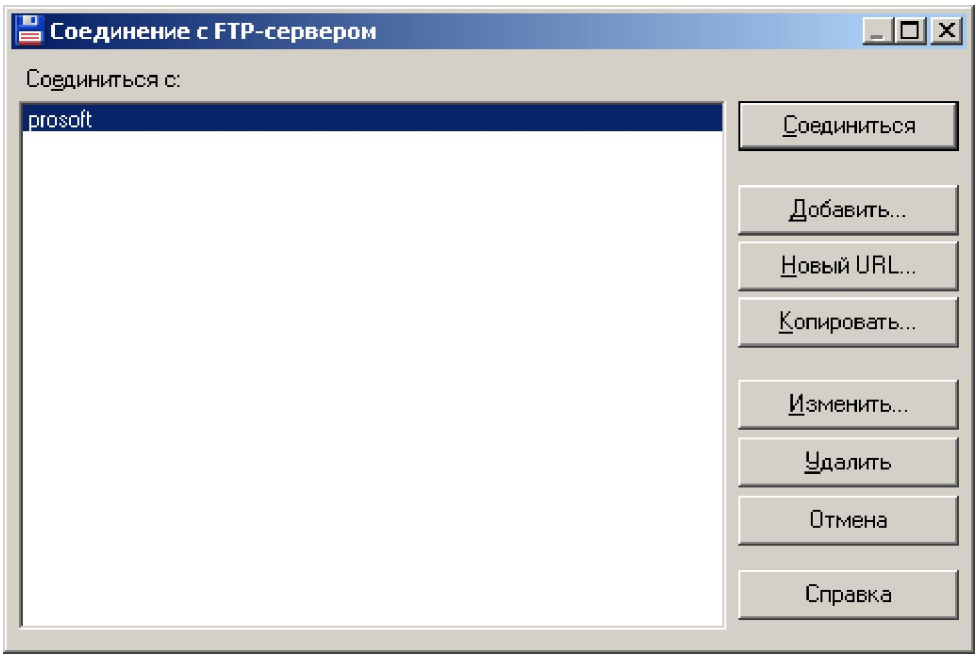


Рис. 3.2. Окно **Соединение с FTP-сервером**

Рассмотрим настройку FTP-клиента и выполнение работ с FTP-сервером на примере обращений к серверу ***ftp://ftp.prosoft.ru /pub/Hardware/***.

Для создания учетной записи нажать кнопку **Добавить** и заполнить формы окна (рис. 3.3):

- **Заголовок** – имя новой учетной записи ***prosoft***;
- **Имя сервера** – в данном случае берется из URL ***ftp.prosoft.ru***;
- **Учетная запись** – имя FTP-пользователя, в данном случае ***anonymous***;
- для анонимного соединения нажать соответствующую кнопку и в качестве пароля ввести адрес электронной почты (рис. 3.4);
- **Удаленная папка** – вводится путь к искомой папке на сервере, в данном случае ***pub/Hardware*** (если форму не заполнять, то откроется корневой каталог FTP-сервера);
- **Тип сервера** – установить автоопределение;
- обязательно установить **Пассивный режим обмена**;
- если клиент работает через прокси-сервер или брандмауэр, надо указать его IP-адрес и номер порта;
- заполнение остальных окон необязательно.

