Оглавление

Лабораторная работа 1.

Форматы кадров Ethernet

Цель работы: изучение форматов кадров Ethernet и особенностей передачи кадров по сети; получение навыков работы с анализаторами пакетов.

1.1 Теоретические сведения

В сетях Ethernet для идентификации узлов используются 6-байтовые **аппаратные** или **MAC-адреса**, которые присваиваются сетевым адаптерам Ethernet. Их принято записывать в виде 6 чисел в шестнадцатеричном формате, разделенных двоеточиями или дефисами, например, 00-01-B5-85-FC-A9. Первые три байта адреса определяют фирму-производителя сетевого адаптера, а остальные — уникальный адрес адаптера.

Некоторые MAC-адреса являются **групповыми** (**широковещательными**), то есть, определяют не отдельный узел, а группу узлов сети. Примером может служить адрес FF-FF-FF-FF-FF-FF (локальный широковещательный адрес). Кадры Ethernet, отправленные на этот адрес, принимают все узлы данной сети.

Существует 4 типа кадров Ethernet:

- кадр 802.3/LLC — описан в стандартах IEEE 802.3 и 802.2;
- кадр Raw 802.3 (или Novell 802.3) — разработан фирмой

Кадр 802.3/LLC

6	6	2	1	1	1 (2)	46 - 1497 (1496)	4
DA	SA	L	DSAP	SSAP	Control	Data	FCS
			Заголовок LLC				

Кадр Raw 802.3/Novell 802.3

6	6	2	46 - 1500	4
DA	SA	L	Data	FCS

Кадр Ethernet II (DIX)

6	6	2	46 - 1500	4
DA	SA	T	Data	FCS

Кадр Ethernet SNAP

6	6	2	1	1	1	1	1	46 - 1492	4
DA	SA	L	DSAP	SSAP	Control	OUI	T	Data	FCS
			Заголовок LLC			Заголовок SNAP			

Рис. 1.1: Форматы кадров Ethernet.

Novell для ускорения работы стека протоколов IPX/SPX в сетях Ethernet;

- кадр Ethernet II (или Ethernet DIX) — представлен консорциумом фирм Digital, Intel и Xerox;
- кадр Ethernet SNAP — разработан комитетом IEEE 802.2 для приведения существующих форматов к некоему общему стандарту.

Форматы кадров всех четырех типов изображены на рис. 1.1

1.1.1 Кадр 802.3/LLC

Формат этого кадра соответствует стандартам IEEE 802.x, согласно которым, канальный уровень в локальных сетях делится на два подуровня.

- Подуровень логической передачи данных (Logical Link Control, LLC) отвечает за передачу данных между узлами локальной сети с требуемой степенью надежности, а также реализует функции интерфейса с сетевым уровнем. Описан в стандарте IEEE 802.2.
- Подуровень управления доступа к среде (Media Access Control) обеспечивает совместное использование общей среды передачи данных узлами сети.

Кадр 802.3/LLC состоит из следующих полей (на рис. 1.1 поле преамбулы и начальный ограничитель кадра не показаны):

- Преамбула (Preamble) занимает 7 байт, необходима для синхронизации работы принимающего и передающего узлов.
- Начальный ограничитель кадра (Start-of-frame-delimiter, SFD) состоит из одного байта, служит указателем на то, что следующий байт — первый байт заголовка кадра.
- Адрес назначения (Destination Address, DA) длиной 6 байт содержит MAC-адрес узла, которому адресован данный кадр.
- Адрес источника (Source Address, SA) длиной 6 байт содержит MAC-адрес узла, отправившего данный кадр.
- Длина (Length, L) — определяет длину поля данных в кадре, занимает 2 байта.
- Адрес точки входа службы назначения (Destination Service Access Point, DSAP) указывает, какой службе верхнего (т.е., сетевого) уровня предназначен данный кадр; занимает 1 байт. Для идентификации служб используются адреса точки входа службы (Service Access Point, SAP), описанные в стандарте 802.2 (см. табл. 1.1).
- Адрес точки входа службы источника (Source Service Access Point, SSAP) также занимает 1 байт и указывает, какая служба сетевого уровня отправила данные, содержащиеся в кадре.

Таблица 1.1: Некоторые адреса точек входа в LLC

Десятичный код	Hex	Описание
6	06	Протокол IP
8	08	Протокол SNA
170	AA	Кадр Ethernet SNAP
224	E0	Протокол IPX
240	F0	Протокол NetBIOS

- **Управляющее поле (Control)** занимает 1 или 2 байта и используется в том случае, если протокол канального уровня передает данные с установлением соединения или с подтверждением приема.
- **Поле данных (Data)** может содержать от 46 до 1497 байт. Если фактическая длина предаваемых данных меньше 46 байт, то поле заполняется нулями до минимально допустимого значения 46 байт.
- **Поле контрольной суммы (Frame Check Sequence, FCS)** состоит из 4 байт, содержащих контрольную сумму кадра, вычисленную по алгоритму CRC-32. Используя это поле, узел назначения может определить, не искажен ли полученный кадр.

Поля DA, SA и L относятся к заголовку подуровня MAC, а поля DSAP, SSAP и Control — к заголовку подуровня LLC.

1.1.2 Кадр Raw 802.3

Этот формат кадра был разработан фирмой Novell для сетей Netware. В них на сетевом уровне используется только один протокол IPX, а протокол канального уровня работает всегда без установления соединения и без подтверждения доставки кадров. Поэтому необходимость в полях заголовка LLC отсутствует, и кадр Raw 802.3 содержит только заголовок уровня MAC (поля DA, SA и L). Максимальная длина поля данных в этом формате кадре равна 1500 байт.

Таблица 1.2: Некоторые коды протоколов в Ethernet II

Десятичный код	Hex	Описание
2048	0800	Протокол IP
2054	0806	Протокол разрешения адреса ARP
32821	8035	Обратный протокол ARP (RARP)
33079-33080	8137-8138	Протокол IPX
33100	814C	Протокол управления сетью SNMP

1.1.3 Кадр Ethernet II

Структура этого кадра совпадает со структурой кадра Raw 802.3, описанной в предыдущем подразделе, однако, вместо поля Длина (L) используется поле Тип (Type, T). В нем указывается код протокола сетевого уровня, который пересылает данные с помощью этого кадра (см. табл. 1.2).

Поле T является аналогом полей DSAP и SSAP заголовка LLC, однако занимает 2 байта, поэтому коды протоколов в Ethernet II не совпадают с однобайтовыми адресами SAP, описанными в стандарте 802.2. Максимальная длина поля данных кадра Ethernet II равна 1500 байт.

1.1.4 Кадр Ethernet SNAP

Кадр этого формата представляет собой расширение формата 802.3/LLC за счет введения дополнительного заголовка SNAP (SubNetwork Access Protocol), состоящего из двух полей.

- Тип (Type, T) занимает 2 байта и определяет адрес точки входа службы сетевого уровня. При этом для идентификации служб используются те же коды протоколов, что и в поле Type кадра Ethernet II.
- Идентификатор организации (Organizationally Unique Identifier) занимает 3 байта и определяет идентификатор организации, ко-

торая контролирует коды протоколов в поле Type. Для IEEE определено значение 0x000000.

Отличительным признаком кадра Ethernet SNAP является значения полей DSAP и SSAP, равные 0xAA.

1.2 Задание

1. Объединить компьютеры в сеть согласно схеме, полученной от преподавателя.
2. Запустить на всех компьютерах анализатор пакетов.
3. Проанализировать перехваченные пакеты, определить, кадры каких форматов используются в данной сети, какие поля заголовка в них используются, какие протоколы сетевого уровня используются в данной сети, какие MAC-адреса имеют узлы сети.

Лабораторная работа 2.

Адресация в IP-сетях

Цель работы: изучение адресации в сетях TCP/IP и разделения сетей на подсети с помощью масок, ознакомление с настройкой стека TCP/IP в различных операционных системах.

2.1 Теоретические сведения

Для адресации узлов в сетях TCP/IP используются 4-байтовые **IP-адреса**. IP-адрес состоит из двух логических частей: номера сети и номера узла. Обычно IP-адрес записывают в виде четырех десятичных чисел, представляющих значения каждого байта, разделенных точками, например **213.182.98.129**. IP-адрес присваивается не компьютеру, а сетевой интерфейсу (например, сетевой карте или соединению через модем).

2.1.1 Классы IP-адресов

Все пространство IP-адресов разделено на 5 классов. К какому классу принадлежит адрес, определяется значением первых бит адреса.

К **классу A** относятся адреса, начинающиеся с **0**. В IP-адресах этого класса номер сети занимает один байт, а остальные три байта определяют номер узла (см. рис. 2.1). Сети класса A должны принадлежать крупным компаниям.

Рис. 2.1: Классы IP-адресов

Если IP-адрес начинается с **10**, то он относится к **классу B**. В этом случае два байта отводится на номер сети, и два байта — на номер узла. Сети класса B — это сети средних компаний.

IP-адреса, начинающиеся с **110**, относятся к **классу C**. Номер сети в сетях этого класса занимает три байта, и один байт отводится на номер узла. Сети класса C выделяются мелким компаниям.

Класс D состоит из адресов, начинающихся на **1110**. Это так называемые групповые адреса (**multicast**), которые могут назначаться сразу нескольким узлам сети. Они используются, например, для организации аудио- и видеоконференций, потокового вещания по сети Internet, а также для обмена информацией между маршрутизаторами.

Адреса, начинающиеся с **11110**, принадлежат к **классу E** и зарезервированы для будущих применений.

Диапазоны номерв сетей, количество возможных сетей и максимальное число узлов, для каждого класса сети приведены в таблице 2.1.

2.1.2 Зарезервированные IP-адреса

Некоторые IP-адреса имеют особое назначение и интерпретируются программным обеспечением узлов и маршрутизаторов специальным образом. Поэтому они не могут быть назначены никаким реальным сетевым интерфейсам.

Таблица 2.1: Классы IP-адресов

Класс	**Наименьший номер сети**	**Наибольший номер сети**	**Число сетей**	**Число узлов в сети**
A	1.0.0.0	126.0.0.0	126	$2^{24} = 16777216$
B	128.0.0.0	191.255.0.0	16320	$2^{16} = 65536$
C	192.0.0.0	223.255.255.0	2080800	$2^8 = 256$
D	224.0.0.0	239.255.255.255	–	–
E	240.0.0.0	247.255.255.255	–	–

- Если номер сети отличен от нуля, а номер узла состоит из одних нулей, то такой IP-адрес интерпретируется как адрес сети в целом.
- Если номер узла состоит из одних единиц, то это **широковещательный IP-адрес**. Пакет, отправленный на широковещательный адрес, доставляется всем узлам данной сети.

Например, все IP-адреса, начинающиеся с **197.13.20**, относятся к сети класса C, адрес которой **197.13.20.0**, а широковещательный адрес — **197.13.20.255**. Так как эти адреса не могут быть назначены никаким узлам, то максимальное число реальных узлов в сети любого класса на два меньше максимального числа IP-адресов для данной сети.

Существует ряд других соглашений об особом использовании некоторых IP-адресов.

- Адреса вида 127.x.x.x назначаются только **псевдоинтерфейсу loopback** (**«обратная петля»**) Пакеты, отправленные на этот интерфейс, не передаются в сеть, а возвращаются модулям транспортного уровня, так, как будто они приняты из сети. Таким образом имитируется работа с сетью.
- Адрес 0.0.0.0 , интерпретируется либо как свой IP-адрес (т.е., пакеты, отправленные на этот адрес, будут приняты тем же узлом, который их отправил), либо любой IP-адрес (таким образом

он используется в модулях маршрутизации в некоторых операционных системах).

- Если в IP-адресе номер сети состоит из одних нулей, а номер узла отличен от нуля, то считается, что он относится к той же сети, что и данный узел (т.е., при отправке пакета на такой адрес номер сети будет заполнен сетевым модулем узла отправителя, исходя из его IP-адреса).

2.1.3 Маски подсетей

В настоящее время используется более гибкая схема разграничения номера сети и номера узла в IP-адресе, основанная на масках подсети. **Маска** — это 4-х байтовое число, использующееся в паре с IP-адресом. Двоичная запись маски содержит единицы в тех разрядах, которые в IP-адресе интерпретируются как номер сети, и нули в разрядах, соответствующих номеру узла. Применение масок позволяет отвести на номер сети любое количество битов. Маски, как и IP-адреса, принято записывать в виде 4-х десятичных чисел, разделенных точками.

Для стандартных классов IP-адресов маски имеют следующие значения:

- класс A — 255.0.0.0;
- класс B — 255.255.0.0;
- класс C — 255.255.255.0.

Пример. При использовании стандартных классов, адреса 212.192.98.45 и 212.192.98.67 относятся к одной сети класса C, номер которой занимает 3 байта, поэтому ее адрес 212.192.98.0.

Пусть теперь для этих адресов указана маска 255.255.255.224. Представим адреса и маску в двоичном виде:

```
212.192.98.45:    11010100.11000000.01100010.00101101
212.192.98.67:    11010100.11000000.01100010.01000011
255.255.255.224:  11111111.11111111.11111111.11100000
```

Отсюда видно, что адрес 212.192.98.45 теперь принадлежит сети, имеющей адрес в двоичной записи

```
11010100.11000000.01100010.00100000,
```

или 212.192.98.32 в десятичной. Адрес 212.192.98.67 теперь относится к *другой* сети, адрес которой в двоичной записи

```
11010100.11000000.01100010.01000000,
```

а в десятичной — 212.192.98.64.

Таким образом, с помощью маски единая сеть класса C оказывается разделенной на несколько **подсетей**. Их количество определяется тем, сколько «лишних» единиц (по сравнению со стандартной для сети класса C маской 255.255.255.0) содержится в заданной маске. В нашем примере их 3, то есть, на номер сети отводится дополнительно 3 бита, что позволяет адресовать $2^3 = 8$ подсетей.

С помощью масок можно также объединить несколько сетей данного класса в одну **надсеть**. Например, адрес 212.192.96.0 и маска 255.255.252.0 определяют надсеть, составленную из 4-х сетей класса C с адресами 212.192.96.0, 212.192.97.0, 212.192.98.0 и 212.192.99.0 (такая возможность часто используется в таблицах маршрутизации).

2.2 Задание

1. Даны адрес сети и маска. Определить:

   - на сколько подсетей разбивает указанная маска данную сеть и какое максимальное количество узлов будут содержать подсети;
   - какие адреса будут иметь подсети;
   - какие широковещательные адреса будут у созданных подсетей;
   - на сколько (по сравнению со стандартным для данного класса) увеличится количество IP-адресов, которых нельзя назначить реальным узлам?

2. Даны адрес сети, маска и ряд IP-адресов. Определить, какие из этих адресов принадлежат указанной сети.

3. Даны адрес сети и маска. Назначить адреса из этой сети реальным компьютерам, сконфигурировать TCP/IP протокол, проверить работоспособность сети.

Вариант 1.

1. Сеть: 215.62.216.0, маска: 255.255.255.192;
2. Сеть: 213.215.230.128, маска: 255.255.255.240, адреса: 213.215.230.126, 213.215.230.131, 213.215.230.140, 213.215.230.145;
3. Сеть: 192.168.162.80, маска: 255.255.255.240.

Вариант 2.

1. Сеть: 206.235.25.0, маска: 255.255.255.128;
2. Сеть: 193.187.146.208, маска: 255.255.255.240, адреса: 193.187.146.209, 193.187.146.217, 193.187.146.220, 193.187.146.226;
3. Сеть: 192.168.172.192, маска: 255.255.255.224.

Вариант 3.

1. Сеть: 222.8.188.0, маска: 255.255.255.240;
2. Сеть: 202.100.248.0, маска: 255.255.255.240, адреса: 202.100.248.5, 202.100.248.8, 202.100.248.13, 202.100.248.18;
3. Сеть: 192.168.174.48, маска: 255.255.255.240.

Вариант 4.

1. Сеть: 199.95.127.0, маска: 255.255.255.128;
2. Сеть: 209.242.236.64, маска: 255.255.255.248, адреса: 209.242.236.62, 209.242.236.66, 209.242.236.70, 209.242.236.73;
3. Сеть: 192.168.189.0, маска: 255.255.255.128.

Вариант 5.

1. Сеть: 206.217.107.0, маска: 255.255.255.240;
2. Сеть: 194.7.7.0, маска: 255.255.255.224, адреса: 194.7.7.4, 194.7.7.11, 194.7.7.23, 194.7.7.37;
3. Сеть: 192.168.173.32, маска: 255.255.255.224.

Вариант 6.

1. Сеть: 199.216.194.0, маска: 255.255.255.224;
2. Сеть: 220.71.119.16, маска: 255.255.255.240, адреса: 220.71.119.13, 220.71.119.17, 220.71.119.28, 220.71.119.34;
3. Сеть: 192.168.181.128, маска: 255.255.255.224.

Вариант 7.

1. Сеть: 197.239.196.0, маска: 255.255.255.192;
2. Сеть: 205.166.73.128, маска: 255.255.255.128, адреса: 205.166.73.125, 205.166.73.130, 205.166.73.161, 205.166.73.230;
3. Сеть: 192.168.186.64, маска: 255.255.255.224.

Вариант 8.

1. Сеть: 212.232.143.0, маска: 255.255.255.248;
2. Сеть: 199.57.154.0, маска: 255.255.255.192, адреса: 199.57.154.2, 199.57.154.29, 199.57.154.43, 199.57.154.78;
3. Сеть: 192.168.170.224, маска: 255.255.255.224.

Вариант 9.

1. Сеть: 211.108.87.0, маска: 255.255.255.192;
2. Сеть: 195.217.172.96, маска: 255.255.255.224, адреса: 195.217.172.91, 195.217.172.99, 195.217.172.113, 195.217.172.129;
3. Сеть: 192.168.164.176, маска: 255.255.255.240.

Вариант 10.

1. Сеть: 209.204.141.0, маска: 255.255.255.192;
2. Сеть: 198.219.31.0, маска: 255.255.255.128, адреса: 198.219.31.10, 198.219.31.44, 198.219.31.93, 198.219.31.114;
3. Сеть: 192.168.162.192, маска: 255.255.255.192.

Вариант 11.

1. Сеть: 221.191.33.0, маска: 255.255.255.248;
2. Сеть: 222.136.137.24, маска: 255.255.255.248, адреса: 222.136.137.21, 222.136.137.25, 222.136.137.30, 222.136.137.38;
3. Сеть: 192.168.170.112, маска: 255.255.255.240.

Вариант 12.

1. Сеть: 214.75.92.0, маска: 255.255.255.240;
2. Сеть: 217.26.170.208, маска: 255.255.255.240, адреса: 217.26.170.204, 217.26.170.211, 217.26.170.219, 217.26.170.226;
3. Сеть: 192.168.176.48, маска: 255.255.255.240.

Вариант 13.

1. Сеть: 193.207.246.0, маска: 255.255.255.224;
2. Сеть: 205.229.86.160, маска: 255.255.255.224, адреса: 205.229.86.162, 205.229.86.175, 205.229.86.189, 205.229.86.194;
3. Сеть: 192.168.166.144, маска: 255.255.255.240.

