Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Институт повышения квалификации и переподготовки

Кафедра «Профессиональная переподготовка»

А. Н. Осипенко

КОМПЬЮТЕРНЫЕ СЕТИ

ПРАКТИКУМ
по одноименному курсу
для слушателей специальностей
1-40 01 74 «Web-дизайн и компьютерная графика»
и 1-40 01 73 «Программное обеспечение информационных систем»
заочной формы обучения

Гомель 2015

УДК 681.3.06(075.8)
ББК 32.973.202я73
О-74

Рекомендовано кафедрой «Профессиональная переподготовка»
ГГТУ им. П. О. Сухого ИПКиП
(*протокол № 1 от 25.09.2015 г.*)

Рецензент: канд. техн. наук, доц. каф. математических проблем управления
ГГУ им. Ф. Скорины *С. Ф. Маслович*

Осипенко, А. Н.

О-74 Компьютерные сети : практикум по одноим. курсу для слушателей специальностей 1-40 01 74 «Web-дизайн и компьютерная графика» и 1-40 01 73 «Программное обеспечение информационных систем» заоч. формы обучения / А. Н. Осипенко. – Гомель : ГГТУ им. П. О. Сухого, 2015. – 88 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: https://elib.gstu.by. – Загл. с титул. экрана.

Практикум содержит теоретические сведения и задания для практического освоения курса «Компьютерные сети». Задания включают в себя такие темы, как системные службы операционных систем мониторинга и настройки сети; сети Ethernet: используемое сетевое оборудование, топологии сетей; стек протоколов TCP/IP; создание правил маршрутизации и конфигурирование сетевых интерфейсов.

Издание адресовано слушателям ИПКиП специальностей 1-40 01 74 «Web-дизайн и компьютерная графика» и 1-40 01 73 «Программное обеспечение информационных систем».

УДК 681.3.06(075.8)
ББК 32.973.202я73

Оглавление

Предисловие 4
1 Лабораторная работа №1. Системные службы операционных систем мониторинга и настройки сети 5
1.1 Теоретические сведения. Основные команды ОС Windows для работы с сетевыми ресурсами 5
1.2 Задание на лабораторную работу 33
1.3 Индивидуальные задания 33
1.4 Контрольные вопросы 34
2 Лабораторная работа №2. Сети Ethernet: используемое оборудование. Топологии сетей 36
2.1 Теоретические сведения. Описание программы моделирования работы компьютерных сетей *Network Emulator* 36
2.2 Задание на лабораторную работу 39
2.3 Контрольные вопросы 40
3 Лабораторная работа №3. Сетевое оборудование Ethernet 41
3.1 Теоретические сведения 41
3.1 Задание на лабораторную работу 71
3.2 Контрольные вопросы 71
4 Лабораторная работа №4. Стек протоколов TCP/IP. Создание правил маршрутизации 73
4.1 Теоретические сведения 73
4.2 Задание на лабораторную работу 77
4.3 Контрольные вопросы 78
5 Лабораторная работа №5. Конфигурирование сетевых интерфейсов 79
5.1 Теоретические сведения 79
5.2 Задание на лабораторную работу 87
5.3 Контрольные вопросы 87
Список использованных источников 88

Предисловие

Без освоения базовых принципов построения компьютерных систем, в частности, Интернета, невозможно стать полноценным веб-дизайнером или программистом. Сетевые технологии с каждым годом все больше внедряются в повседневную жизнь всего населения и, в первую очередь, – инструментальную среду IT-специалистов.

Настоящий лабораторный практикум разработан в соответствии с программой дисциплины «Компьютерные сети» и предназначен для обучения слушателей ИПК и П заочной формы обучения по специальностям 1-40 01 74 «Web-дизайн и компьютерная графика» и 1-40 01 73 «Программное обеспечение информационных систем». Он включает в себя задания для лабораторных занятий, необходимые теоретические сведения и пояснения к работе с сетевым программным обеспечением, список литературы.

Цель заданий – приобретение практических навыков по наиболее сложным темам курса, требующим умения работать с сетевыми инструментами операционной системы применительно к локальной сети или сети Интернет.

Задания пяти лабораторных работ включают в себя такие темы, как системные службы операционных систем мониторинга и настройки сети; сети Ethernet: используемое сетевое оборудование. топологии сетей; стек протоколов TCP/IP; создание правил маршрутизации и конфигурирование сетевых интерфейсов.