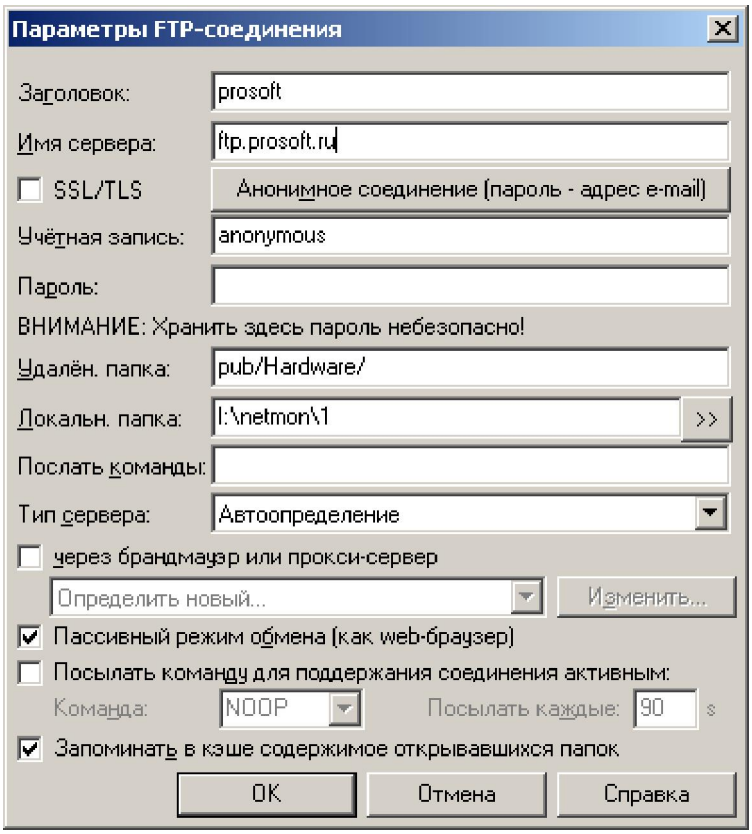


Рис. 3.3. Окно **Параметры FTP-соединения**

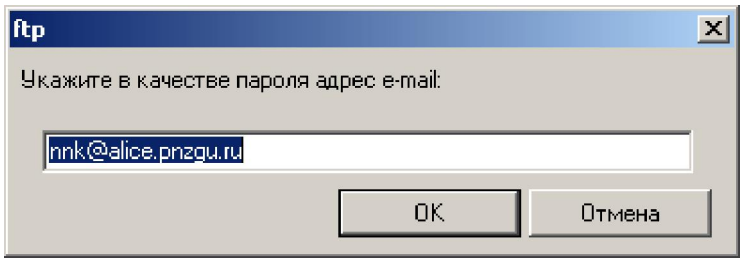


Рис. 3.4. Окно ввода пароля анонимного подключения

Для активации учетной записи нажать кнопку **ОК**.

Для обращения к серверу по созданному соединению выделить имя соединения в окне **Соединение с FTP-сервером** и нажать кнопку **Соединение**.

При успешном соединении в главном окне появится строка **FTP** с окном **Режим обмена**, где отображаются установленные параметры соединения, кнопка **Отключение** и окно протокола, двойной щелчок по которому позволяет просмотреть протокол обмена управляющей информацией при работе установленного FTP-соединения в редакторе *Lister*, встроенном в программу *Total Commander* (рис. 3.5).