Вариант 14.

1. Сеть: 192.199.148.0, маска: 255.255.255.248;
2. Сеть: 221.56.18.192, маска: 255.255.255.192, адреса: 221.56.18.190, 221.56.18.196, 221.56.18.221, 221.56.18.241;
3. Сеть: 192.168.187.64, маска: 255.255.255.240.

Вариант 15.

1. Сеть: 199.109.95.0, маска: 255.255.255.240;
2. Сеть: 223.38.150.32, маска: 255.255.255.240, адреса: 223.38.150.30, 223.38.150.35, 223.38.150.41, 223.38.150.50;
3. Сеть: 192.168.184.176, маска: 255.255.255.240.

Лабораторная работа 3.

Маршрутизация в сетях IP

Цель работы: изучение маршрутизации в сетях IP, ознакомление с настройками маршрутизации в различных операционных системах, получение навыков работы с утилитой *traceroute*.

3.1 Теоретические сведения

3.1.1 Принципы маршрутизации

С точки зрения сетевого уровня, сеть является **составной**, то есть, представляет собой совокупность нескольких сетей, соединенных между собой **маршрутизаторами**. Для передачи данных в составляющих сетях используются некоторые, не обязательно одинаковые, технологии канального уровня.

В модели TCP/IP для идентификации узлов составной сети и составляющих сетей в целом используются IP-адреса (см. Лабораторную работу № 3). Маршрутизаторы по определению входят в несколько сетей одновременно, поэтому каждый порт (сетевой интерфейс) маршрутизатора имеет собственный IP-адрес.

Сети, в которые входит данный узел или маршрутизатор называются **непосредственно присоединенными**; все остальные сети являются **удаленными**. Доставка данных узлам удаленных сетей возможна

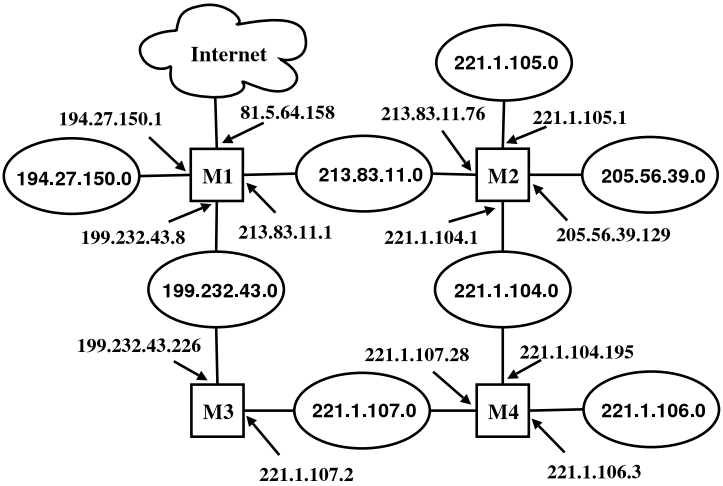

Рис. 3.1: Пример составной сети.

только через маршрутизаторы. Последовательность маршрутизаторов, которые должен пройти пакет от узла отправления до узла назначения называется **маршрутом**.

Пример. В сети, изображенной на рис. 3.1, для маршрутизатора M4 непосредственно присоединенными сетями являются 221.1.107.0, 221.1.104.0 и 221.1.106.0. Все остальные сети являются удаленными. Маршрут от узла, принадлежащего сети 221.1.106.0, до узла из сети 213.83.11.0 может выглядеть как M4 – M2; существует также альтернативный маршрут M4 – M3 – M1. Задачу выбора наиболее рационального маршрута из нескольких альтернативных решают маршрутизаторы. Конкретные способы решения этой задачи зависят от реализации стека TCP/IP на маршрутизаторах.

Следует отметить, что в IP-сетях при выборе маршрута определяется только следующий шаг маршрута (адрес следующего маршрутизатора), а не вся последовательность маршрутизаторов от начального до конечного узла. Такой алгоритм маршрутизации называется **одношаговым**.

3.1.2 Таблицы маршрутизации

Для определения маршрута доставки пакета, конечные узлы и маршрутизаторы используют специальную информационную структуру — **таблицу маршрутизации**. Каждая запись этой таблицы содержит информацию о способе доставки пакета до определенной сети или хоста. Запись состоит из нескольких полей. Следующие поля являются обязательными:

- IP-адрес сети (или хоста);
- маска сети (в случае хоста указывается маска 255.255.255.255);
- IP-адрес следующего маршрутизатора (точнее, порта маршрутизатора), на который нужно отправить пакет, чтобы доставить его в сеть с указанными адресом и маской.

Запись может содержать также другие поля, например, идентификатор сетевого интерфейса, через который нужно отправлять пакет на следующий маршрутизатор, расстояние до сети назначения в определенных единицах (т.н., **метрику**), и др. Явный вид записи зависит от конкретной реализации стека TCP/IP.

В таблице маршрутизации в записях, соответствующих непосредственно присоединенным сетям, в качестве адреса следующего маршрутизатора используется адрес сетевого интерфейса, соединяющего маршрутизатор с данной сетью (либо специальный адрес 0.0.0.0, который в данном случае понимается как «один из адресов данного маршрутизатора, принадлежащий к указанной сети»). Адреса маршрутизаторов, через которые доступны удаленные сети, обязательно должны принадлежать непосредственно присоединенным сетям.

В таблицах маршрутизации обычно также содержится запись, соответствующая «**маршрутизатору по умолчанию**». Считается, что все сети, маршруты до которых не указаны явно, доступны через этот маршрутизатор. Использование маршрутизации по умолчанию позволяет сократить объем таблиц маршрутизации, так как избавляет от необходимости прописывать в них маршрут до каждой сети Internet. В таблицах маршрутизации эта запись обозначается ключевым словом **default**, либо в ней в качестве IP-адреса сети и маски подсети указывается 0.0.0.0.

Когда на маршрутизатор поступает пакет, из его заголовка извлекается IP-адрес узла назначения. Затем последовательно перебираются все записи таблицы маршрутизации и выполняются следующие действия:

- маска подсети, извлеченная из записи, накладывается на IP-адрес узла назначения;
- полученный адрес сети сравнивается с адресом сети из данной записи;
- если эти адреса совпадают, пакет отправляется на адрес узла назначения, если сеть является непосредственно подключенной, или на адрес маршрутизатора, указанный в записи, если сеть удаленная;
- если адреса не совпадают, рассматривается следующая запись.

Если IP-адрес узла назначения не принадлежит ни одной из сетей, содержащихся в таблице маршрутизации, то он отправляется на адрес маршрутизатора по умолчанию.

Пример. Составим таблицу маршрутизации для маршрутизатора M2 (см. рис. 3.1).

Сети 221.1.105.0, 205.56.39.0, 213.83.11.0 и 221.1.104.0 являются непосредственно подключенными, поэтому в качестве адреса следующего маршрутизатора нужно указать 0.0.0.0. Сети 221.1.106.0 и 221.1.107.0 доступны через маршрутизатор M4. Из трех его адресов в сети, непосредственно подключенной к M2, находится 221.1.104.195, поэтому его и нужно указать в качестве адреса следующего маршрутизатора. Сети 199.232.43.0 и 194.27.150.0 доступны через M1, адресом следующего маршрутизатора для них является 213.83.11.1. Остальные сети Internet также доступны через маршрутизатор M1, поэтому он будет маршрутизатором по умолчанию.

В результате таблица маршрутизации будет иметь вид:

Адрес сети	Маска	Адрес шлюза
221.1.105.0	255.255.255.0	0.0.0.0
205.56.39.0	255.255.255.0	0.0.0.0
213.83.11.0	255.255.255.0	0.0.0.0
221.1.104.0	255.255.255.0	0.0.0.0
221.1.106.0	255.255.255.0	221.1.104.195
221.1.107.0	255.255.255.0	221.1.104.195
199.232.43.0	255.255.255.0	213.83.11.1
194.27.150.0	255.255.255.0	213.83.11.1
default		213.83.11.1

Таким образом, пакет, пришедший на маршрутизатор M2 и адресованный узлу с IP-адресом 221.1.107.45, будет отправлен на адрес 221.1.104.195, а пакет с адресом назначения 212.192.97.18 — на адрес 213.83.11.1.

Замечание 1. Записи, соответствующие сетям 199.232.43.0 и 194.27.150.0, можно исключить из таблицы маршрутизации, т.к. они доступны через маршрутизатор M1, указанный в качестве маршрутизатора по умолчанию.

Конечные узлы для определения маршрута доставки пакетов также используют таблицу маршрутизации. Однако эта таблица обычно состоит только из записи о сети, к которой принадлежит данный хост, и записи о маршрутизаторе по умолчанию.

3.2 Задание

Для выполнения данной работы сеть должна иметь структуру, изображенную на рис 3.2.

1. Назначить узлам сети и сетевым интерфейсам маршрутизаторов M1, M2 и M3 IP-адреса из соответствующих сетей.
2. Настроить таблицы маршрутизации на маршрутизаторах M1, M2 и M3 так, чтобы информация корректно доставлялась меж-

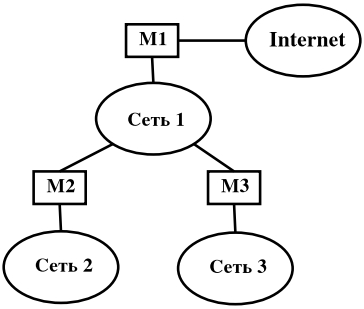

Рис. 3.2: Структура сети для выполнения лабораторной работы по маршрутизации.

ду любыми двумя хостами данной составной сети, а также во внешнюю сеть.

3. Проверить работоспособность сети, используя утилиты *ping* и *tracert* (см. Приложение D.).

Вариант 1. Сеть 1: 192.168.183.240, маска: 255.255.255.248. Сеть 2: 192.168.164.64, маска: 255.255.255.192. Сеть 3: 192.168.179.192, маска: 255.255.255.240.

Вариант 2. Сеть 1: 192.168.176.0, маска: 255.255.255.224. Сеть 2: 192.168.175.128, маска: 255.255.255.128. Сеть 3: 192.168.178.208, маска: 255.255.255.248.

Вариант 3. Сеть 1: 192.168.189.64, маска: 255.255.255.192. Сеть 2: 192.168.168.128, маска: 255.255.255.128. Сеть 3: 192.168.172.240, маска: 255.255.255.248.

Вариант 4. Сеть 1: 192.168.175.80, маска: 255.255.255.240. Сеть 2: 192.168.169.128, маска: 255.255.255.128. Сеть 3: 192.168.185.128, маска: 255.255.255.224.

Вариант 5. Сеть 1: 192.168.180.152, маска: 255.255.255.248. Сеть 2: 192.168.181.192, маска: 255.255.255.192. Сеть 3: 192.168.181.128, маска: 255.255.255.128.

Вариант 6. Сеть 1: 192.168.170.192, маска: 255.255.255.248. Сеть 2: 192.168.172.128, маска: 255.255.255.128. Сеть 3: 192.168.189.96, маска: 255.255.255.240.

Вариант 7. Сеть 1: 192.168.189.224, маска: 255.255.255.224. Сеть 2: 192.168.191.64, маска: 255.255.255.192. Сеть 3: 192.168.165.128, маска: 255.255.255.240.

Вариант 8. Сеть 1: 192.168.190.192, маска: 255.255.255.248. Сеть 2: 192.168.167.0, маска: 255.255.255.192. Сеть 3: 192.168.187.128, маска: 255.255.255.128.

Вариант 9. Сеть 1: 192.168.181.0, маска: 255.255.255.224. Сеть 2: 192.168.191.192, маска: 255.255.255.192. Сеть 3: 192.168.178.128, маска: 255.255.255.240.

Вариант 10. Сеть 1: 192.168.178.0, маска: 255.255.255.128. Сеть 2: 192.168.190.128, маска: 255.255.255.224. Сеть 3: 192.168.191.32, маска: 255.255.255.224.

Вариант 11. Сеть 1: 192.168.171.128, маска: 255.255.255.240. Сеть 2: 192.168.171.192, маска: 255.255.255.192. Сеть 3: 192.168.173.0, маска: 255.255.255.248.

Вариант 12. Сеть 1: 192.168.163.160, маска: 255.255.255.224. Сеть 2: 192.168.179.192, маска: 255.255.255.248. Сеть 3: 192.168.191.0, маска: 255.255.255.128.

Вариант 13. Сеть 1: 192.168.167.176, маска: 255.255.255.240. Сеть 2: 192.168.172.184, маска: 255.255.255.248. Сеть 3: 192.168.172.80, маска: 255.255.255.240.

Вариант 14. Сеть 1: 192.168.173.192, маска: 255.255.255.192. Сеть 2: 192.168.170.136, маска: 255.255.255.248. Сеть 3: 192.168.181.64, маска: 255.255.255.224.

Вариант 15. Сеть 1: 192.168.164.64, маска: 255.255.255.192. Сеть 2: 192.168.171.0, маска: 255.255.255.224. Сеть 3: 192.168.172.16, маска: 255.255.255.240.

Лабораторная работа 4.

Форматы пакетов сетевого уровня

Цель работы: ознакомление с форматами пакетов протоколов IP и ICMP, изучение обработки пакетов в процессе их доставки,

4.1 Теоретические сведения

4.1.1 Формат заголовка IP-пакета

IP-пакет состоит из заголовка и данных. Заголовок пакета (рис. 4.1.1) имеет переменную длину, всегда кратную 32 бит. Непосредственно за заголовком следуют данные, передаваемые в дейтаграмме.

Ниже приведены значения полей заголовка.

- Версия (Version, Ver) (4 бита) — версия протокола IP, в настоящий момент используется версия 4 (IPv4), в будущем планируется переход на версию 6 (IPv6).

- Длина заголовка (Internet Header Length, IHL) (4 бита) — длина заголовка в 32-битных словах. Минимальная длина заголовка — 5, (20 байт, в этом случае поле Options отсутствует), максимальная — 15 (60 байт, из них 40 байт опций).

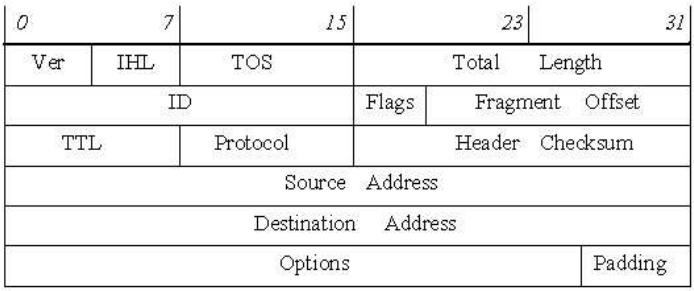

Рис. 4.1: Формат заголовка IP-пакета.

- **Тип сервиса (Type Of Service, TOS)** (8 бит) — приоритет дейтаграммы и желаемый тип маршрутизации (см. подраздел 4.1.2).

- **Общая длина (Total Length)** (16 бит) — длина пакета в байтах, включая заголовок и данные. Максимальное значение — 65535, минимальное — 21 (заголовок без опций и один байт в поле данных).

- **Идентификатор пакета (Identification, ID)** (16 бит) — используется для распознавания пакетов при фрагментации (см. подраздел 4.1.3). У всех пакетов, полученных при фрагментации исходного пакета значение этого поля должно быть одинаковым.

- **Флаги (Flags)** (3 бит) — признаки пакета, связанные с фрагментацией (см. подраздел 4.1.3).

- **Смещение фрагмента (Fragment Offset)** (13 бит) — задает смещение (в байтах) поля данных фрагмента от начала поля данных исходного пакета.

- **TTL (Time To Live)** (8 бит) — время жизни пакета. Оно устанавливается отправителем, формально измеряется в секундах. Каждый маршрутизатор, через который проходит пакет, уменьшает значение TTL на единицу. Поэтому, фактически, **TTL** определяет максимальное количество узлов сети, через которые разрешено пройти пакету, прежде чем он достигнет узла назначения. При достижении значения **TTL** = 0 пакет уничтожается (при этом

Таблица 4.1: Коды некоторых протоколов Internet

Десятичный код	**Ключевое слово**	**Протокол**
1	ICMP	Протокол служебных сообщений ICMP
2	IGMP	Протокол управления группами IGMP
4	IPIP	Инкапсуляция IP-пакетов в IP-пакеты (туннелирование)
6	TCP	Транспортный протокол TCP
8	EGP	Протокол внешней маршрутизации EGP
17	UDP	Транспортный протокол UDP
88	IGRP	Протокол групповой маршрутизации IGRP
89	OSPF	Протокол динамической маршрутизации OSPF

отправителю может быть послано соответствующее ICMP-сообщение). Это предотвращает бесконечное зацикливание пакетов в сети.

- Протокол (Protocol) (8 бит) — определяет, какому протоколу верхнего уровня предназначена информация, содержащаяся в поле данных пакета. Значения кодов некоторых протоколов приведены в табл. 4.1.

- Контрольная сумма (Header Checksum) (16 бит) — контрольная сумма заголовка, которая подсчитывается как дополнение к сумме всех 16-битовых слов заголовка. Поскольку в процессе доставки пакета значения некоторых полей заголовка (например, поля TTL) изменяются, контрольная сумма каждым маршрутизатором пересчитывается заново. При вычислении контрольной

суммы значение самого поля Header Checksum обнуляется. Если при проверке контрольной суммы обнаруживается ошибка, пакет уничтожается.

- IP-адрес источника (Source Address) (32 бит) — IP-адрес узла-отправителя.
- IP-адрес назначения (Destination Address) (32 бит) — IP-адрес узла-получателя.
- Опции (Options) — опции, необязательное поле переменной длины (может быть одна опция, несколько или ни одной). Опции могут использоваться для различных целей, например, содержать отладочную информацию, регистрировать проходимые пакетом маршрутизаторы, определять дополнительные способы обработки пакета модулями IP, и т.д.
- Выравнивание (Padding) — используется для выравнивания заголовка по границе 32-битного слова, если поле Options занимает нецелое число 32-битных слов. Поле Padding заполняется нулями.

4.1.2 Тип сервиса

Разработчики протокола IP предусмотрели возможность указания типа сервиса для каждого IP-пакета. Это позволяет организовать более быструю доставку некоторых пакетов до узла назначения, а также задать критерии выбора маршрута доставки пакета при наличии нескольких альтернативных маршрутов. Для этого в заголовке IP-пакета определено поле Type of Service.

Первые три бита поля Type of Service образуют подполе Precedence и определяют приоритет пакета от самого низкого (0) до самого высокого (7). Ниже приведены возможные значения этого подполя

- 000 — обычный уровень;
- 001 — приоритетный;
- 010 — немедленный;
- 011 — срочный;
- 100 — экстренный;

- 101 — критический (CRITIC/ECP);
- 110 — межсетевое управление;
- 111 — управление сетью.