Каждая тема сопровождается методическими рекомендациями, содержащими базовые понятия, необходимые для выполнения заданий. Для закрепления материала в конце каждой лабораторной работы даются контрольные вопросы.

1 Лабораторная работа №1. Системные службы операционных систем мониторинга и настройки сети

1.1 Теоретические сведения. Основные команды ОС Windows для работы с сетевыми ресурсами

Arp

Служит для вывода и изменения записей кэша протокола ARP, который содержит одну или несколько таблиц, использующихся для хранения IP-адресов и соответствующих им физических адресов Ethernet или Token Ring. Для каждого сетевого адаптера Ethernet или Token Ring, установленного в компьютере, используется отдельная таблица. Запущенная без параметров, команда **arp** выводит справку.

Синтаксис

arp [**-a** [*инет_адрес*] [**-N** *иф_адрес*]] [**-g** [*инет_адрес*] [**-N** *иф_адрес*]] [**-d** *инет_адрес* [*иф_адрес*]] [**-** *инет_адрес е_адрес* [*иф_адрес*]]

Параметры

-a [*инет_адрес*] [**-N** *иф_адрес*] – вывод таблиц текущего протокола ARP для всех интерфейсов. Чтобы вывести записи ARP для определенного IP-адреса, воспользуйтесь командой **arp -a** с параметром *инет_адрес*, где *инет_адрес* — это IP-адрес. Чтобы вывести таблицы кэша ARP для определенного интерфейса, укажите параметр **-N** *иф_адрес*, где *иф_адрес* — это IP-адрес, назначенный интерфейсу. Параметр **-N** вводится с учетом регистра;

-g [*инет_адрес*] [**-N** *иф_адрес*] – совпадает с **–a;**

-d *инет_адрес* [*иф_адрес*] – удаление записи с определенным IP-адресом, где *инет_адрес* — это IP-адрес. Чтобы запись таблицы для определенного интерфейса, укажите параметр *иф_адрес*, где *иф_адрес* — это IP-адрес, назначенный интерфейсу. Чтобы удалить все записи, введите звездочку (*) вместо параметра *инет_адрес;*

-s *инет_адрес е_адрес* [*иф_адрес*] – добавление статической записи, которая сопоставляет IP-адрес *инет_адрес* с физическим адресом *е_адрес*, в кэш ARP. Чтобы добавить статическую запись кэша ARP в таблицу для определенного интерфейса, укажите параметр *иф_адрес*, где *иф_адрес* — это IP-адрес, назначенный интерфейсу;

/? – отображение справки в командной строке.

Заметки

- IP-адреса для параметров *инет_адрес* и *иф_адрес* записываются в точечно-десятичной нотации;
- физический адрес для параметра *е_адрес* состоит из шести байт, записанных в шестнадцатеричном формате и разделенных дефисами (например 00-AA-00-4F-2A-9C);
- записи, добавленные с параметром **-s**, являются статическими и не удаляются из кэша ARP после истечения периода времени. Записи удаляются, если остановлен и запущен протокол TCP/IP. Чтобы создать постоянные статические записи кэша ARP, введите соответствующие команды **arp** и воспользуйтесь **планировщиком заданий** для выполнения этого файла при запуске.

Примеры

Чтобы вывести таблицы кэша ARP для всех интерфейсов, введите:

arp -a

Чтобы вывести таблицу кэша ARP для интерфейса, которому назначен IP-адрес 10.0.0.99, введите:

arp -a -N 10.0.0.99

Чтобы добавить статическую запись кэша ARP, которая сопоставляет IP-адрес 10.0.0.80 с физическим адресом 00-AA-00-4F-2A-9C, введите:

arp – 10.0.0.80 00-AA-00-4F-2A-9C

Hostname

Отображение имени узла, входящего в состав полного имени компьютера.

Синтаксис

hostname

Параметры

/? – отображение справки в командной строке;

Ipconfig

Служит для отображения всех текущих параметров сети TCP/IP и обновления параметров DHCP и DNS. При вызове команды **ipconfig** без параметров выводится только IP-адрес, маска подсети и основной шлюз для каждого сетевого адаптера.