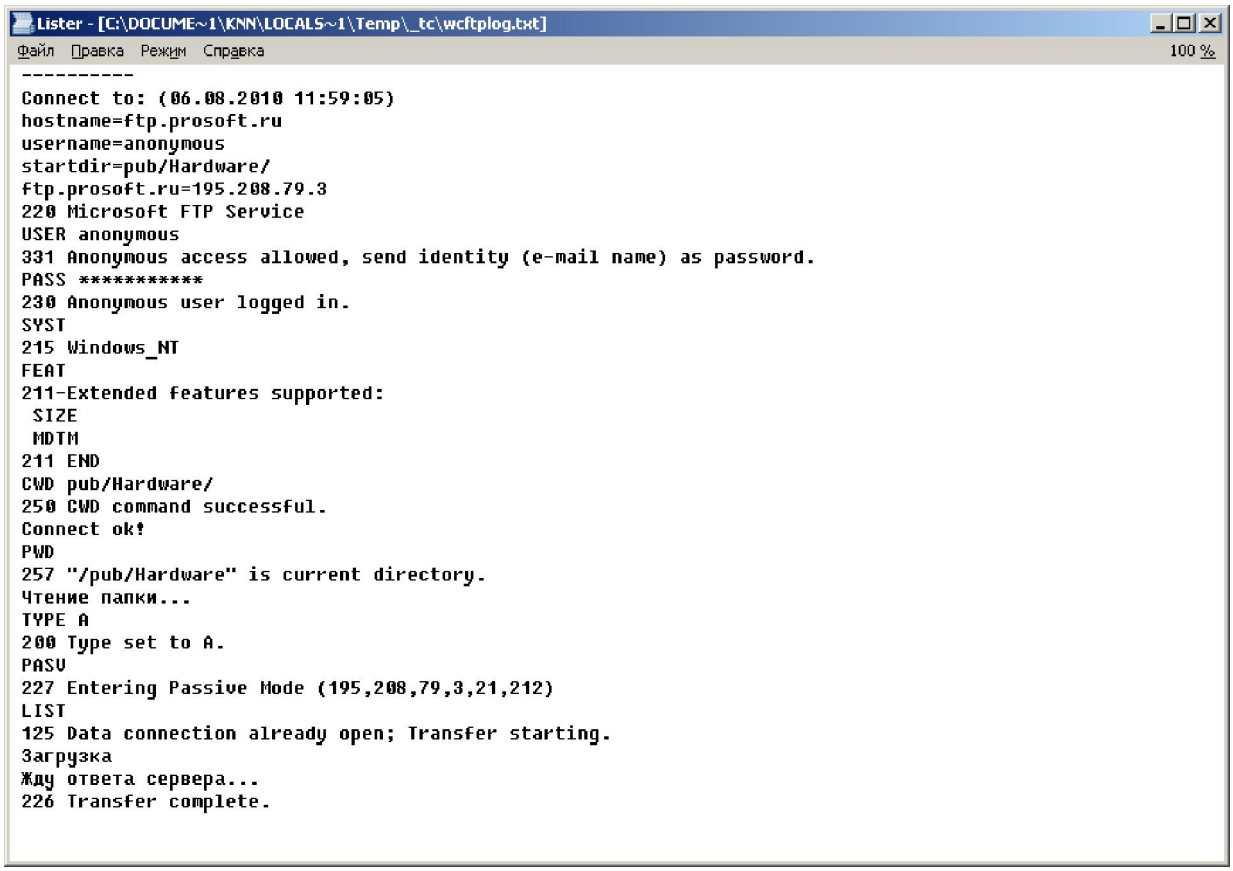


Рис. 3.5. Протокол обмена командами и ответами

В левой и в правой панелях *Total Commander* будут открыты заданные папки на клиентской машине и FTP-сервере.

Для закрытия FTP-соединения нажать кнопку **Отключение**.

## 3.3. Задание на лабораторную работу

3.3.1. С помощью программы *Network Monitor* выполнить мониторинг обмена информацией при обращении к веб-сайту, для чего:

- настроить Сетевой монитор, установив фильтры по протоколам IP, по адресам – IP-адрес своего компьютера и любой сторонний;
- запустить браузер, загрузить с его помощью главную страницу с веб-сайта по указанию преподавателя, сохранить ее в файле, последовательно перейти на две внутренние страницы, закрыть браузер;
- остановить сбор данных и отобразить захваченные кадры;
- сохранить собранные кадры в файле.

3.3.2. С помощью программы *Network Monitor* проанализировать работу протокола HTTP при обращении к веб-сайту. Для этого:

• настроить Сетевой монитор, выполнив фильтрацию записанных данных по протоколу HTTP;

• проанализировать заголовки запросов и ответов, определить режимы работы браузера и сервера;

• открыть с помощью браузера сохраненную страницу и по ее HTML-коду определить количество и назначение запросов к серверу, выполняемых при ее открытии, проверить по содержанию перехваченные Сетевым анализатором пакеты.