Биты D, T, R, C определяют способ доставки пакета при наличии нескольких альтернативных маршрутов до узла назначения. Если установлен бит D (Delay), то выбирается маршрут с минимальной задержкой, бит T (Throughput) — с максимальной пропускной способностью, бит R (Reliability) — с максимальной надежностью, бит C (Cost) — с минимальной стоимостью. В заголовке пакета может быть установлен только один из битов D, T, R, C. Последний бит поля Type of Service не используется.

Реально использование поля TOS зависит от маршрутизатора и его настроек. Маршрутизатор может поддерживать обработку всех типов TOS, только некоторых из них или игнорировать TOS вообще. Кроме того, маршрутизатор может быть настроен так, чтобы учитывать значение приоритета только при обработке пакетов, исходящих из некоторого ограниченного множества узлов сети, или вовсе игнорировать приоритет.

4.1.3 Фрагментация дейтаграмм

Различные среды передачи имеют разный максимальный размер передаваемого блока данных (Maximum Transfer Unit, MTU). Например, в технологии Ethernet размер MTU равен 1500 байт, в FDDI — 4096 байт, и т.д.

При передаче пакета из среды с большим MTU в среду с меньшим MTU может возникнуть необходимость во фрагментации пакета. Фрагментация и сборка пакета осуществляются модулем протокола IP. При этом используются поля Identification, Flags и Fragment Offset заголовка пакета.

При фрагментации исходный пакет разбивается на несколько фрагментов, размер которых (с учетом длины заголовка IP-пакета) не превышает меньшего MTU. Все фрагменты имеют одинаковое значение поля Identification (устанавливается узлом, производящим фрагментацию). Поле Fragment Offset указывает, на какой позиции в поле данных исходного пакета находится данный фрагмент (смещение должно быть кратно 8). У первого фрагмента смещение равно нулю.

Поле **Flags** состоит из 3 бит, первый из которых всегда сброшен. Второй бит — **DF (Don't Fragment)** — служит для управления фрагментацией на промежуточных маршрутизаторах (0 — фрагментация разрешена, 1 — фрагментация запрещена). Третий бит — **MF (More Fragments)** — определяет, является ли данный фрагмент последним (0) или за ним следуют другие фрагменты (1).

У нефрагментированых IP-пакетов бит **MF** и поле **Fragment Offset** устанавливаются в ноль.

Необходимость во фрагментации возникает в двух случаях:

- Протокол верхнего уровня (обычно UDP или ICMP) передает модулю IP для отправки пакет, размер которого превышает MTU используемой среды передачи данных. Тогда процедура фрагментации осуществляется на узле-отправителе.
- На промежуточный маршрутизатор приходит пакет, размер которого превышает MTU для того сетевого интерфейса, через который его нужно доставить. Тогда фрагментацию должен производить маршрутизатор. Однако, если в заголовке пакета установлен бит **DF**, то такой пакет уничтожается, а узлу-отправителю посылается ICMP-сообщение о необходимости фрагментации пакетов, отправляемых на данный узел назначения.

Сборку исходного пакета из фрагментов всегда осуществляет узел назначения. При получении первого фрагмента модуль IP выделяет буфер, в который помещаются поступающие фрагменты, и запускает таймер, определяющий максимальное время ожидания прихода остальных фрагментов пакета. Если таймер истекает ранее прибытия последнего фрагмента, то все полученные до этого фрагменты уничтожаются. Отправителю посылается сообщение об ошибке с помощью протокола ICMP.

4.1.4 Протокол служебных сообщений ICMP

ICMP-протокол служит для передачи различных служебных сообщений. Эти сообщения позволяют проводить диагностику сети, запрашивать некоторую дополнительную информацию, необходимую для работы сети, а также сообщать о сбоях и ошибках.

ICMP-сообщения различаются по типам и кодам (см. табл. 4.2). Тип определяет, какая информация содержится в данном сообщении, а код уточняет ее.

Все ICMP-пакеты начинаются с полей **Тип (Type)** (8 бит), **Код (Code)** (8 бит) и **Контрольная сумма заголовка (Checksum)** (16 бит, вычисляется по заголовку ICMP при отправке пакета). Какие еще поля есть в заголовке, зависит от типа и кода сообщения.

Сообщения типов 8 («*Echo Request*») и 0 («*Echo Reply*») используются для диагностики работы сети (например, они генерируются командами ping и traceroute). Узел, получивший эхо-запрос, посылает на адрес его отправителя эхо-ответ. В заголовках этих сообщений присутствуют поля **Идентификатор (Identifier)** и **Номер по порядку (Sequence Number)**. Они служат для связи между запросов и откликов между собой.

С помощью информационных запросов (типы 10, 13, 17) узлы могут получить дополнительную информацию о сети; ответами на них служат сообщения типов 9, 14, 18. В их заголовках присутствуют поля, которые содержат запрошенную информацию (адрес маршрутизатора, маску подсети и т.д.). Большинство современных ОС игнорирует информационные сообщения из соображений безопасности.

Сообщения об ошибках (типы 3, 4, 5, 11, 12) служат для оповещения узла-отправителя о проблемах, возникших в процессе обработки IP-пакетов на маршрутизаторах или узлах назначения. В их заголовки включается IP-заголовок и 64 байта данных (в которые попадают заголовки протоколов транспортного уровня) IP-пакета, вызвавшего ошибку. В заголовок сообщения типа 5 включается также адрес маршрутизатора, который нужно в дальнейшем использовать для доставки пакетов, подобных тому пакету, в ответ на который послано данное сообщение.

ICMP-сообщения об ошибках никогда не порождаются при невозможности доставки:

- ICMP-сообщений об ошибках;
- не первых фрагментов IP-пакетов;
- пакетов, направленных по групповому или широковещательному адресу;
- пакетов, адрес отправителя которых нулевой, широковещательный или групповой.

Это позволяет уменьшить количество передаваемых служебных сообщений и предотвратить рассылку ICMP-сообщений на широковещательные и групповые адреса.

Таблица 4.2: Типы и коды ICMP-сообщений.

Тип	**Код**	**Описание**
0	0	Эхо-ответ («*Echo Reply*»)
3		Адресат недостижим («*Destination Unreachable*»)
	0	сеть недоступна («*Net Unreachable*»)
	1	хост недоступен («*Host Unreachable*»)
	2	протокол недоступен («*Protocol Unreachable*»)
	3	порт недоступен («*Port Unreachable*»)
	4	необходима фрагментация, но она запрещена (установлен флаг DF)
	5	невозможно выполнить маршрутизацию от источника («*Source Route failed*»)
	9	связь с сетью назначения административно запрещена
	10	связь с хостом назначения административно запрещена
	13	связь административно запрещена с помощью фильтра
4	0	Отключение источника («*Source Quench*»)
5		Выбрать другой маршрутизатор для доставки пакетов («*Redirect*»)
	0	в данную сеть
	1	на данный хост
	2	в данную сеть с данным TOS
	3	на данный хост с данным TOS
8	0	Эхо-запрос («*Echo Request*»)
9	0	Объявление маршрутизатора («*Router Advertisement*»)
10	0	Запрос объявления маршрутизатора («*Router Solicitation*»)

11		Время жизни дейтаграммы истекло («*Time Exceeded*»)
	0	при передаче
	1	при сборке
12		Ошибка в параметрах («*Parameter problem*»)
	0	ошибка в IP-заголовке
	1	отсутствует необходимая опция
13	0	Запрос временной метки («*Timestamp*»)
14	0	Ответ на запрос временной метки («*Timestamp Reply*»)
17	0	Запрос сетевой маски («*Address Mask Request*»)
18	0	Ответ на запрос сетевой маски («*Address Mask Reply*»)

4.2 Задание

1. С помощью анализатора пакетов перехватить 20 – 30 IP-пакетов. Для любых 5 пакетов, указанных преподавателем, определить их харакеристики (длину заголовка и общую длину пакета, тип сервиса, время жизни, протокол верхнего уровня, адреса отправителя и получателя, опции, является ли данный пакет фрагментом).
2. Запустив анализатор пакетов, выполнить указанную преподавателем команду. Проанализировав перехваченные ICMP-сообщения, определить, какая проблема возникает при передаче данных. Описать формат заголовка соответствующего ICMP-сообщения.

Лабораторная работа 5.

Протокол TCP

Цель работы: изучение формата TCP-сегментов, процедур установления и разрыва TCP-соединения и особенностей работы протокола TCP.

5.1 Теоретические сведения

Протокол TCP реализует надежную доставку данных с предварительным установлением соединения между приложениями, работающими на различных узлах сети. На базе этого протокола работают многие службы сети Internet.

5.1.1 TCP-соединение

TCP-соединение представляет собой двунаправленный поток данных между приложениями на концах соединения. Каждая сторона соединения идентифицируется двумя параметрами — IP-адресом хоста, на котором работает приложение, и номером **порта**, который присваивается очереди обмена данными между TCP-модулем и приложением. Пару «IP-адрес — порт» называют **сокетом**. TCP-соединение идентифицируется парой сокетов.

Данные передаются в виде пакетов, которые принято называть **сегментами**. Для реализации контроля доставки данных, каждый октет (байт) в потоке нумеруется. Контроль доставки данных осуществля-

ется с помощью **метода позитивного подтверждения и повторной передачи**. Если данные доставлены получателю без ошибок, то получатель подтверждает это, посылая отправителю специальный пакет (**квитанцию**). Если в течение некоторого времени отправитель не получает подтверждения доставки, то он отправляет данные повторно. Если после трех попыток отправить сегмент подтверждение о доставке не получено, то инициируется разрыв данного TCP-соединения.

Процедуру установления TCP-соединения принято называть «**тройным рукопожатием**», так как при этом присходит обмен тремя служебными TCP-пакетами. Приложение, которое инициирует соединение (клиент), отправляет на сервер запрос, содержащий в заголовках следующие параметры:

- начальное значение порядкового номера октета в потоке (**Initial Sequence Number, ISN**) со стороны клиента (алгоритм выбора ISN зависит от конкретной реализации протокола TCP);
- максимальный размер TCP-сегмента (**Maximum Segment Size, MSS**) для данного соединения;
- сведения о расширениях протокола TCP, которые поддерживает клиент.

Сервер, получив данный запрос, отправляет клиенту ответ, в котором подтверждает получение запроса (а, следовательно, и параметров сеоединения), и посылает аналогичный набор параметров со своей стороны. Затем клиент подтверждает на получение этого пакета и TCP-соединение считается установленным. Таким образом, клиент и сервер в процессе установления TCP-соединения сообщают друг другу и согласовывают (синхронизируют) параметры соединения

5.1.2 Формат TCP-сегмента

TCP-сегмент состоит из заголовка и данных. Заголовок сегмента имеет переменную длину, зависящую от размера поля Options, но всегда кратную 32 битам. За заголовком непосредственно следуют данные. Формат заголовка TCP-сегмента представлен на рис. 5.1.2.

Значения полей заголовка следующие.

- Source Port (16 бит) — номер порта процесса-отправителя.
- Destination Port (16 бит) — номер порта процесса-получателя.

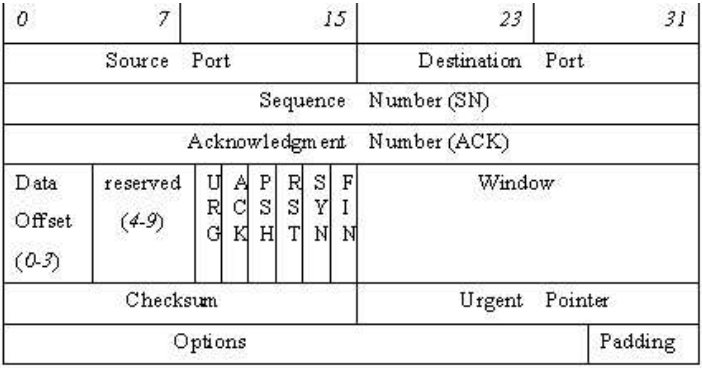

Рис. 5.1: Формат заголовка TCP-пакета.

- Sequence Number (SN) (32 бит) — порядковый номер первого октета в поле данных сегмента в общем потоке данных текущего TCP-соединения.
- Acknowledgement Number (ACK) (32 бит) — номер подтверждения, сообщающий отправителю количество полученных от него октетов; определяется как номер первого неподтвержденного октета в потоке.
- Data Offset (4 бита) — смещение поля данных относительно начала сегмента (т.е., фактически, длина TCP-заголовка) в 32-битных словах. При минимальном размере заголовка значение этого поля равно 5 словам (20 байт).
- Reserved (6 бит) — зарезервировано; заполняется нулями.
- Control Bits (6 бит) — управляющие биты (флаги). Ниже приведен их список.
  - URG — указывает, что сегмент содержит срочные данные, положение которых в сегменте определяется значением поля Urgent Pointer (см. ниже).
  - ACK — указывает, что данный сегмент является квитанцией (подтверждением).
  - PSH — указывает, что нужно немедленно передать все буферизованные данные, не дожидаясь заполнения сегмента.

  - RST — указывает на необходимость немедленного разрыва текущего TCP-соединения.
  - SYN — указывает, что сегмент содержит начальное значение порядкового номера октета в потоке данных (ISN) в поле Sequence Number; такие сегменты являются частью процедуры установления TCP-соединения.
  - FIN — указывает, что сегмент является сообщением о прекращении передачи данных одной из сторон соединения.

- Window (16 бит) — размер окна приема в октетах; определяет, сколько октетов готов принять получатель. На основании значения этого поля отправитель пакета вычисляет размер окна передачи, то есть, объем данных, которые можно отправить получателю, не дожидаясь подтверждения ранее посланных данных. С помощью этого поля стороны TCP-соединения могут регулировать скорость передачи данных.
- Checksum (16 бит) — контрольная сумма, которая вычисляется по 96-битному псевдозаголовку, содержащему кроме заголовка TCP еще некоторые поля заголовка IP. Такой подход обеспечивает дополнительную защиту протокола TCP от переданных с ошибкой сегментов.
- Urgent Pointer (16 бит) — используется для указания длины срочных данных, которые размещаются в начале поля данных сегмента. Указывает смещение октета, следующего за срочными данными, относительно первого октета в сегменте. Поле Urgent Pointer задействовано, если установлен флаг URG.
- Options — поле переменной длины, может отсутствовать, либо содержать различные опции.
- Padding — поле выравнивания заголовка по границе 32-битного слова, если список опций занимает нецелое число 32-битных слов. Поле Padding заполняется нулями.

5.1.3 Опции

Опции TCP содержат дополнительную служебную информацию. Опция состоит из октета Тип опции, за которым могут следовать октет Длина опции (в октетах) и октеты с данными для опции.

Стандарт протокола TCP определяет три опции (типы 0, 1, 2).

Опция типа 0 (**Конец списка опций**) состоит только из одного октета (**Тип опции**). При обнаружении данной опции дальнейший разбор опций прекращается, даже если длина заголовка сегмента (**Data Offset**) еще не исчерпана.

Опция типа 1 (**Нет операции**) также состоит из одного октета и может использоваться для выравнивания между опциями по границе 32 бит.

Опция типа 2 (**Максимальный размер сегмента**) состоит из четырех октетов: октета **Тип опции**, октета длины, значение которого равно четырем, и двух октетов, содержащих максимальный размер сегмента, который способен получать TCP-модуль, отправивший сегмент с данной опцией. Эта опция обычно используется в сегментах с установленным битом **SYN** на этапе установки соединения;

Другие опции используются для реализации дополнительных возможностей TCP-протокола.

5.2 Задание

Запустив анализатор пакетов, выполнить команду, полученную от преподавателя.

1. Указать, какие TCP-соединения были установлены при выполнении этой команды.

2. Для нескольких TCP-соединений описать процедуры установки и завершения, указать параметры соединения: ISN со стороны клиента и сервера, максимальный размер сегмента.

Лабораторная работа 6.

DNS

Цель работы: ознакомление с основами работы службы DNS, изучение правил описания зон DNS, приобретение практических навыков администрирования DNS-серверов.

6.1 Теоретические сведения

6.1.1 Зоны и сервера DNS

Система доменных имен (**Domain Name System, DNS**) обеспечивает соответствие между символьными (доменными) именами хостов сети и IP-адресами.

Доменная система имен образует иерархическую древовидную структуру. Дерево имен начинается с корня, который имеет пустое имя. Ниже расположены имена первого (или верхнего) уровня, например *com*, *org*, *ru*; еще ниже — имена второго уровня и т.д. Полное имя каждого узла состоит из последовательности имен узлов от данного узла до корня; имена разделяются точками.

Множество имен узлов дерева, расположенных ниже некоторого узла, образует **домен имен**.

Функционирование службы DNS обеспечивается системой серверов DNS. Каждый сервер хранит информацию о сопоставлениях «доменное имя — IP-адрес» для некоторого подмножества имен домена. Это подмножество хостов образует **зону ответственности** данного DNS-сервера (часто ее называют просто **зоной DNS**).

Зона может совпадать с доменом; тогда один сервер DNS хранит отображения «доменное имя — IP-адрес» для домена и всех его поддоменов. Однако если в домен входит большое количество хостов и ряд поддоменов, то такое решение нецелесообразно.

Чаще применяется подход, когда в большинстве поддоменов создаются свои DNS-сервера, которые обеспечивают соответствие между доменными именами и IP-адресами для хостов, принадлежащих поддоменам. Тогда зоной ответственности DNS-сервера домена является только то подмножество узлов домена, информация о которых находится непосредственно на этом сервере. Такая передача части функций одного DNS-сервера другому называется **делегированием**. При этом на DNS-сервере домена хранится также информация о DNS-серверах, отвечающих за его поддомены.

Таким образом, DNS-сервера также образуют иерархическую структуру. На вершине этой иерархии находятся DNS-сервера корневого домена (или корневые DNS-сервера). Они содержат информацию о DNS-серверах, отвечающих за домены верхнего уровня (Top Level Domain, TLD). Домены верхнего уровня хранят информацию об узлах и доменах второго уровня, и т.д.

В каждой зоне DNS имеется один главный (**primary**, **master**) DNS-сервер. Он хранит главную копию **описания зоны**, то есть, информацию о соответствии доменных имен IP-адресам для узлов данной зоны. Главную копию можно редактировать.

Кроме главного, в зоне могут существовать также один или несколько вторичных (**slave**, **secondary**) DNS-серверов. Они хранят резервные копии описания зоны, которые периодически получают с главного сервера. Изменять описание зоны на вторичном DNS-сервере нельзя. Отвечать на запросы об IP-адресах хостов, принадлежащих зоне, могут как главный, так и вторичные сервера DNS. Вторичные сервера нужны для снижения нагрузки на главный DNS-сервер, и для сохранения работоспособности системы в случае отказа главного сервера.

6.1.2 Описание зоны DNS

Описание зоны DNS содержит, главным образом, записи описания ресурса (Resource Record, RR). Кроме того, в нем могут присутствовать директивы и комментарии (обычно начинаются со знака «;»).

Директивы состоят из имени, которое начинается со знака «$»,

и аргумента. Несколько наиболее важных директив будут упомянуты ниже.