Синтаксис

ipconfig [**/all**] [**/renew** [*адаптер*]] [**/release** [*адаптер*]] [**/flushdns**] [**/displaydns**] [**/registerdns**] [**/showclassid** *адаптер*] [**/setclassid** *адаптер* [*код_класса*]]

Параметры

/all – вывод полной конфигурации TCP/IP для всех адаптеров. Без этого параметра команда **ipconfig** выводит только IP-адреса, маску подсети и основной шлюз для каждого адаптера. Адаптеры могут представлять собой физические интерфейсы, такие как установленные сетевые адаптеры, или логические интерфейсы, такие как подключения удаленного доступа;

/renew [*адаптер*] – обновление конфигурации DHCP для всех адаптеров (если адаптер не задан) или для заданного *адаптера*. Данный параметр доступен только на компьютерах с адаптерами, настроенными для автоматического получения IP-адресов. Чтобы указать адаптер, введите без параметров имя, выводимое командой **ipconfig**;

/release [*адаптер*] – отправка сообщения DHCPRELEASE серверу DHCP для освобождения текущей конфигурации DHCP и удаление конфигурации IP-адресов для всех адаптеров (если адаптер не задан) или для заданного *адаптера*. Этот адаптер отключает протокол TCP/IP для адаптеров, настроенных для автоматического получения IP-адресов. Чтобы указать адаптер, введите без параметров имя, выводимое командой **ipconfig**;

/flushdns – сброс и очистка содержимого кэша сопоставления имен DNS клиента. Во время устранения неполадок DNS эту процедуру используют для удаления из кэша записей отрицательных попыток сопоставления и других динамически добавляемых записей;

/displaydns – отображение содержимого кэша сопоставления имен DNS клиента, включающего записи, предварительно загруженные из локального файла Hosts, а также последние полученные записи ресурсов для запросов на сопоставление имен. Эта информация используется службой DNS клиента для быстрого сопоставления часто встречаемых имен без обращения к указанным в конфигурации DNS-серверам;

/registerdns – динамическая регистрация вручную имен DNS и IP-адресов, настроенных на компьютере. Этот параметр полезен при устранении неполадок в случае отказа в регистрации имени DNS или при выяснении причин неполадок динамического обновления между

клиентом и DNS-сервером без перезагрузки клиента. Имена, зарегистрированные в DNS, определяются параметрами DNS в дополнительных свойствах протокола TCP/IP;

/showclassid *адаптер* – отображение кода класса DHCP для указанного адаптера. Чтобы просмотреть код класса DHCP для всех адаптеров, вместо параметра *адаптер* укажите звездочку (*****). Данный параметр доступен только на компьютерах с адаптерами, настроенными для автоматического получения IP-адресов;

/setclassid *адаптер* [*код_класса*] – задание кода класса DHCP для указанного адаптера. Чтобы задать код класса DHCP для всех адаптеров, вместо параметра *адаптер* укажите звездочку (*****). Данный параметр доступен только на компьютерах с адаптерами, настроенными для автоматического получения IP-адресов. Если код класса DHCP не задан, текущий код класса удаляется.

Заметки

- Данная команда доступна только на компьютерах с адаптерами, настроенными для автоматического получения IP-адресов. Это позволяет пользователям определять, какие значения конфигурации были получены с помощью DHCP, APIPA или другой конфигурации;
- если имя *адаптер* содержит пробелы, его следует заключать в кавычки (т. е. **"***имя_адаптера***"**);
- в именах адаптеров, задаваемых для команды **ipconfig**, поддерживается использование подстановочного знака звездочки (*) для задания имен, начинающихся с указанной строки или содержащих указанную строку. Например, имя **Подкл*** будет включать все адаптеры, начинающиеся со строки «Подкл», а имя ***сет*** — все адаптера, содержащие строку «сет».

Примеры

Чтобы вывести основную конфигурацию TCP/IP для всех адаптеров, введите:

ipconfig

Чтобы вывести полную конфигурацию TCP/IP для всех адаптеров, введите:

ipconfig /all

Чтобы обновить конфигурацию IP-адреса, назначенного DHCP-сервером, только для адаптера **Подключение по локальной сети**, введите:

ipconfig /renew "Подключение по локальной сети"

Чтобы сбросить кэш сопоставления имен DNS при наличии неполадок в сопоставлении имен, введите:

ipconfig /flushdns

Чтобы вывести код класса DHCP для всех адаптеров с именами, начинающимися со слова *Подключение*, введите:

ipconfig /showclassid Подключение*

Чтобы задать код класса DHCP *TEST* для адаптера **Подключение по локальной сети**, введите:

ipconfig /setclassid "Подключение по локальной сети" TEST

Nbtstat

Служит для отображения статистики протокола NetBIOS over TCP/IP (NetBT), таблиц имен NetBIOS для локального и удаленного компьютеров, а также кэша имен NetBIOS. Команда **Nbtstat** позволяет обновить кэш имен NetBIOS и имена, зарегистрированные в службе имен Интернета Windows (WINS). Запущенная без параметров, команда **nbtstat** выводит справку.