3.3.3. В отчете привести:

• содержание заголовков пакетов при обращении к веб-сайту, дать объяснение всем параметрам;

• текст с HTML-кодом открытой страницы, указать, какие теги вызвали запрос к веб-серверу, перечислить назначение и наименование файлов, загруженных при открытии веб-страницы.

3.3.4. С помощью программ *Network Monitor* и *Total Commander* выполнить мониторинг обмена информацией при обращении к FTP-серверу, для чего:

• в программе *Total Commander* создать и настроить FTP-соединение, адрес сервера и параметры соединения получить у преподавателя;

• настроить Сетевой монитор, установив фильтры по протоколам IP, по адресам – IP-адрес своего компьютера и любой сторонний;

• запустить захват данных в программе *Network Monitor*;

• запустить созданное FTP-соединение в программе *Total Commander*;

• открыть в редакторе *Lister* протоколирование работы установленного FTP-соединения;

• зайти на заданную FTP-серверу папку и скачать один файл;

• закрыть FTP-соединение;

• остановить сбор данных и отобразить захваченные кадры;

• сохранить собранные кадры в файле;

• сохранить в файле результаты протоколирования работы установленного FTP-соединения, открытые редактором *Lister*.

3.3.5. С помощью программы *Network Monitor* проанализировать работу при обращении к FTP-серверу, для чего:

• настроить Сетевой монитор, выполнив фильтрацию записанных данных по протоколу FTP;

• проанализировать заголовки запросов и ответов, определить режимы работы клиента и сервера;

• установить соответствие с информацией, полученной с помощью программы *Total Commander*.

3.3.6. В отчете привести:

• содержание заголовков пакетов при обращении к FTP-серверу на этапах идентификации, выбора каталога, чтения файла, завершения процедуры;

• результаты протоколирования работы установленного FTP-соединения с комментариями.

## 3.4. Вопросы для самопроверки

1. Какую роль выполняет протокол HTTP в сети Internet?

2. Какие параметры заголовка протокола HTTP определяют список допустимых форматов ресурса?

3. Что такое URI?

4. Какие параметры HTTP-запроса определяют поддерживаемый язык?

5. В чем особенность работы веб-сервера при обслуживании запроса в режиме ***Keep-Alive****?*

6. Как отфильтровать информацию, принадлежащую протоколу HTTP в захваченных данных?

7. Каково назначение протокола FTP?

8. В чем отличие активного и пассивного режимов работы протокола FTP?

9. Что определяет команда ***TYPE A***?

10. Почему протокол FTP использует два TCP-соединения для передачи данных?

11. Как выполняется в протоколе FTP аутентификация пользователя?

# Список литературы

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010.

2. Уилсон, Э. Мониторинг и анализ сетей. Методы выявления неисправностей / Э. Уилсон. – М. : ЛОРИ, 2002.

3. Филимонов, А. Протоколы Интернета / А. Филимонов. – СПб. : BHV-Санкт-Петербург, 2003.

4. Золотов, С. Протоколы Internet / С. Золотов. – СПб. : BHV-Санкт-Петербург, 1998.

5. Семенов, Ю. А. Telecommunication technologies – телекоммуникационные технологии (v3.3, 10 мая 2010 г.) / Ю. А. Семенов. – URL: http://book.itep.ru/

СОДЕРЖАНИЕ

Учебное издание

**Коннов** Николай Николаевич,
**Механов** Виктор Борисович

# Анализ сетевых протоколов

Лабораторный практикум по курсу
«Сети ЭВМ и телекоммуникации»

Часть 1

Редактор *Т. Н. Судовчихина*
Корректор *Ж. А. Лубенцова*
Компьютерная верстка *Р. Б. Бердниковой*

Подписано в печать 20.10.10. Формат 60x84 1/16.
Усл. печ. л. 3,95. Тираж 100.
Заказ № 646.

---

Издательство ПГУ.
440026, Пенза, Красная, 40.