Доменные имена, встречающиеся в описаниях зоны, считаются **полными**, если они оканчиваются точкой. В противном случае они считаются **неполными** и расширяются именем текущей зоны DNS, либо именем, заданным директивой $ORIGIN.

Записи RR имеют следующий формат (здесь и далее необязательные аргументы заключаются в квадратные скобки):

```
name    [ttl]    [class]    type    data
```

- name — имя хоста, домена или поддомена.
- ttl — время жизни записи в кэше локального DNS-сервера или клиента DNS в секундах. Если ttl не указано, то время жизни записи определяется параметром minimum записи SOA (см. ниже), либо аргументом директивы $TTL.
- class — класс записи RR. Для IP-сетей значение этого поля всегда IN.
- type — тип записи. Наиболее распространенными типами записей являются SOA, NS, A, MX, PTR. Описания этих типов приведены ниже.
- data — данные. Это поле содержит информацию, зависящую от типа записи.

Запись SOA

SOA (Start Of Authority) — начальная запись типа RR в файле описания зоны (в каждом файле может быть только одна такая запись), содержит управляющую информацию о зоне в целом. Поле data записи SOA имеет формат:

```
origin    contact    (serial refresh retry expire minimum)
```

Смысл параметров:

- origin — доменное имя DNS-сервера, отвечающего за данную зону.
- contact — e-mail адрес администратора, отвечающего за данную зону (при этом символ «@» в адресе заменяется на точку).

- serial — серийный номер файла описания зоны. Действует правило: копирование информации о зоне с главного DNS-сервера на вторичный выполняется только если *серийный номер копии описания зоны на вторичном сервере меньше, чем серийный номер описания зоны на главном сервере*. Поэтому при любом изменении информации о зоне нужно увеличивать серийный номер.
- refresh — время (в секундах), по истечении которого происходит обновление описания зон вторичными серверами.
- retry интервал времени (в секундах), после которого вторичный сервер должен повторить запрос к главному серверу, если предыдущий запрос был выполнен неуспешно.
- expire — время жизни (в секундах) описания зоны на вторичном сервере, если попытка получить обновленную информацию о зоне с главного сервера завершилась неудачно.
- minimum — время жизни (в секундах) записей из данной зоны в кэше локальных DNS-серверов или DNS-клиентов по умолчанию. В настоящее время клиенты DNS интерпретируют этот параметр как время жизни негативных ответов (то есть, ответов на запрос несуществующего доменного имени). Время жизни позитивных ответов определяется аргументом директивы $TTL.

Запись NS

Запись NS (Name Server) определяет сервер имен для зоны. Таких записей в описании зоны может быть несколько. Формат записи NS:

```
[domain] [ttl]      [IN]      NS      [server]
```

Поле domain содержит имя зоны, для которой указывается DNS-сервер. Если это поле пустое, то предполагается, что указанный сервер отвечает за текущую зону DNS. Если указано имя поддомена, то запись определяет имя DNS-сервера, отвечающего за данный поддомен (делегирование полномочий).

Поле server определяет доменное имя или IP-адрес сервера, ответственного за зону, указанную в поле domain.

Обычно в описании зоны имеется несколько записей NS, которые определяют все сервера, отвечающие за данную зону и сервера, которым делегировано управление поддоменами.

Запись A

Запись A (Address) имеет формат

[host] [ttl] [IN] A address

и определяет соответствие доменного имени хоста, указанного в поле **host**, IP-адресу, указанному в поле **address**.

Если поле **host** пустое, считается, что оно такое же, как у предыдущей записи. Таким образом обычно указывают несколько IP-адресов для одного доменного имени.

Запись MX

MX (Mail Exchanger) — запись, определяющая сервера электронной почты для данной зоны. Формат этой записи следующий:

[name] [ttl] [IN] MX preference host

Поле **name** определяет, для какой зоны указывается сервер электронной почты, а поле **host** — доменное имя или IP-адрес этого сервера. Если поле **name** пустое, то предполагается, что это имя текущей зоны.

Наличие в описании зоны *example.org* **MX**-записи означает, что вся электронная почта, отправляемая на адреса вида *user@example.org* будет доставляться на сервер, указанный в поле **host** этой записи Подчеркнем, что это правило не распространяется на почту, адресованную на хосты данной зоны. То есть, сообщение, отправленное на *user@host.example.org*, будет доставлено на хост *host.example.org*[1].

Записей **MX** может быть несколько; в этом случае для каждой из них в поле **preference** указывается число, определяющее приоритет сервера. Чем меньше это число, тем выше приоритет. Таким образом можно определить один или несколько резервных серверов электронной почты, на которые будет доставляться почта, если основной сервер (имеющий наибольший приоритет) недоступен.

В поле **name** можно также указать конкретный хост, и тем самым перенаправить электронную почту, адресованную на этот хост, на другой сервер.

Кроме вышеописанных, существуют другие типы записей **RR**, например **HINFO**, содержащая информацию о хосте, или **CNAME**, которая определяет синонимы для данного доменного имени. Их описание можно найти в литературе, посвященой системе DNS.

[1]Если не существует **MX**-записи для хоста *host.example.org* (см. ниже).

6.1.3 Описание обратной зоны DNS

Для осуществления поиска доменного имени хоста по IP-адресу существует специальный домен *in-addr.arpa*. В этом домене IP-сетям соответствуют имена зон, которые совпадают с номерами сетей в обратном порядке. Например, сети класса C 212.192.16.0 будет соответствовать зона *16.192.212.in-addr.arpa*. DNS-сервер, отвечающий за эту зону, будет хранить информацию о доменных именах, соответствующих IP-адресам из данной сети. Такие зоны DNS принято называть **обратными зонами DNS**.

В описание обратной зоны DNS могут входить только записи SOA, NS и PTR.

Запись PTR

Запись типа PTR (Pointer) устанавливает соответствие между IP-адресом и доменным именем. Формат записи PTR:

```
address  [ttl]  [IN]  PTR  domain
```

Поле address задает IP-адрес, которому соответствует доменное имя, указанное в поле domain. Обычно в поле address указывается только десятичное значение последнего байта IP-адреса. Оно дополняется до имени домена, соответствующего сети, в которую входит данный адрес, например *16.192.212.in-addr.arpa*. В поле domain обязательно нужно указывать полное доменное имя.

6.2 Задание

Для выполнения данной работы сеть должна иметь структуру, изображенную на рис. 6.1. Сетям 1 и 2 присваиваются адреса класса C. В каждой сети существует DNS-сервер: сервер 1 в сети 1 и сервер 2 в сети 2. На всех компьютерах сети 1 в качестве предпочитаемого DNS-сервера в настройках TCP/IP должен быть указан сервер 1, на компьютерах сети 2 — сервер 2.

1. Присвоить компьютерам сетей 1 и 2 доменные имена, принадлежащие указанным в задании зонам DNS.

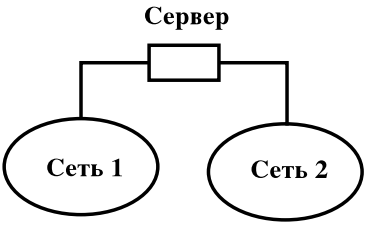

Рис. 6.1: Структура сети для выполнения лабораторной работы по настройке DNS.

2. Создать на DNS-серверах 1 и 2 указанные зоны и добавить в их описания информацию о входящих в них компьютерах. Описание настройки системы DNS в Windows 2003 Server см. в Приложении Е..
3. Проверить работу системы DNS данной сети. Система функционирует правильно, если:
   - между любыми двумя компьютерами, принадлежащими одной зоне DNS, проходит *ping* по доменному имени компьютера;
   - с компьютера, принадлежащего некоторому домену, проходит *ping* по доменному имени до компьютеров из поддоменов данного домена;
   - утилита *nslookup* возвращает IP-адрес по доменному имени и доменное имя по IP-адресу для любого компьютера из той же зоны DNS (описание утилиты *nslookup* см. в Приложении Е.).

Вариант 1. Сети 1 и 2 образуют зону *kerait.mo*. Сервер 1 — главный DNS-сервер для этой зоны и обратных зон DNS для сетей 1 и 2. Сервер 2 — резервный DNS-сервер для этих же зон.

Вариант 2. Сеть 1 входит в зону *tele.local*, сеть 2 — в зону *dabo.tele.local*. Сервер 1 — главный DNS-сервер для зоны *tele.local* и обратной зоны DNS для сети 1. Сервер 2 — главный DNS-сервер для зоны *dabo.tele.local* и обратной зоны DNS для сети 2.

Вариант 3. Сеть 2 входит в зону *niichavo.su*, сеть 1 — в зону *ols.niichavo.su*. Сервер 2 — главный DNS-сервер для зоны *niichavo.su* и обратной зоны DNS для сети 2. Сервер 1 — главный DNS-сервер для зоны *ols.niichavo.su* и обратной зоны DNS для сети 1.

Вариант 4. Сети 1 и 2 образуют зону *togon.local*. Сервер 2 — главный DNS-сервер для этой зоны и обратных зон DNS для сетей 1 и 2. Сервер 1 — резервный DNS-сервер для этих же зон.

Вариант 5. Сети 1 и 2 образуют зону *silla.kr*. Сервер 1 — главный DNS-сервер для этой зоны и обратных зон DNS для сетей 1 и 2. Сервер 2 — резервный DNS-сервер для этих же зон.

Вариант 6. Сеть 1 входит в зону *nushibi.ztk*, сеть 2 — в зону *chumi.nushibi.ztk*. Сервер 1 — главный DNS-сервер для зоны *nushibi.ztk* и обратной зоны DNS для сети 1. Сервер 2 — главный DNS-сервер для зоны *chumi.nushibi.ztk* и обратной зоны DNS для сети 2.

Вариант 7. Сеть 2 входит в зону *tardush.tu*, сеть 1 — в зону *tokuz-oguz.tardush.tu*. Сервер 2 — главный DNS-сервер для зоны *tardush.tu* и обратной зоны DNS для сети 2. Сервер 1 — главный DNS-сервер для зоны *tokuz-oguz.tardush.tu* и обратной зоны DNS для сети 1.

Вариант 8. Сети 1 и 2 образуют зону *dulu.ztk*. Сервер 2 — главный DNS-сервер для этой зоны и обратных зон DNS для сетей 1 и 2. Сервер 1 — резервный DNS-сервер для этих же зон.

Вариант 9. Сети 1 и 2 образуют зону *tabgach.local*. Сервер 1 — главный DNS-сервер для этой зоны и обратных зон DNS для сетей 1 и 2. Сервер 2 — резервный DNS-сервер для этих же зон.

Вариант 10. Сеть 1 входит в зону *tolis.ztk*, сеть 2 — в зону *buri.tolis.ztk*. Сервер 1 — главный DNS-сервер для зоны *tolis.ztk* и обратной зоны DNS для сети 1. Сервер 2 — главный DNS-сервер для зоны *buri.tolis.ztk* и обратной зоны DNS для сети 2.

Лабораторная работа 7.

Электронная почта

Цель работы: ознакомление с работой электронной почты в сети Internet, изучение работы почтового транспортного агента Sendmail, получение навыков работы с системным журналом Sendmail.

7.1 Теоретические сведения

7.1.1 Организация электронной почты в Internet

Электронная почта — это служба, позволяющая доставлять сообщения от одного узла сети к другому.

Для адресации сообщений электронной почты используются **email-адреса**, состоящие из имени почтового ящика и имени почтового домена, разделенных значком «@». Имя почтового ящика обычно совпадает с регистрационным именем пользователя на почтовом сервере, обслуживающем данный почтовый домен. Например, пользователь почтового сервера *valeo.tmsu.ru* с именем *filipok* будет иметь email-адрес *filipok@valeo.tmsu.ru*.

Работа электронной почты обеспечивается системой **почтовых серверов**. На почтовом сервере находятся почтовые ящики пользователей данного сервера, а также работают следующие программы:

- **транспортный агент** (Mail Transfer Agent, MTA), принимающий входящие сообщения и определяющий способ их доставки;

- **агент доставки** (Mail Delivery Agent, MDA), осуществляющий доставку сообщений;
- **сервер входящей почты**, служащий для доставки почты с почтового сервера на компьютер пользователя и использующий протоколы **POP** (Post Office Protocol) или **IMAP** (Internet Mail Access Protocol).

Для работы с элетронной почтой на компьютере пользователя должен быть установлен **почтовый клиент** (например, Outlook Express, The Bat! и т.д.). Для корректной работы в настройках почтового клиента нужно указать, как минимум, следующие данные:

- email-адрес пользователя;
- доменное имя или IP-адрес SMTP-сервера (сервера исходящей почты);
- протокол, используемый для работы с сервером входящей почты;
- доменное имя или IP-адрес сервера входящей почты;
- регистрационное имя пользователя на сервере входящей почты.

В Outlook Express по умолчанию имеются следующие почтовые папки: **Входящие** — для вновь поступающих сообщений, **Отправленные** — для отправленных сообщений, **Исходящие** — для сообщений, приготовленных для отправки, но еще не отправленных, **Удаленные** — для удаленных сообщений. Пользователь также имеет возможность создавать свои почтовые папки. При использовании протокола POP3, эти папки хранятся на компьютере пользователя. При использовании протокола IMAP4, папки могут размещаться как на почтовом сервере, так и на клиентском компьютере.

7.1.2 Работа почтового транспортного агента

В данном разделе рассматриваются принципы работы транспортного агента на примере MTA Sendmail.

MTA работает как SMTP-сервер и ожидает запросы клиентов на порту 25/TCP. В роли SMTP-клиентов выступают пользовательские почтовые программы, либо агенты доставки на удаленные сервера.

Согласно протоколу SMTP, клиент сначала передает серверу email-адреса отправителя и получателя, а затем само сообщение. Если

на каком-то этапе SMTP-диалога возникает ошибка, сервер возвращает сообщение об ошибке. Если сообщение успешно принято, MTA сохраняет его во временном файле и присваивает ему алфавитно-цифровой иденитификатор (Queue Identifier, **QID**), например, j3M7NMY5002806. Клиенту при этом возвращается QID и сообщение «*Message accepted for delivery*». QID также используется во всех записях системного журнала, связанных с данным письмом (см. ниже).

Если в процессе доставки письма возникают проблемы, MTA информирует об этом отправителя письма с помощью служебных почтовых сообщений. Обратный адрес таких сообщений *MAILER-DAEMON@<имя_сервера>*.

Для каждого сообщения MTA по адресу получателя определяет способ доставки. Различают два основных способа доставки:

- **локальная**, которая выполняется в случае, если доменное имя в email-адресе получателя совпадает с одним из доменных имен данного почтового сервера;
- **пересылка** (англ. **relaying**), которая выполняется, если сообщение адресовано на другой почтовый сервер.

При локальной доставке проверяется, существует ли на данном сервере почтовый ящик с именем, указаным в email-адресе получателя. В случае отрицательного результата клиенту возвращается сообщение об ошибке «*User unknown*». Если ящик существует, то сообщение передается агенту доставки **local**, который, как правило, просто помещает сообщение в почтовый ящик получателя и завершает работу. MTA по коду возврата определяет, было ли сообщение доставлено успешно или произошла ошибка. В случае ошибки MTA формирует служебное сообщение об этой ошибке и посылает его отправителю письма. Если ошибок не возникает, сообщение считается доставленным получателю. По умолчанию локальная доставка выполняется *всегда, вне зависимости от того, откуда получено данное сообщение*.

Пересылка сообщений на другой почтовый сервер разрешена только с некоторого множества хостов, указанных в настройках MTA. Если запрос на пересылку придет с компьютера, не входящего в это множество, MTA вернет ошибку «*Relaying denied*».

Если сообщение принято для пересылки, MTA определяет, на какой сервер нужно отправить данное сообщение. При этом в DNS сна-

чала производится поиск почтовых серверов, обслуживающих почтовый домен получателя, т.е., поиск MX-записей для этого домена. Если MX-записи не найдены, предполагается, что имя почтового сервера совпадает с именем почтового домена, и производится поиск IP-адреса, соответствующего этому имени.

Затем MTA передает адрес сервера и сообщение агенту доставки **smtp**, который пытается переслать сообщение на выбранный сервер. Когда агент доставки завершает работу, MTA по коду возврата определяет, успешно ли произведена доставка. В случае фатальной ошибки (например, удаленный сервер возвращает ошибку «*User unknown*») отправителю посылается служебное сообщение о невозможности доставить письмо. Если возникла нефатальная ошибка (например, сервер не отвечает), сообщение помещается в очередь недоставленных сообщений. Через определенные промежутки времени (обычно 30–60 мин) MTA осуществляет повторный вызов агента доставки для новой попытки доставить сообщение. Если сообщение не удается доставить в течение определенного срока (по умолчанию, 4 часа) MTA посылает пользователю предупреждающее сообщение. При этом сообщение сохраняется в очереди и попытки его доставить будут продолжаться.

Максимальный срок хранения писем в очереди недоставленных сообщений по умолчанию составляет 5 дней. Если в течение этого срока сообщение не удается отправить, оно уничтожается и MTA посылает отправителю сообщение об ошибке.

7.2 Системный журнал MTA Sendmail

Информацию обо всех своих действиях Sendmail записывает в **системный журнал**. В Debian Linux этот журнал находится в файле */var/log/mail.log*.

Каждая запись в системном журнале состоит из **времени записи**, **имени компьютера**, **имени и идентификатора программы**, сгенерировавшей запись, **идентификатора почтового сообщения (QID)**, с которым связана запись и сообщения. Сообщения состоят из последовательности полей в формате **имя=значение**.

Существует два главных типа записей. Запись первого типа добавляется в журнал при получении почтового сообщения; для каждого сообщения имеется в точности одна такая запись. Она содержит следующие поля:

from — адрес отправителя;

size — размер сообщения в байтах;

class — класс (числовой приоритет) сообщения;

pri — начальный приоритет сообщения;

nrcpts — число получателей данного сообщения;

msgid — идентификатор сообщения (из заголовка письма);

proto — протокол, использовавшийся при получении письма;

daemon — имя почтового сервера (берется из настроек Sendmail);

relay — доменное имя или IP-адрес компьютера, с котрого получено сообщение.

Записи второго типа заносятся в журнал при каждой попытке доставить сообщение. Запись включает следующие поля:

to — список получателей сообщения, разделенных запятыми;

ctladdr — имя и идентификаторы пользователя, от имени которого осуществляется доставка;

delay — полная задержка между временем получения сообщения и данной попыткой его доставить;

xdelay — время, потраченное на данную попытку достаавки сообщения;

mailer — имя MDA, осуществлявшего доставку;

relay — имя компьютера, принявшего или отклонившего сообщение;

dsn — код ошибки согласно RFC 2034;

stat — статус попытки доставки сообщения.

Не все поля обязательно присутствуют в записях; например relay обычно не указывается при локальной доставке.

7.3 Задание

Для выполнения данной работы сеть должна состоять из двух подсетей, в каждой из которых существует почтовый сервер, обслуживающий данную подсеть. Студенты должны иметь удаленный доступ через на эти сервера с помощью ssh-клиента *putty*.

1. С помощью почтового клиента Outlook Express отправить письма на указанные преподавателем адреса.

2. В системных журналах почтовых серверов найти записи, соответствующие этим письмам, проанализировать их и описать, как проходила доставка каждого сообщения.

Лабораторная работа 8.

Формат сообщения электронной почты

Цель работы: изучение формата почтового сообщения,

8.1 Теоретические сведения

Формат почтового сообщения определен в документе RFC-822. Почтовое сообщение состоит из **конверта**, **заголовока** и **тела сообщения**. Конверт используется только программами доставки, поэтому здесь не рассматривается.

Заголовок содержит служебную информацию о сообщении. Он всегда находится перед телом сообщения и отделяется от него пустой строкой. Тело сообщения — это собственно сообщение.

8.1.1 Формат заголовка почтового сообщения

Согласно RFC-822, заголовок состоит из полей. Поля состоят из имени поля и значения поля, разделенных символом двоеточия. Имена и значения полей заголовока имеют текстовый формат. Ниже перечислены наиболее важные поля.

Date — дата отправки почтового сообщения в общепринятом для Internet формате.

From — адрес отправителя сообщения.

To — адрес получателя сообщения.

Subject — тема письма; указывается отправителем при подготовке сообщения.

Reply-To (или Return-Path) — email-адрес, который нужно использовать при ответе на данное сообщение. Этот адрес может не совпадать с указанным в поле From.

Message-ID — уникальный идентификатор сообщения, который присваивается ему почтовым клиентом при отправке.

In-Reply-To — идентификатор сообщения, в ответ на которое отправлено данное сообщение.

References — ссылки; в этом поле перечисляются идентификаторы всех сообщений, связанных с данным сообщением.

Received — информация о том, откуда, кем, для кого и когда получено сообщение. *Каждый* SMTP-сервер, принимавший участие в его доставке сообщения, добавляет к его заголовку одно поле Received, поэтому таких полей в заголовке обычно несколько, и по ним можно определить путь письма.

Формат поля Received поясним на примере:

```
Received: from lab1123.spti.tsu.ru ([212.192.127.156])
    by phys.tsu.ru (8.12.8p1/8.12.8)
    with ESMTP id i8N7tLHL001052
    for <pecher@cmm.univer.omsk.su>;
    Thu, 23 Sep 2004 14:55:28 +0700 (NOVST)
```

Поле содержит информацию о том, откуда (from), кем (by), для кого (for) и когда получено сообщение. В данном случае его принял сервер *phys.tsu.ru* от компьютера *lab1123.spti.tsu.ru* (в скобках указан его IP-адрес) для *pecher@cmm.univer.omsk.su* 23 сентября 2004 года в 14:55 по новосибирскому (NOVST) времени. Указаны также идентификатор, присвоенный сообщению на сервере *phys.tsu.ru* (i8N7tLHL001052) и версия транспортного агента *Sendmail*, установленного на этом сервере (8.12.8p1/8.12.8)[1].

Отметим также, что поля Received в заголовке следуют в обратном порядке, то есть, поле Received, добавленное последним SMTP-сервером, окажется первым полем этого типа в заголовке сообщения.

[1]Sendmail ранее являлся практически единственным транспортным агентом, поэтому он указывает в поле Received только свою версию. Другие транспортные агенты указывают также свое название, например, Postfix, Qmail и др.

RFC-822 допускает использование полей, которые не определены в стандарте, но содержат дополнительные данные или служебную информацию о сообщении. Их имена должны начинаться с префикса «X-». Чаще всего встречается поле X-Mailer (или X-Sender), содержащее название почтового клиента, с помошью которого отправлено данное письмо. Если письмо на некотором почтовом сервере было проверено на вирусы, то к заголовку добавляется поле X-AntiVirus (или X-Virus-Scanned), в котором указывается название антивируса:

```
X-AntiVirus: Checked by Dr.Web [version: 4.32b, ...]
```

Программы фильтрации спама добавляют в заголовок поля, начинающиеся с префикса «X-Spam»; например, поле X-Spam-Checker-Version содержит название и версию спам-фильтра.

Подчеркнем, что заголовок сообщения содержит информацию *только о процессе его доставки на почтовый сервер получателя*. При доставке сообщения с почтового сервера на клиентский компьютер по протоколам POP или IMAP дополнительные поля в заголовок добавлятся не будут.

8.1.2 Стандарт MIME

Электронная почта изначально была ориентирована на доставку текстовых сообщений. Двоичные файлы (например, исполнимые файлы, документы в формате Word, графику и др.), пересылаемые по электронной почте, нужно специальным образом кодировать, чтобы избежать искажения информации.

Способы кодирования двоичных данных и формат почтового сообщения, содержащего вложенные файлы, определяет стандарт **MIME** (Multipurpose Internet Mail Encoding). В настоящее время все распространенные почтовые клиенты поддерживают этот стандарт. Пользователь для пересылки по электронной почте двоичного файла указывает, какой файл он хочет отправить. Почтовый клиент определяет тип содержащихся в нем данных, кодирует их и нужным образом вставляет в тело сообщения, а также добавляет в заголовок ряд полей, содержащих информацию о вложенном файле. Эту информацию почтовый клиент получателя использует для корректного извлечения двоичного файла из сообщения.

Согласно MIME, существует пять основных типов данных[2]: **text**

[2]Есть также два специальных типа, которые будут описаны ниже.

— текст; **image** — графический файл; **audio** — звуковой файл; **video** — видеофильм; **application** — приложение (программа) или файл в формате, используемом каким-либо приложением.

Каждый тип делится на подтипы, например, **text/plain** — простой текст, **image/gif** — графический файл в формате GIF, **application/msword** — документ Microsoft Word и др. Список наиболее распространенных типов и подтипов MIME приведен в приложении. Для некоторых подтипов можно указывать дополнительные параметры (примеры будут приведены ниже).

Стандарт MIME определяет также ряд способов кодирования информации. Наиболее распространенными являются:

- **8bit** или **7bit** — информация передается «как есть», то есть, как набор 8- или 7-битных символов. Такой способ кодирования используется для текстовых сообщений.
- **quoted-printable** — символы исходного сообщения с кодами от 32 до 127 передаются как есть, все остальные символы заменяются на последовательность, состоящую из знака «=», за которым следует шестнадцатиричный код символа (например, `=D0=D2=CF=C5=CB=D4`). Применяется, в основном, для кодирования текстовых сообщений, содержащих символы национальных алфавитов (в частности, русского).
- **base64** — каждые 3 байта исходного сообщения преобразуются в 4 6-битовые группы, каждая группа интерпретируется как целое число от 0 до 63 и заменяется символом, соответствующим этому числу в кодировочной таблице. Эта таблица содержит прописные и строчные буквы латинского алфавита, цифры и символы "+"и "/", поэтому кодированное сообщение выглядит примерно так: `7cnOydPU0iAtIMvP2sXMCg`. Base64 — основной метод кодирования двоичной информации, рекомендуемый MIME.

Стандарт MIME определяет несколько дополнительных полей заголовка почтового сообщения, которые описывают структуру тела сообщения и типы содержащихся в нем данных (в отличие от полей стандарта RFC-822, которые содержат информацию о механизме доставки письма).

MIME-Version — версия MIME, используемая в данном письме.

Таблица 8.1: Наиболее распространенные типы и подтипы MIME

Тип	**Подтип**	**Описание**
text	plain	Простой текст
text	html	Текст в формате HTML (который применяется для создания Web-страниц)
image	gif	Графический файл в формате GIF
image	jpeg	Графический файл в формате JPEG
image	bmp	Графический файл в формате BMP
audio	mpeg	Аудио-файл в MPEG-совместимом формате (например, в MP3)
video	mpeg	Видео-файл в MPEG-совместимом формате (например, в MPG)
application	octet-stream	Произвольный двоичный файл, чаще всего исполнимый
application	msword	Документ в формате Microsoft Word
application	vnd.ms-excel	Электронная таблица в формате Microsoft Excel
application	vnd.ms-powerpoint	Электронная презентация в формате Microsoft Powerpoint
application	zip	Архив в формате ZIP
application	pdf	Документ в формате PDF (Portable Document Format)

Тип	Подтип	Описание
multipart	mixed	Документ, состоящий из нескольких независимых фрагментов
multipart	alternative	Документ, состоящий из нескольких альтернативных фрагментов
multipart	related	Документ, состоящий из нескольких связанных между собой фрагментов
message	rfc822	Почтовое сообщение с заголовком в формате RFC-822
message	partial	Почтовое сообщение большого размера, пересылаемое по частям (поддерживается не всеми почтовыми клиентами).

Content-Type — определяет тип и подтип MIME данных, содержащихся в теле сообщения.

Content-Transfer-Encoding — транспортная кодировка, т.е., способ кодирования данных, содержащихся в теле сообщения.

Существуют также необязательные поля, например Content-ID — идентификатор содержания письма, Content-Description — комментарий к содержимому и др.

Согласно стандарту MIME, почтовое сообщение может состоять из нескольких фрагментов, содержащих данные различных типов и подтипов MIME. Для описания таких сообщений определен специальный тип **multipart**. У него имеется обязательный параметр boundary, определяющий **граничную строку**, которая используется как разделитель отдельных частей письма.

Составное сообщение начинается с **общего заголовка**, который состоит из полей стандартов RFC-822 и MIME и содержит информацию о письме в целом. В поле Content-Type общего заголовка указывается тип multipart, и, с помощью параметра boundary, определяется граничная строка.

Каждый фрагмент начинается с граничной строки, за которой следует заголовок данного фрагмента. Он содержит только поля стандарта MIME и определяет тип и подтип MIME, транспортную кодировку и другие характеристики тела фрагмента. Содержимое фрагмента отделяется от его заголовка пустой строкой.

Признаком конца составного сообщения является граничная строка, к которой присоединены два символа «–».

У MIME-типа multipart существует несколько подтипов. Если тело сообщения состоит из нескольких независимых фрагментов (например, к текстовому сообщению прикреплены несколько файлов), то оно принадлежит к подтипу **mixed**.

В некоторых случаях сообщение состоит из нескольких взаимосвязанных частей, например, HTML-документа и нескольких рисунков, на которые в документе имеются ссылки. Такое сообщение относится к подтипу **related**. Каждому фрагменту сообщения при этом присваивается идентификатор, указанный в поле Content-ID. Он используется для ссылок на данный фрагмент из других частей сообщения.

Другим распространенным подтипом MIME-типа multipart является **alternative**. Сообщение, относящееся к этому подтипу, обычно состоит из двух фрагментов, одинаковых по содержанию, но различающихся визуальным оформлением. Например, один из них может быть простым текстом, а другой — тем же текстом, но в формате HTML. Если почтовый клиент получателя поддерживает HTML, то он отображает второй фрагмент, если нет — то первый.

Иногда требуется вложить одно почтовое сообщение в другое. Для таких случаев существует еще один специальный MIME-тип — **message**. Например, если один из фрагментов представляет собой полное почтовое сообщение со своими заголовком и телом, то он будет относится к типу **message/rfc822**.

8.1.3 Примеры анализа заголовков

Пример 1. Рассмотрим следующий заголовок сообщения.

```
From sharikov@sptu-49.udoev.ru Sat Nov  4 10:34:10 2000
Return-Path: <sharikov@sptu-49.udoev.ru>
Received: from sptu-49.udoev.ru  ([214.22.231.1])
    by valeo.tmsu.ru with SMTP (MDaemon.v2.7.SP5.R)
    for <filipok@valeo.tmsu.ru>;
    Sat, 4 Nov 2000 10:33:39 +0300 (MSD)
```

```
Received: from teacher ([214.22.231.91])
    by sptu-49.udoev.ru (8.9.3/8.9.3)
    with SMTP id i4MCBwTj000253
    for <filipok@valeo.tmsu.ru>;
    Sat, 4 Nov 2000 10:30:11 +0300 (MSD)
Message-ID: <801be7049$3558ad00$2380a8c0@sptu-49.udoev.ru>
From: "Vladimir Sharikov" <sharikov@sptu-49.udoev.ru>
To: Vladimir Filipok <filipok@valeo.tmsu.ru>
Subject: Re: Предложения по работе
In-Reply-To: <01C045AD.800A6EC0.filipok@valeo.tmsu.ru>
Date: Sat Nov  4 10:29:46 2000 +0300
MIME-Version: 1.0
Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: quoted-printable
X-Mailer: Microsoft Outlook Express 5.00.2919.6600

=EB=EF=ED=F0=F8=E0=F4=E5=F2=EE=F9=E5
=EF=E2=F5=FE=E1=E0=FD=E9=E5
=F3=E9=F3=F4=E5=ED=F9
```

Из него можно извлечь следующую информацию. Адрес отправителя данного сообщения — Vladimir Sharikov <*sharikov@sptu-49.udoev.ru*> (поля From и Return-Path); адрес получателя — Vladimir Filipok <*filipok@valeo.tmsu.ru*> (поле To), идентификатор — <801be70493558*ad*002380a8c0@sptu-49.udoev.ru> (поле Message-ID). Сообщение отправлено 4 ноября 2000 г в 10:29 (поле Date). Оно является ответом на некоторое сообщение (т.к. имеется поле In-Reply-To). Тема сообщения — «Re: Предложения по работе» (поле Subject), ключевое слово "Re:"в начале темы также указывает на то, что данное сообщение является ответом. Отправитель пользовался почтовой программой Microsoft Outlook Express версии 5.0 (поле X-Mailer).

Заголовок содержит два поля Received, следовательно, данное сообщение проходило через два почтовых сервера. Из второго поля Received следует, что сначала оно было принято сервером *sptu-49.udoev.ru*, IP-адрес которого 214.22.231.1, с клиентского компьютера *teacher*, IP-адрес которого 214.22.231.91. Далее это сообщение было принято сервером *valeo.tmsu.ru* (первое поле Received). Этот сервер является почтовым сервером получателя, поэтому доставка сообщения на этом заканчивается.

Содержание данного письма соответствует стандарту MIME версии 1.0 (поле MIME-Version). Данные, содержащиеся в теле письма представляют собой простой текст (поле Content-Type) в кодировке KOI8-R (параметр charset). Транспортная кодировка письма — quoted-printable (поле Content-Transfer-Encoding).

Отметим, наконец, что первая строчка в заголовке, начинающаяся со слова From, — это так называемый UUCP-путь сообщения. Протокол UUCP (Unix-Unix Copy) ранее применялся для доставки сообщений между хостами, соединенными коммутируемыми линиями связи. В настоящее время он практически не используется, поэтому указание UUCP-пути потеряло смысл, однако заголовки сообщений всегда начинаются с данной строки. Фактически, она служит признаком начала сообщения.

Пример 2. Рассмотрим сообщение, к которому прикреплен файл. В данном примере рассматриваем только поля стандарта MIME. Исходный текст этого письма будет выглядеть следующим образом (в заголовке письма опущены поля Received, тело 2-го фрагмента приведено не полностью):

```
From filipok@valeo.tmsu.ru  Thu May  15 18:44:44 2003
Return-Path: <filipok@valeo.tmsu.ru>
...
Message-ID: <000901c55170$354309a0$bf23c0d4@valeo.tmsu.ru>
From: Vladimir Filipok <filipok@valeo.tmsu.ru>
To: "Vladimir Sharikov" <sharikov@sptu-49.udoev.ru>
Subject: О реформе образования
Date: Thu, 15 May 2003 18:44:40 +0400
MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="----=_NextPart_000_0005_01C551A2.7F1D36E0"
X-Mailer: Microsoft Outlook Express 5.00.2919.6600

This is a multi-part message in MIME format.

------=_NextPart_000_0005_01C551A2.7F1D36E0
Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: 8bit

Это соображения по реформе образования.
Посмотри, пожалуйста.
```

```
С уважением,
Владимир Филипок

------=_NextPart_000_0005_01C551A2.7F1D36E0
Content-Type: application/msword;
                name="Реформа образования.doc"
Content-Transfer-Encoding: base64

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAA
EAAALAAAAAEAAAD+////AAAAACkAAAD/////////////////////////
...
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA==

------=_NextPart_000_0005_01C551A2.7F1D36E0--
```

Данное сообщение состоит из нескольких независимых фрагментов, так как его содержимое принадлежит к типу multipart/mixed (поле Content-Type общего заголовка). Граничная строка имеет вид `----=_NextPart_000_0005_01C551A2.7F1D36E0` (параметр boundary).

Данное сообщение состоит из двух фрагментов[3]. Первый фрагмент является текстовым сообщением в кодировке koi8-r (поле Content-Type), содержащимся в теле письма «как есть» (поле Content-Transfer-Encoding).

Следующий фрагмент представляет собой документ в формате Microsoft Word (поле Content-Type), имя которого *Реформа образования.doc* (параметр name). Транспортная кодировка данного фрагмента — base64 (поле Content-Transfer-Encoding). Этот фрагмент является последним, так как за ним следует признак конца составного сообщения — граничная строка, к которой присоединены два символа «минус».

[3]Строка «This is a multi-part message in MIME format.» вставляется в составные сообщения по историческим причинам и почтовыми клиентами не отображается.

8.2 Задание

1. Проанализировать предложенные преподавателем почтовые сообщения и составить их описание, по следующему шаблону:

   - email-адрес отправителя, email-адрес получателя, идетификатор, и дата отправки сообщения;
   - путь письма (доменные имена и IP-адреса клиентского компьютера, с которого было отправлено сообщение, и всех почтовых серверов, через которые оно проходило);
   - информация, содержащаяся в дополнительных полях (начинающихся с префикса «X-»);
   - информация о содержании сообщения (из полей стандарта MIME);
   - если сообщение состоит из нескольких фрагментов, то описать его структуру и содержание каждого фрагмента.

Лабораторная работа 9.

Сети Unix на основе NFS и NIS

Цель работы: ознакомление с работой сетей Unix, получение практических навыков конфигурирования служб NFS и NIS.

9.1 Теоретические сведения

9.1.1 Работа NFS и NIS

Сети Unix традиционно базируются на совместном использовании двух служб: **NFS** (**Network File System**) и **NIS** (**Network Information System**).

NFS позволяет произвольной группе пользователей совместно использовать общую файловую систему на сервере. NFS-сервер **экспортирует** (предоставляет для доступа удаленным клиентам) один или несколько каталогов. Клиенты получают к ним доступ путем монтирования в некоторую точку локального дерева каталогов.

NFS использует два протокола прикладного уровня. Первый управляет удаленным монтированием. Второй используется для операций над удаленными файлами и каталогами.

NIS обеспечивает централизованное хранение служебной информации в сети. В Unix служебная информация находится в **системных базах данных** (см. табл. 9.1). Обычно системные базы данных яв-

Таблица 9.1: Системные базы данных Unix

Имя базы	**Описание**
passwd	Информация о пользователях: регистрационное имя (логин), идентификатор (UID), идентификатор группы (GID), к которой принадлежит пользователь, имя домашнего каталога и др.
shadow	Шифрованные пароли пользователей.
group	Информация о группах пользователей.
hosts	Информация о соответствии имен компьютеров IP-адресам.
services	Информация о соответствии имен сетевых служб номерам портов.

ляются текстовыми файлами в каталоге */etc*, например, */etc/passwd*, */etc/shadow* и т.д.

При использовании NIS служебная информация хранится на серверах NIS в базах данных NIS. Совокупность компьютеров, использующих одну и ту же совокупность баз данных NIS, образует **домен NIS**. Внутри домена должен быть хотя бы один сервер, хранящий главную копию базы данных NIS (**master NIS-сервер**). Остальные сервера NIS являются резервными (slave) и загружают копию системной базы данных с master-сервера.

Каждой системной базе данных в NIS соответствует одна или несколько баз, называемых **NIS-картами**; они различаются параметрами, используемыми в качестве ключа базы. Например, NIS-карты *passwd.byname* и *passwd.byuid* содержат информацию из системной базы данных *passwd*, но в первой из них ключом служит логин пользователя, а во второй — UID пользователя. NIS-карты создаются при конфигурировании master-сервера NIS из локальных системных файлов.

Для корректной работы ОС Unix с NIS нужно указать системе, что для поиска служебной информации нужно обращаться к этой службе. В современных версиях Unix для этого используется технология Name Service Switch (NSS). Она позволяет указать для каждой системной базы данных одну или несколько служб, в которых будет

производится поиск информации из этой базы.

NFS и NIS используют технологию **удаленного вызова процедур** (Remote Procedure Call, RPC), поэтому в сети должен работать вспомогательный сервер *portmap*. При запуске серверные программы, обеспечивающие работу NFS и NIS, регистрируют в *portmap* свои имена и номера портов, которые они используют. Клиенты перед обращением к NFS или NIS, запрашивают номера портов соответствующих служб у *portmap*. Сервер *portmap* использует UDP-порт 111.

9.1.2 Настройка NFS

Настройка NFS-сервера На сервере NFS должны работать серверы *mountd* и *nfsd*, а также служба *portmap*. *Mountd* обслуживает запросы клиентов на удаленное монтирование экспортируемых каталогов. *Nfsd* обеспечивает доступ клиентов к файлам в этих каталогах.

В Debian Linux для управления службой NFS используются команды

- */etc/init.d/nfs-user-server start* — запуск NFS,
- */etc/init.d/nfs-user-server stop* — остановка NFS.

Список экспортируемых каталогов содержится в файле */etc/exports*. Записи в этом файле имеют следующий формат:

<catalog> <hosts>(<options>)

- *<catalog>* — имя экспортируемого каталога.
- *<hosts>* — список доменных имен хостов сети, которые могут монтировать данный каталог. Вместо доменных имен можно указывать IP-адреса хостов или сетей.
- *<options>* — список опций монтирования каталога для данного хоста. Наиболее важными опциями являются:
  - *ro* — каталог монтируется только для чтения;
  - *rw* — каталог монтируется для чтения и записи;
  - *root_squash* — запрет на доступ к файлам каталога с правами суперпользователя (рекомендуется для повышения безопасности NFS).

Для одного каталога можно указывать несколько пар *<hosts>(<options>)*.

После внесения изменений в файл */etc/exports* службу NFS необходимо остановить и запустить заново.

Для проверки работы NFS-сервера нужно:

1. Проверить, зарегистрированы ли в *portmap* серверы *mountd* и *nfsd* с помощью команды *rpcinfo -p*
2. Проверить список экспортируемых каталогов командой *showmount -e localhost*

Настройка NFS-клиента На стороне клиента поддержка NFS должна быть включена в ядро Linux (в Debian Linux она присутствует по умолчанию). Для монтирования удаленного каталога нужно выполнить команду:

mount -t nfs <server>:<catalog> <mount_point>

Аргументы имеют следующий смысл:

- *server* — доменное имя или IP-адрес сервера NFS;
- *catalog* — имя экспортируемого каталога на этом сервере (указанное в файле */etc/exports*);
- *mount_point* — точка монтирования удаленного каталога в локальное дерево каталогов.

9.1.3 Настройка NIS

В Debian Linux для управления службой NIS (как на клиенте, так и на сервере) используются команды

- */etc/init.d/nis start* — запуск NIS,
- */etc/init.d/nis stop* — остановка NIS.

Параметры службы NIS в целом содержатся в файле */etc/default/nis*. В частности, здесь указывается, является ли данный компьютер клиентом NIS, master-сервером или slave-сервером NIS.

Настройка NIS-сервера На сервере NIS должны быть запущены сервер *ypserv* и служба *portmap*.

Настройки сервера храняться в следующих файлах.

/etc/defaultdomain — содержит имя домена NIS, который обслуживает данный сервер.

/etc/ypserv.conf — содержит настройки NIS-сервера.

/etc/ypserv.securenets — содержит список сетей, с которых может принимать запросы данный NIS-сервер. Запросы с остальных сетей будут игнорироваться. Записи в этом файле имеют формат *<маска_сети> <адрес_сети>*

Для создания баз данных NIS используется команда
/usr/lib/yp/ypinit -m
Она создает все необходимые NIS-карты в каталоге */var/yp/<имя_домена>*.

После внесения изменений в конфигурационные файлы службу NIS необходимо остановить и запустить заново.

Если локальные системные файлы изменялись после создания NIS-карт (например, в систему был добавлен новый пользователь), то для обновления баз NIS нужно перейти в каталог */var/yp* и дать команду *make*.

Для проверки работы сервера нужно:

1. Проверить, зарегистрирован ли в *portmap* сервер *ypserv* с помощью команды *rpcinfo -p*
2. Проверить существование карт NIS на данном компьютере командой *ypcat -h localhost <имя_NIS_карты>*

Настройка клиента NIS Для настройки клиента NIS нужно сконфигурировать клиентскую часть NIS, а затем настроить службу NSS для поиска системной информации в NIS.

На клиентском компьютере должен быть запущен демон *ypbind*.

Настройки клиента храняться в следующих файлах:

/etc/defaultdomain — содержит имя домена NIS, к которому принадлежит данный компьютер.

/etc/yp.conf — содержит IP-адреса NIS-серверов, обслуживающих данный домен.

После внесения изменений в конфигурационные файлы службу NIS необходимо остановить и запустить заново. При этом *ypbind* производит поиск NIS-серверов, обслуживающих домен, к которому принадлежит компьютер. Если этот поиск неуспешен, выдается сообщение об ошибке.

Проверка, может ли клиент получать NIS-карты с сервера, производится командой *ypcat <имя_NIS_карты>*.

Настройки NSS содержатся в файле */etc/nsswitch.conf*. Записи в этом файле имеют следующий формат:

<database> <lookup_order>

- *<database>* — имя системной базы данных, например:
- *<lookup_order>* — список имен служб, которые будут использоваться для поиска системной информации. Система будет обращаться к указанным службам в порядке перечисления. Для каждой базы данных список может быть своим. Поддерживаемые имена служб приведены в табл. 9.2.

Чтобы изменения в файле *nsswitch.conf* вступили в силу, нужно перезагрузить операционную систему.

9.2 Задание

Задание выполняется на двух-трех компьютерах под управлением Linux. Один их компьютеров выполняет функции сервера NFS и NIS, а остальные являются клиентами.

1. На сервере сконфигурировать службы NFS и NIS так, как указано в задании. Если экспортируемый каталог не существует на сервере, его нужно создать.
2. Настроить клиентские компьютеры для работы с NFS и NIS, проверить работоспособность полученной сети. Если в задании не указаны точки монтирования удаленных каталогов в локальное дерево каталогов, то их нужно создать в каталоге */mnt*.
3. С помощью анализатора пакетов проследить, к каким службам на сервере обращается клиент при выполнени указанных в задании действий, и какой обмен данными при этом происходит.

Таблица 9.2: Службы доступа к системной информации в Unix

Имя службы	**Описание**
files	Локальные файлы (*/etc/passwd*, */etc/group*, */etc/shadow* и др.).
db	Локальные файлы в формате Unix database (испольуются в некоторых Unix-системах вместо текстовых файлов).
nis	Служба NIS.
nisplus	Служба NIS+ (усовершенствованная NIS, применяемая в ОС фирмы Sun).
dns	Служба DNS (используется только для базы *hosts*).
compat	Устаревшая схема поиска системной информации в NIS, для использования которой нужно модифицировать локальные файлы. В современных Unix использовать эту службу не нужно.

Указания

1. Служба NFS настроена правильно, если клиент может смонтировать экспортируемые сервером каталоги с соответствующими правами доступа.
2. Если каталог на сервере экспортируется для чтения/записи, то на него нужно предоставить права на чтение и запись всем пользователям с помощью команды *chmod 777 <имя_каталога>*.
3. Для проверки работы NIS требуется создать на NIS-сервере нового пользователя командой *adduser <имя_пользователя>*. NIS работает правильно, если этот пользователь может зарегистрироваться на клиентском компьютере.

Вариант 1. Экспортировать каталог */home* для чтения/записи всем хостам сети. (*Указание:* на клиенте его нужно монтировать в */home*.) Сконфигурировать master NIS-сервер для домена *karashar*. Проанализировать запрос к NIS при входе пользователя в систему.

Вариант 2. Экспортировать каталог */var/tmp* для чтения/записи всем хостам сети. Сконфигурировать master NIS-сервер для домена *beytin*. Проанализировать запрос на удаленное монтирование каталога.

Вариант 3. Экспортировать каталог */var/tmp* для чтения/записи для одного хоста и только для чтения всем остальным хостам сети. Сконфигурировать master NIS-сервер для домена *aksu*. Проанализировать запрос на переименование файла в удаленном каталоге.

Вариант 4. Экспортировать каталог */var/tmp/public* для чтения/записи всем хостам сети, и каталог */var/tmp/readonly* только для чтения всем хостам сети. Сконфигурировать master NIS-сервер для домена *yarkend*. Проанализировать запрос на копирование файла в удаленный каталог.

Вариант 5. Экспортировать каталог */media/cdrom0*, в который монтируются компакт-диски, только для чтения всем хостам сети. (*Указание:* чтобы компакт-диск прочитался на клиенте, его нужно **перед удаленным монтированием** смонтировать на сервере командой *mount /media/cdrom*.) Сконфигурировать master NIS-сервер для домена *kashgar*. Проанализировать запрос на чтение файла с удаленного каталога.

Вариант 6. Экспортировать каталог */mnt/disk_d* только для чтения/записи всем хостам сети. Сконфигурировать master NIS-сервер для домена *turfan*. Проанализировать запрос к NIS при входе пользователя в систему.

Лабораторная работа 10.

Службы сетей Windows

Цель работы: изучение работы служб сетей Windows: службы имен NetBIOS и службы обозревателей сети.

10.1 Теоретические сведения

Первоначально прикладные службы сетей Windows использовали для обмена данными протокол **NetBIOS** (или его улучшенный вариант **NetBEUI**), который выполнял функции протоколов транспортного уровня. В дальнейшем, чтобы сети Windows могли использовать другие стеки протоколов, были разработаны реализации NetBIOS, работавшие поверх этих стеков. В этом случае NetBIOS выполняет роль протокола сеансового уровня.

В IP-сетях используется реализация **NetBIOS over TCP/IP** или **NBT**. С NBT работают сети Windows 9x и Windows NT, а также одноранговые сети Windows 2000 и Windows XP. Домены Windows 2000 и выше не используют NBT.

10.1.1 NetBIOS-имена

Для идентификации узлов и ресурсов сети Windows используются символьные **NetBIOS-имена** длиной 16 байт.

NetBIOS-имя имеет следующую структуру: 15 байт занимает собственно имя ресурса, а 16-й называется **NetBIOS-суффиксом** и опре-

деляет тип ресурса (клиент сети, файловый сервер, контроллер домена и т.д.). Список наиболее употребительных суффиксов приведен в таблице 10.1. NetBIOS-имена принято записывать в виде *name<20>*, где <20> — шестнадцатеричное представление суффикса.

NetBIOS-имена делятся на несколько классов:

1. **Уникальные** (Unique), которые могут использоваться только одним узлом в данной сети. Пример: собственное имя компьютера.
2. **Групповые** (Group), которые могут принадлежать нескольким узлам данной сети. Пример: имя рабочей группы или домена.

Адресация в сети NetBIOS является динамической. С каждым компьютером сети связан некоторый список имен. Различные службы при запуске могут добавлять к этому списку новые имена и удалять их при завершении работы. Процесс добавления имени называется **регистрацией**.

Список NetBIOS-имен компьютера можно просмотреть с помощью команды *nbtstat*.

10.1.2 Служба имен NetBIOS. WINS

Служба имен NetBIOS обеспечивает соответствие между NetBIOS-именем ресурса и IP-адресом компьютера. Она использует UDP-порт 137 и должна быть запущена на каждом узле сети. Список основных сообщений службы имен приведен в табл. 10.2.

Если вся сеть представляет собой одну IP-сеть, эта служба работает с помощью широковещательных сообщений. Для регистрации нового NetBIOS-имени компьютер несколько раз посылает на широковещательный адрес запрос, в котором указывает это NetBIOS-имя и свой IP-адрес. Если не будет получено сообщений о том, что данное имя уже принадлежит какому-нибудь узлу данной сети, процесс регистрации считается успешно завершенным. Если узел прекращает использование некоторого NetBIOS-имени, он оповещает об этом остальные узлы сети с помощью широковещательного сообщения.

Для определения IP-адреса, соответствующего NetBIOS-имени, компьютер посылает широковещательный запрос, содержащий это имя. Узел, которому принадлежит запрашиваемое имя, посылает в ответ пакет, в котором сообщает свой IP-адрес.

Полученное таким образом соответствие «NetBIOS-имя — IP-адрес» на некоторое время (обычно 10 минут) кэшируется. Поэтому в

Таблица 10.1: NetBIOS-суффиксы

NetBIOS-имя	Описание
machine<00>	**Workstation Service**, NetBIOS-имя компьютера в сети.
machine<03>	**Messenger Service**, служба обмена сообщениями с помощью *WinPopup* или *net send*.
machine<20>	**File Server Service**, файловый сервер.
workgroup<00>	**LAN Manager Browse Service**, имя рабочей группы (или домена), к которой принадлежит компьютер.
workgroup<1b>	**Domain Master Browser**, главный обозреватель домена в рабочей группе или в домене *workgroup*.
nt_domain<1c>	**Domain Controller**, любой контроллер домена *nt_domain* (главный или резервный). Главный контроллер регистрирует также имя *nt_domain<1b>*.
workgroup<1d>	**Local Master Browser**, главный обозреватель в рабочей группе *workgroup*.
workgroup<1e>	**Browse Election Service**, потенциальный обозреватель в рабочей группе *workgroup*.
__MSBROWSE__<01>	**Local Master Browser**, главный обозреватель сети; это имя используется для обмена информацией между главными обозревателями разных рабочих групп.
username<03>	**Messenger Service**, служба обмена сообщениями; используется так же, как *machine<03>*, но сообщение посылается пользователю *username*.

Таблица 10.2: Основные сообщения службы имен NetBIOS

Имя	Описание
Name Query Request	Запрос IP-адреса по NetBIOS-имени.
Name Query Responce	Ответ на запрос Name Query Request. Может быть положительным (Positive), если запрашваемое имя существует, либо отрицательным (Negative) в противном случае.
Name Registration Request	Запрос на регистрацию нового NetBIOS-имени в сети.
Name Registration Responce	Ответ на запрос Name Registration Request. Может быть положительным (Positive), если имя успешно зарегистрировано, либо отрицательным (Negative) в противном случае.
Name Release Request	Сообщение о прекращении использования NetBIOS-имени.
Name Release Responce	Подтверждение на сообщение Name Release Request (посылается WINS-сервером).
Name Refresh Request	Подтверждение использования NetBIOS-имени.

несоставной сети узлы обычно используют следующий алгоритм поиска IP-адреса по NetBIOS-имени: (1) кэш, (2) широковещательный запрос, (3) статический файл *lmhosts*. Узлы, работающие в таком режиме, называются **b-узлами**.

Для сети, состоящей из нескольких IP-сетей, предыдущий способ не подходит, так как широковещательные запросы распространяются только в пределах одной IP-сети. Поэтому в составной сети применяется служба **WINS** — Windows Internet Name Service (иногда употребляется термин **NBNS** — NetBIOS Name Service), которая также использует порт 137/UDP. Один компьютер в сети конфигурируется администратором как WINS-сервер, и его IP-адрес прописывается в

настройках каждого узла сети. WINS-сервер хранит централизованную базу NetBIOS-имен данной сети и соответствующих им IP-адресов.

В сети, использующей WINS, запрос на регистрацию нового NetBIOS-имени посылается WINS-серверу. Если данное имя не используется, узлу-отправителю посылается подтверждение, и новая запись заносится в базу. Каждая запись в базе имеет время жизни (TTL), по истечении которого она удаляется из базы, если от соответствующего узла не поступило подтверждение, что имя еще используется. Запись также удаляется из базы, если узел может явно сообщает WINS-серверу о прекращении использования NetBIOS-имени.

Для определения IP-адреса по NetBIOS-имени в составной сети узел отправляет запрос WINS-серверу, который посылает в ответ пакет, содержащий запрашиваемое имя и соответствующий IP-адрес. Результаты запросов кэшируются.

Таким образом, в составной сети для поиска IP-адреса по NetBIOS-имени используется следующий порядок: (1) кэш, (2) запрос к WINS-серверу, (3) файл *lmhosts*. Узлы, применяющие такой алгоритм, называются **р-узлами**.

Современные версии Windows могут использовать для установления соответствия между NetBIOS-именами и IP-адресами также службу DNS.

10.1.3 Служба обозревателей сети

Служба обозревателей сети (Browse service) отвечает за составление и хранения списка ресурсов рабочей группы или домена (**browse list**), который пользователи видят в окне «Сетевое окружение». Данный список является динамическим, так как необходимо отслеживать включение и выключение компьютеров, появление новых узлов и т.д.

Служба обозревателей сети представляет собой распределенную базу данных, хранящую browse list. Любой компьютер, который участвует в поддержке данной базы, называется **обозревателем сети (browser)**. Сообщения этой службы отправляются через службу NetBIOS-датаграмм, которая использует UDP-порт 138. Список основных сообщений службы обозревателей сети приведен в табл. 10.3.

Все обозреватели сети можно разделить на следующие категории.

1. **Потенциальные обозреватели** — компьютеры, которые способны поддерживать browse list. Они регистрируют специальное NetBIOS-имя *workgroup<1e>*.
2. **Главный обозреватель (Local Master Browser, LMB)** — компьютер, который хранит главную копию browse list для данной рабочей группы и отслеживает все происходящие с этим списком изменения. LMB дополнительно регистрирует NetBIOS-имена *workgroup<1d>* и *MSBROWSE<01>*[1]. Главным обозревателем становится один из потенциальных обозревателей, победивший на выборах LMB.
3. **Резервные обозреватели (Backup Browser)** — компьютеры, который также поддерживают службу обозревателей сети, но копию browse list получают от LMB. Резервными обозревателями становятся потенциальные обозреватели, либо получив специальное сообщение от LMB, либо по «собственной инициативе» (при этом они сообщают LMB о своем новом статусе). Список резервных обозревателей в данной рабочей группе известен LMB.

Остальные компьютеры являются клиентами службы обозревателей сети.

Служба обозревателей сети функционирует следующим образом. Каждый узел сети периодически посылает широковещательное сообщение о себе главному обозревателю. LMB, таким образом, формирует browse list для своей рабочей группы. Каждый резервный обозреватель периодически запрашивает и получает от LMB копию browse list.

Клиенты службы обозревателей сети запрашивают у LMB список резервных обозревателей, затем случайным образом выбирают из него три обозревателя и обращаются к ним за browse list. Это делается для уменьшения нагрузки на LMB. Передача browse list с обозревателя на клиентский компьютер осуществляется через службу NetBIOS-сессий, которая использует TCP-порт 139.

Если в сети существует несколько рабочих групп, то в каждой из них имеется свой главный обозреватель сети. Они периодически обмениваются широковещательными сообщениями, отправляемыми на

[1]На самом деле *<01><02>__MSBROWSE__<02><01>*, но далее будет использоваться сокращенное обозначение.

Таблица 10.3: Основные сообщения службы обозревателей сети

Имя	Описание
Host Announcement	Сообщение, содержащее информацию о некотором узле сети и о выполняемых им функциях (файловый сервер, потенциальный или резервный обозреватель и т.д.).
Local Master Announcement	Сообщение о существовании в рабочей группе главного обозревателя.
Domain Announcement	Сообщение о существовании в сети главного обозревателя некоторой рабочей группы. Содержит имя этой рабочей группы. Отправляется главным обозревателям других рабочих групп.
Get Backup List Request	Запрос на получение списка резервных обозревателей сети.
Get Backup List Response	Ответ на Get Backup List Request. Содержит список резервных обозревателей сети.
Become Backup Request	Сообщение, отправляемое потенциальному обозревателю для превращения его в резервный. В ответ новый резервный обозреватель посылает пакет Host Announcement, в котором указывает свой новый статус.
Request Election	Запрос на участие в выборах нового главного обозревателя сети. Содержит параметры претендента (уровень ОС UpTime и др.).

стандартное NetBIOS-имя *MSBROWSE<01>*. Такой запрос будет принят всеми узлами сети, но обработан только теми, у кого в списке имен присутствует *MSBROWSE<01>*, то есть, LMB во всех рабочих группах [2]. Таким образом, в browse list каждой рабочей группы имеются NetBIOS-имена главных обозревателей всех рабочих групп в данной IP-сети. Следовательно, клиенты имеют возможность запросить browse list любой рабочей группы.

В составной сети описанная схема не работает, так как обмен широковещательными запросами между разными IP-сетями невозможен. Для обеспечения правильной работы службы обозревателей в этом случае создается **главный обозреватель домена** (**Domain Master Browser, DMB**), который конфигурируется администратором[3]. DMB регистрирует в WINS уникальное имя *workgroup<1b>*. Главные обозреватели рабочих групп во всех подсетях, обнаружив с помощью WINS главный обозреватель домена, сообщают ему свои IP-адреса и периодически обмениваются с ним своими browse list. Таким образом, обмен информацией о ресурсах составной сети происходит через DMB.

Главный обозреватель сети периодически оповещает о своем существовании всех резервных обозревателей в своей рабочей группе. Если такие сигналы перестают поступать, один из резервных обозревателей инициирует процедуру выборов LMB. Запрос на процедуру выборов может дать вновь начавший работу потенциальный обозреватель «желающий» стать LMB.

В процессе выборов потенциальные обозреватели посылают широковещательные сообщения, в которых сообщают друг другу свои параметры. Если обозреватель обнаруживает, что параметры другого претендента лучше, чем его собственные, он считается проигравшим и не участвует в дальнейшем обмене сообщениями. Выборы проводятся в несколько раундов, по окончании которых должен остаться один победитель, который и становится главным обозревателем.

Способность потенциального обозреваеля победить на выборах определяется по нескольким параметрам. Прежде всего учитывает-

[2]Этот запрос является примером реализации схемы «multicast» через «broadcast», то есть, обмена данными по схеме «один – группе» через широковещательный запрос

[3]Теоретически главный обозреватель домена может существовать в сети, даже если в ней отсутствуют домены Windows. Практически сервер Windows не будет функционировать как DMB, если он не является главным контроллером домена, даже если указать в реестре соответствующее значение.

ся **уровень ОС** (OS level) узла, который зависит от установленной на нем операционной системы. Уровень ОС равен 32 для Windows NT Server, 16 для Windows NT Workstation и т.д. Далее учитывается еще ряд параметров, например, вероятность выиграть выборы повышается у главного контроллера домена, главного обозревателя домена, резервного обозревателя и т.д. Если все эти параметры одинаковы у двух претендентов, то сравнивается UpTime (время работы), и побеждает тот, у которого он больше. Если одинаковы UpTime, то победителем становится тот, чье NetBIOS-имя идет раньше по алфавиту.

10.2 Задание

Задание выполняется на 3-4 компьютерах под управлением Windows. Если требуется настроить WINS, один из компьютеров должен работать под управлением серверного варианта Windows.

1. Организовать одноранговую сеть Windows так, как указано в задании. Определив NetBIOS-имена, регистрируемые узлами сети, описать функции каждого компьютера в сети.
2. Пронаблюдать с помощью анализатора пакетов, какой обмен данными происходит при выполнении указанных в задании действий. Описать, какие сообщения и на какие адреса при этом посылаются.

Вариант 1. Объединить все компьютеры сети в рабочую группу *abar*. Проанализировать процесс регистрации NetBIOS-имен при включении компьютера в следующих случаях: (1) все имена уникальны и (2) имена двух компьютеров совпадают.

Вариант 2. Объединить все компьютеры сети в рабочую группу *eftalit*. Проанализировать процесс выборов главного обозревателя сети после выключения компьютера, который выполнял эту функцию.

Вариант 3. Объединить все компьютеры сети в рабочую группу *tele*. Проанализировать процесс получения browse list данной рабочей группы. (*Указание:* чтобы инициировать этот процесс, нужно зайти в сетевое окружение.)

Вариант 4. Объединить все компьютеры сети в рабочую группу *mukrin*. Проанализировать обмен данными при выполнении команды *nbtstat ip-addr*, где *ip-addr* — IP-адрес удаленного узла.

Вариант 5. Объединить все компьютеры сети в рабочую группу *savir*. Проанализировать процесс получения browse list другой рабочей группы. (*Указание:* чтобы инициировать этот процесс, нужно зайти в сетевое окружение.)

Вариант 6. Сконфигурировать один из компьютеров как WINS-сервер, остальные настроить на использование WINS. Объединить все компьютеры сети в рабочую группу *hionit*. Проанализировать процесс регистрации NetBIOS-имен при включении компьютера в следующих случаях: (1) все имена уникальны и (2) имена двух компьютеров совпадают.

Вариант 7. Сконфигурировать один из компьютеров как WINS-сервер, остальные настроить на использование WINS. Объединить все компьютеры сети в рабочую группу *karluk*. Проанализировать процесс выборов главного обозревателя сети после выключения компьютера, который выполнял эту функцию.

Вариант 8. Сконфигурировать один из компьютеров как WINS-сервер, остальные настроить на использование WINS. Объединить все компьютеры сети в рабочую группу *basmal*. Проанализировать процесс получения browse list данной рабочей группы. (*Указание:* чтобы инициировать этот процесс, нужно зайти в сетевое окружение.)

Вариант 9. Сконфигурировать один из компьютеров как WINS-сервер, остальные настроить на использование WINS. Объединить все компьютеры сети в рабочую группу *turgesh*. Проанализировать обмен данными при выполнении команды *nbtstat ip-addr*, где *ip-addr* — IP-адрес удаленного узла.

Вариант 10. Сконфигурировать один из компьютеров как WINS-сервер, остальные настроить на использование WINS. Объединить все компьютеры сети в рабочую группу *kimak*. Проанализировать процесс получения browse list другой рабочей группы. (*Указание:* чтобы инициировать этот процесс, нужно зайти в сетевое окружение.)

Лабораторная работа 11.

Домены Windows 2003

Цель работы: изучение доменов Windows 2000 и выше, получение практических навыков в создании и конфигурировании доменов, изучение работы вспомогательных служб доменов Windows 2000.

11.1 Теоретические сведения

11.1.1 Структура домена Windows 2000

Домен — группа компьютеров, объединенных общей базой данных пользователей. В домене существует как минимум один **контроллер домена** — выделенный сервер, хранящий базу данных пользователей и обслуживающий запросы на аутентификацию пользователей и компьютеров домена.

Особенности доменов Windows 2000 и выше.

1. Службы доменов Windows 2000 работают напрямую со стеком протоколов TCP/IP. Для идентификации доменов и компьютеров домена используются доменные имена. Поэтому для работы доменов Windows 2000 необходима служба DNS.
2. В качестве системы хранения служебной информации используется **служба каталогов Active Directory**. Она содержит информацию об объектах сети, таких как пользователи, компьютеры, сетевые ресурсы и принтеры общего доступа и др. Домены в каталоге Active Directory являются контейнерами, содержащими

входящие в них объекты. Серверами Active Directory являются контроллеры доменов. Для обращения к ним используется модифицированный протокол LDAP (**облегченный протокол доступа к каталогу**).

3. Основным способом аутентификации пользователей и компьютеров в домене является протокол **Kerberos**.

Домены Windows 2000 могут образовывать **деревья** — иерархическую структуру доменов, корнем которой является один родительский домен с уникальным именем. Имена поддоменов образуются по обычным правилам DNS. Группа доменных деревьев образует **лес**. Информация обо всех объектах данного доменного дерева или леса содержится в едином каталоге Active Directory.

Функционирование доменов Windows 2000 обеспечивают следующие службы.

1. Контроллеры домена — сервера Active Directory, хранящие информацию об объектах каталога, входящих в состав данного домена. Для обращения к ним используется TCP-порт 389.
2. **Глобальный справочник (Global Catalog, GC)** — сервер Active Directory, содержащий информацию об объектах каталога, вне зависимости от того, какому домену они принадлежат. Функции глобального справочника выполняют только некоторые контроллеры доменов. GC использует TCP-порт 3268.
3. **Центр распределения ключей Kerberos (Kerberos Key Distribution Center, KDC)** — служба аутентификации пользователей и компьютеров на контроллере домена. KDC использует протоколы TCP или UDP и порт 88.
4. **Служба паролей Kerberos (Kerberos Key Distribution Center, KDC)** — служба, позволяющая пользователям менять пароли для входа в домен. KDC использует протоколы TCP или UDP и порт 464.
5. Служба DNS — хранит информацию о сопоставлении имен компьютеров и доменов IP-адресам. Для обеспечения работы доменов Windows 2000 в DNS добавлен новый тип записей: **SRV (Служба)**. Записи этого типа имеют следующий формат:

```
_Service._Proto.DNS_name  SRV  [][preference][port] host
```

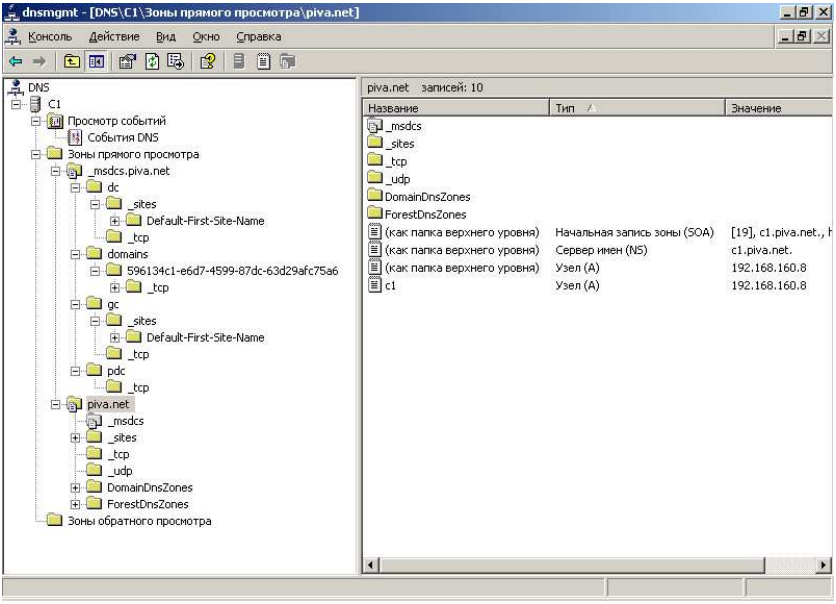

Рис. 11.1: Пример DNS для домена Windows 2000

Здесь:

- **_Service** — название службы (*_ldap* — контроллер домена, *_gc* — глобальный справочник, *_kerberos* — KDC, *_kpasswd* — служба паролей Kerberos);
- **_Proto** — протокол, используемый службой (*_tcp* либо *_udp*);
- **_DNS_name** — имя домена или одного из его поддоменов;
- **host** — доменное имя сервера, на котором работает служба;
- **port** — порт, используемый службой;
- **preference** — приоритет данного сервера.

В каждом домене существует вспомогательный поддомен *_msdcs*, который используется для группировки SRV-записей по ролям, которые выполняют сервера. Внутри него имеются следующие поддомены:

- *dc* — содержит список всех контроллеров данного домена;

- *gc* — содержит список всех серверов, выполняющих функции глобального справочника;
- *domains* — содержит список всех доменов и поддоменов;
- *pdc* — содержит список всех главных контроллеров домена;

Пример зоны DNS для домена Windows 2000 приведен на рис. 11.1. Служба DNS в Windows 2000 интегрирована в Active Directory, т.е., информация DNS хранится в каталоге Active Directory.

11.2 Задание

Задание выполняется на 3-4 компьютерах под управлением Windows. Один из компьютеров должен работать под управлением Windows 2000 Server или Windows 2003 Server.

1. Сконфигурировать сервер так, чтобы он выполнял роль контроллера для указанного в задании домена.
2. Ввести остальные компьютеры сети в домен. Проверить работу домена. (*Указание:* Для проверки работы домена требуется создать в домене нового пользователя. Домен работает правильно, если этот пользователь может зарегистрироваться на клиентском компьютере.)
3. С помощью анализатора пакетов проследить, к каким службам на сервере обращается клиент при выполнени указанных в задании действий, и какой обмен данными при этом происходит. (*Указание:* Анализатор пакетов нужно запускать на контроллере домена, если в задании не указано иное.)

Вариант 1. Объединить все компьютеры сети в домен *abar.local*. Проанализировать процесс ввода нового компьютера в домен.

Вариант 2. Объединить все компьютеры сети в домен *chigili.fcs*. Проанализировать обмена данными при включении входящего в домен компьютера.

Вариант 3. Объединить все компьютеры сети в домен *tatab.local*. Проанализировать процесс регистрации пользователя на клиентском компьютере.

Вариант 4. Объединить все компьютеры сети в домен *tangut.fcs*. Проанализировать процесс обращения пользователя на клиентском компьютере к сетевому ресурсу на другом клиентском компьютере.(*Указание:* Анализатор пакетов должен работать на том компьютере, на котором находится сетевой ресурс.)

Приложение А.

Анализатор пакетов Ethereal

Ethereal — мощный анализатор пакетов с графическим интерфейсом. Это свободно-распространяемый программный продукт, его реализации существуют как для Windows так и для Unix-подобных операционных систем.

А.1 Главное окно

Главное окно программы Ethereal (см. рис. А.1) разделено на три рабочие области. В верхней отображается список перехваченных пакетов. По умолчанию для каждого пакета указываются порядковый номер, время перехвата пакета, адреса отправителя и получателя источника[1], тип пакета (обычно транспортного уровня для данного пакета) и краткая информация об этом пакете. В средней рабочей области отображаются заголовки выбранного пакета. Кликнув на значок "+"около названия заголовка, можно получить развернутый список полей данного заголовка и их значений для данного пакета. Наконец, в нижней рабочей области отображается шестнадцатеричное представление выбранного пакета.

Перехваченные пакеты можно сохранить в файл с помощью меню

[1]В зависимости от типа пакета указываются либо IP-адреса, либо MAC-адреса

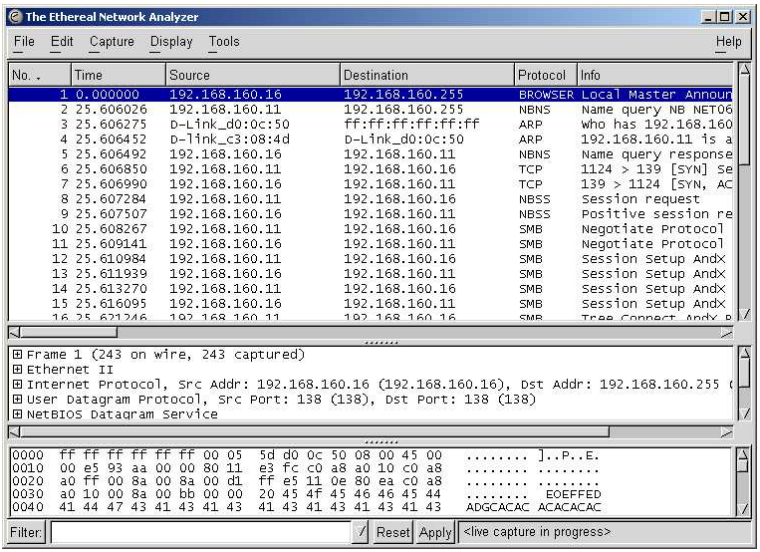

Рис. А.1: Главное окно Ethereal.

File - Save. Ранее сохраненный набор пакетов открывается с помощью меню File - Open.

А.2 Перехват пакетов

Начать перехват пакетов можно с помощью меню Capture - Start. При этом отображается окно, в котором можно задать параметры перехвата (см. рис. А.2). Ниже описаны некоторые из этих параметров.

С помощью выпадающего списка Interface можно выбрать сетевой интерфейс, на котором будет осуществляться перехват пакетов (полезно для компьютеров с несколькими сетевыми картами).

В поле Filter указывается логическое выражение, определяющее, какие именно пакеты будут перехватываться. Если оно не указано, то будут перехватываться все пакеты, пришедшие на выбранный сетевой интерфейс.

Наиболее распространенными выражениями являются:

- ключевое слово, определяющее протокол, например, tcp, udp,

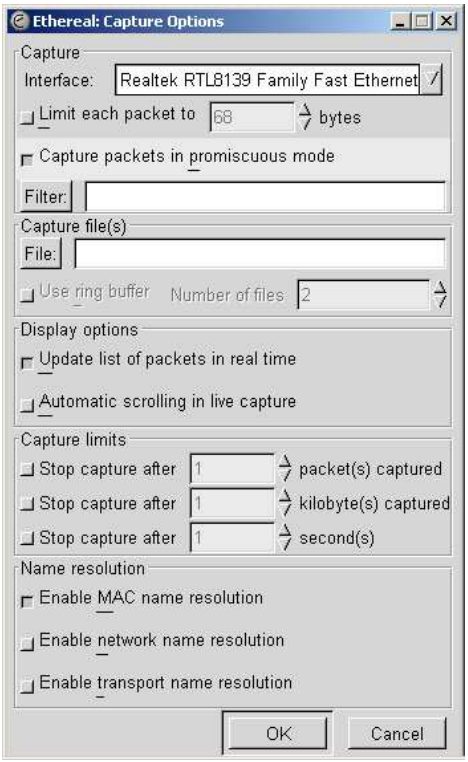

Рис. А.2: Ethereal.

icmp, ip; при этом будут перехвачены только пакеты, относящиеся к данному протоколу;

- host адрес_хоста — будут перехвачены пакеты, отправителем или получателем которых является хост с указанным адресом;
- port номер_порта — будут перехвачены TCP- и UDP-пакеты, у которых номер порта отправления или назначения совпадает с указанным номером порта.

Выражения можно комбинировать с помощью ключевых слов not (отрицание), and (логическое сложение) и or (логическое умножение). Например, если указать в поле Filter выражение tcp and host 192.168.1.1, то будут перехвачены только TCP-пакеты, отправителем

или получателем которых является хост с адресом 192.168.1.1. Более подробную информацию о фильтрации перехваченных пакетов можно получить из справки по программе Ethereal.

Если выбрана опция «Update list of packets in real time», то со списком перехваченных пакетов можно работать в процессе перехвата; вновь полученные пакеты будет добавляться к концу списка. В противном случае список пакетов отображается только после завершения перехвата.

Опции «Enable MAC name resolution», «Enable network name resolution» и «Enable transport name resolution» включают и выключают разпознавания MAC-адресов, IP-адресов и имен протоколов транспортного уровня. Если распознавание включено, то программа пытается вместо числового адреса или номера протокола подставить соответствующее символьное имя (например, доменное имя вместо MAC-адреса). Это часто приводит к замедлению работы программы, поэтому рекомендуется распознавание выключать.

После выбора всех необходимых параметров процесс перехвата можно начать, нажав кнопку «OK». В процессе перехвата на экране отображается окно статистики, сожержащее данные о количестве перехваченных пакетов с разбивкой по различным протоколам. Остановить процесс перехвата можно, нажав кнопку «Cancel» в окне статистики.

Приложение В.

Настройка сети в ОС Windows

В.1 Настройка сетевых интерфейсов в Windows

В ОС Windows настройка сетевых интерфейсов производится с помощью окна *Сеть и удаленный доступ к сети*, которое открывается с помощью *Пуск — Настройки — Панель управления — Сеть и удаленный доступ к сети*.

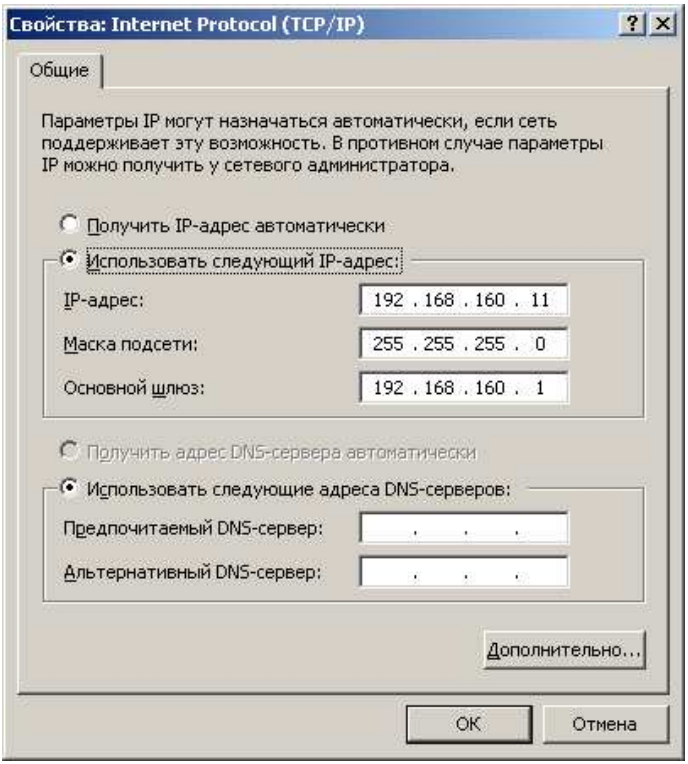

Рис. B.1: Настройка TCP/IP в Windows

Приложение C.

Настройка сети в ОС Linux

C.1 Настройка сетевых интерфейсов в Linux

В Unix-подобных операционных системах сетевые интерфейсы имеют символьные имена. Явный вид этих имен зависит от типа интерфейса и используемой операционной системы. В ОС Linux (независимо от дистрибутива) сетевые интерфейсы, соответствующие адаптерам Ethernet, именуются *eth + номер интерфейса*, начиная с 0. Т.е., если у компьютера одна сетевая карта, то соответствующий интерфейс будет иметь имя *eth0*. Если карт несколько, то связанные с ними интерфейсы будут иметь имена *eth0*, *eth1* и т.д.

Порядок нумерации карт определяется во время загрузки и зависит от конфигурации ядра ОС и настроек системы.

Интерфейс может быть **активным** и **неактивным**. В первом случае он может использоваться для передачи данных, во втором - нет.

Для конфигурирования сетевых интерфейсов в Unix-подобных ОС (в частности, в Linux) используется команда *ifconfig*. Ниже приводится краткий перечень ее опций и аргументов.

/sbin/ifconfig

Показывает список активных интерфейсов.

/sbin/ifconfig -a

Показывает список всех интерфейсов.

/sbin/ifconfig <ifname>

Показывает настройки интерфейса с именем *<ifname>*.

/sbin/ifconfig <ifname> up (down)

Делает интерфейс *<ifname>* активным (неактивным).

/sbin/ifconfig <ifname> <IP-addr> netmask <mask>

Назначает интерфейсу *<ifname>* IP-адрес *<IP-addr>* и маску подсети *<mask>*.

/sbin/ifconfig <ifname> broadcast <bcast>

Задает широковещательный адрес *<bcast>* для сети, к которой принадлежит интерфейс *<ifname>*.

Вообще говоря, широковещательный адрес вычисляется по IP-адресу и маске подсети, однако в Linux команда *ifconfig* делает это некорректно. Поэтому широковещательный адрес лучше указывать явно.

Параметры можно комбинировать, т.е., допустимо написать

/sbin/ifconfig <ifname> <IP-addr> netmask <mask> up

Данная команда назначит интерфейсу *ifname* IP-адрес и маску подсети и сделает его активным.

C.2 Настройка таблицы маршрутизации в Linux

Для манипулирования записями в таблицах маршрутизации в Unix-подобных ОС (в частности, в Linux) используется команда *route*.

Ниже приводится краткий перечень ключей и внутренних команд команды *route*.

/sbin/route

Показывает таблицу маршрутизации

Лучше при этом указывать ключ *-n*, так как в этом случае не производится поиск доменных имен, соответствующих адресам сетей и хостов.

/sbin/route add <type> <addr> netmask <mask> gw <gw_addr>

Добавляет запись в таблицу маршрутизации. Параметры имеют следующий смысл.

<type> — тип добавляемой записи. Возможные значения: *-net* — маршрут до сети; *-host* — маршрут до хоста. Если этот параметр не указан, считается, что запись имеет тип *-host*.

<addr> — адрес сети (или хоста), для которой прописывается маршрут.

<mask> — маска сети. Для хоста этот параметр можно опустить.

<gw_addr> — адрес маршрутизатора, через который доступна данная сеть (или хост).

/sbin/route add default gw <gw_addr>

Добавляет запись о маршруте по умолчанию через маршрутизатор с адресом *<gw_addr>*.

/sbin/route del <type> <addr> netmask <mask> gw <gw_addr>

Удаляет запись из таблицы маршрутизации. Параметры имеют тот же смысл, что и для комманды *add*.

Приложение D.

Утилиты сетевой диагностики

D.1 Утилита ping

Утилита *ping* предназначена для проверки доступности узла сети. Ее синтаксис:

ping [опции] host,

где *host* — IP-адрес или доменное имя узла сети.

При ее выполнении на тестируемый компьютер посылаются ICMP-запросы. Если за определенное время (таймаут) на них приходит ответ, на экран выводится сообщение о его получении. Если ответ не приходит ранее истечения таймаута, выдается сообщение «*Request timeout*». Если при доставке пакета возникла ошибка, то выводится сообщение об этой ошибке.

В ОС Windows команда *ping* по умолчанию посылает на тестируемый хост четыре запроса, после чего выводит на экран статистику (количество отправленных запросов, полученных ответов, процент потерь и др.) и завершается. В Unix-подобных ОС *ping* по умолчанию посылает запросы до тех пор, пока пользователь не прервет ее выполнение с помощью *Ctrl+C*, после чего также выводится статистика и программа завершается. Пример использования команды *ping* в Windows приведен на рис. D.1.

Список часто используемых опции команда *ping* приведен в

Таблица D.1: Опции команды *ping*

Windows	Unix	Описание
-t	—	Отправка пакетов на указанный узел, пока пользователь не прекратит ее с помощью *Ctrl+C*.
-n num	*-c num*	Число отправляемых запросов.
-l size	*-s size*	Размер поля данных отправляемых запросов в байтах (размер самих запросов будет равен size + 28 байт, т. к. к данным добавляются IP- и ICMP-заголовки).
-f	—	Установка флага, запрещающего фрагментацию пакета (в Unix он установлен по умолчанию).
-i ttl	*-t ttl*	Задание времени жизни пакета ("Time To Live").
—	*-R*	Записывать в заголовок IP-пакета информацию о маршруте пакета.

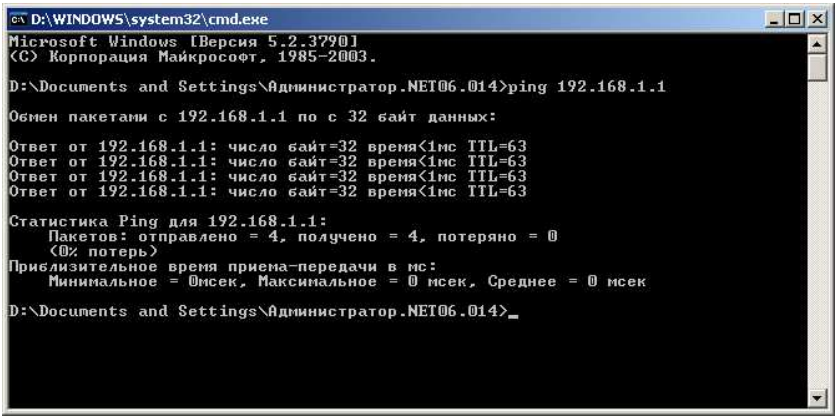

Рис. D.1: Использование команды *ping*

табл. D.1. Более подробную информацию можно получить, набрав *ping* без параметров в OS Windows или с помощью команды *man ping* в Unix-подобных ОС.

D.2 Утилита traceroute

Для определения маршрута до некоторого узла сети используется утилита *traceroute* (в ОС Windows она называется *tracert*).

Traceroute отправляет на тестируемый хост IP-пакеты группами по три. Пакеты сформированны так, что ответы на пакеты первой группы приходит от первого маршрутизатора в маршруте до тестируемого узла, на пакеты следующей группы — от второго, и так далее, пока не будет достигнут тестируемый узел, либо не превышено максимальное количество прыжков в данном сеансе работы (по умолчанию 30). *Traceroute* выводит на экран IP-адрес или доменное имя каждого маршрутизатора, а также интервал между отправкой каждого пакета и получением ответа на него в милисекундах, либо символ «*», если ожидание ответа прервано по таймауту. Пример использования команды *tracert* в Windows приведен на рис. D.2.

Команда *traceroute* имеет следующий синтаксис:

traceroute [опции] host,

где *host* — IP-адрес или доменное имя узла сети. Список основных опций приведен в табл. D.2. Более подробный список можно получить, набрав *tracert* без параметров в OS Windows или с помощью команды *man traceroute* в Unix-подобных ОС.

Таблица D.2: Опции команды *traceroute*

Windows	**Unix**	**Описание**
-d	*-n*	Не преобразовывать IP-адреса маршршрутизаторов в доменные имена.
-h num	*-c num*	Максимальное число прыжков при поиске узла.
-w time	*-w time*	Таймаут каждого ответа в миллисекундах.
-i ttl	*-t ttl*	Задание времени жизни пакета ("Time To Live").
—	*-I*	Использовать для тестирования ICMP-пакеты (в Windows ICMP-пакеты используются по умолчанию).

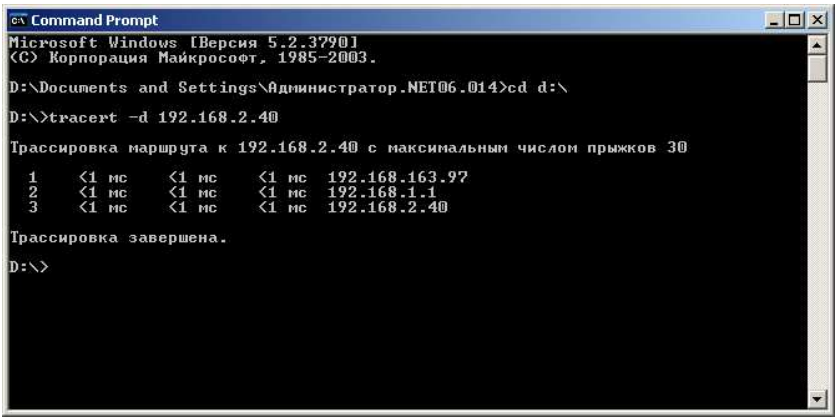

Рис. D.2: Использование команды *tracert*

Приложение Е.

Работа с системой DNS в Windows 2003 Server

Е.1 Настройка сервера DNS

В Windows 2003 Server имеется встроенный сервер DNS. Проверить, работает ли он, можно с помощью окна управления службами: *Пуск — Настройки — Администрирование — Службы* либо *Пуск — Администрирование — Службы*.

Управление сервером DNS осуществляется с помощью консоли DNS, запустить которую можно через меню *Пуск — Настройки — Администрирование — DNS* или *Пуск — Администрирование — DNS*. Главное окно консоли показано на рис. Е.1.

Создание новой зоны DNS. Для создания новой зоны DNS нужно выделить мышью раздел *Зоны прямого просмотра* или *Зоны обратного просмотра* в левой панели окна и в меню *Действие* выбрать пункт *Новая зона*. Будет запущен мастер создания зоны, который запросит у пользователя имя зоны, ее тип (основная или дополнительная) и другие необходимые параметры, а затем создаст указанную зону.

Управление зоной. Для управления зоной нужно выделить мышью название зоны и, кликнув правой кнопкой мыши, выбрать пункт *Свойства* в появившемся контекстовом меню. Откроется окно свойств зоны

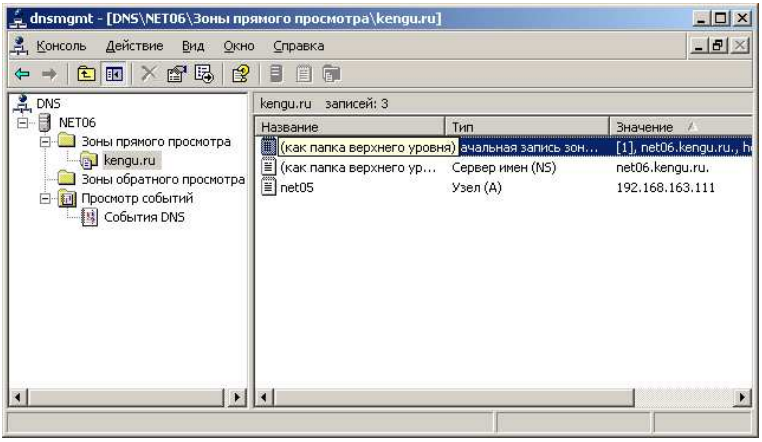

Рис. E.1: Главное окно консоли DNS.

(см. рис. E.2). С его помощью можно:

- управлять параметрами записи SOA данной зоны (вкладка *Начальная запись зоны (SOA)*);
- указать DNS-сервера, отвечающие за данную зону, т.е., создать записи NS для данной зоны (вкладка *Серверы имен*);
- управлять передачей информации о зоне на другие DNS-сервера (вкладка *Передачи зон*).

Модификация описания зоны. Для модификации описания зоны нужно выделить мышью название зоны и войти в меню *Действие*. С помощью этого меню можно добавить в зону записи типов A, MX, PTR (для обратной зоны).

Для удаления записи из зоны нужно выделить ее мышью, вызвать контекстовое меню правой кнопкой и выбрать пункт *Удалить*.

Для редактирования записи в ее контекстовом меню нужно выбрать пункт *Свойства*.

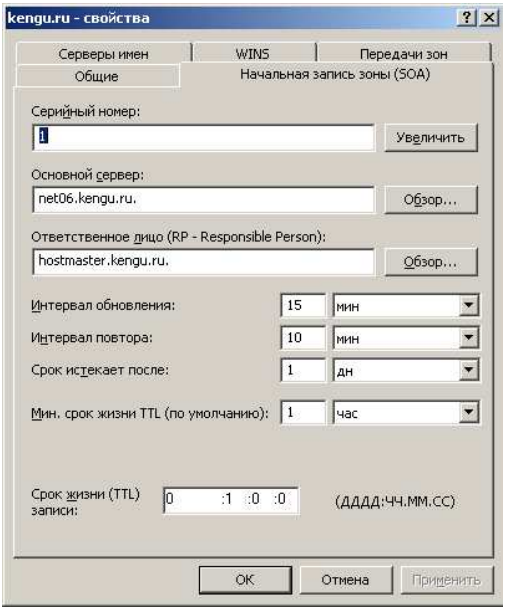

Рис. Е.2: Главное окно консоли DNS.

Е.2 Утилита nslookup

Утилита *nslookup* предназначена для тестирования системы DNS. Ее синтаксис:

nslookup [host] [server],

где *host* — доменное имя или IP-адрес компьютера, *server* — DNS-сервер, которому будет послан запрос на поиск информации. Если аргумент *host* представляет из себя доменное имя, то будет осуществлен поиск IP-адреса, соответствующего этому имени (прямой DNS-запрос). Если этот аргумент — IP-адрес, то будет проведен поиск доменного имени, сопоставленного этому IP-адресу (обратный DNS-запрос).

Если второй аргумент опущен, то запрос будет отправлен предпочитаемому DNS-серверу, который указан в настройках протокола TCP/IP.

Если *nslookup* вызвана без параметров, она переходит в режим

командной строки. Этот режим позволяет посылать на DNS-сервер DNS-запросы разных типов, менять параметры запросов и др. Справку по командам, доступным в этом режиме, можно получить по команде *help*. Выход из режима командной строки осуществляется командой *exit*.