Синтаксис

nbtstat [**-a** *удаленное_имя*] [**-A** *IP-адрес*] [**-c**] [**-n**] [**-r**] [**-R**] [**-RR**] [**-s**] [**-S**] [*интервал*]

Параметры

-a *удаленное_имя* – отображение таблицы имен NetBIOS удаленного компьютера, где *удаленное_имя* является именем NetBIOS удаленного компьютера. Таблица имен NetBIOS является списком имен NetBIOS, соответствующих приложениям NetBIOS, работающим на данном компьютере;

-A *IP-адрес* – отображение таблицы имен NetBIOS удаленного компьютера, заданного IP-адресом (десятичные числа, разделенные точками);

-c – отображение содержимого кэша имен NetBIOS, таблицы имен NetBIOS и их разрешенных IP-адресов;

-n – отображение таблицы имен NetBIOS локального компьютера. Состояние **Зарегистрирован** означает, что это имя зарегистрировано на сервере WINS или в качестве широковещательного адреса;

-r – отображение статистики разрешения имен NetBIOS. На компьютере Windows XP, настроенном для использования WINS, этот

параметр возвращает количество имен, разрешенных и зарегистрированных для широковещательной рассылки или WINS;

-R – очистка содержимого кэша имен NetBIOS и перезагрузка записей #PRE из файла Lmhost;

-RR – освобождение и обновление имен NetBIOS для локального компьютера, зарегистрированного на серверах WINS;

-s – отображение сеансов клиента и сервера NetBIOS с попыткой преобразования конечного IP-адреса в имя;

-S – вывод сведений о работе сервера и клиента NetBIOS; удаленные компьютеры выводятся только по IP-адресам;

интервал – обновление выбранной статистики на экране через промежутки времени, заданные значением *интервал*. Нажатие клавиш CTRL+C останавливает обновление статистики. Если этот параметр не задан, команда **nbtstat** выводит сведения о текущей конфигурации один раз.

Заметки

- При задании параметров команды **nbtstat** учитывается регистр символов;
- в таблице 1.1 приведены заголовки столбцов, отображаемые программой **nbtstat**.

Таблица 1.1 – Заголовки столбцов программы **nbtstat**.

Заголовок	**Описание**
Ввод	Число полученных байт.
Вывод	Число отправленных байт.
Вид	Направление передачи от локального компьютера (Исх) или от удаленного компьютера (Вхд).
Время жизни	Время, оставшееся до сброса элемента кэша таблицы имен.
Локальное имя	Локальное имя NetBIOS, соответствующее данному подключению.
Удаленный узел	Имя или IP-адрес удаленного компьютера.
<03>	Последний байт имени NetBIOS, преобразованный в шестнадцатеричную форму. Каждое имя NetBIOS может иметь длину 16 знаков. Последний байт часто имеет специальное значение, так как одно имя может встречаться несколько раз на одном компьютере, различаясь только последним байтом.

Тип	Тим имени. Имя может быть уникальным именем или именем группы.
Состояние	«Зарегистрирован» (служба NetBIOS работает на удаленном компьютере) или «Конфликт» (в службе уже зарегистрировано такое же имя компьютера).

В таблице1.2 приведены возможные состояния подключения NetBIOS.

Таблица 1.2 – Состояния подключения NetBIOS.

Состояние	Описание
Подключен	Сеансовое подключение установлено.
Назначен	Конечная точка подключения создана и связана с IP-адресом.
Ожидание	Конечная точка доступна для входящих подключений.
Простаивает	Конечная точка создана, но подключение не получено.
Подключается	Сеанс в состоянии подключения, сопоставление имени и IP адреса для точки назначения определено.
Прием	Запрос на входящее подключение принят, подключение будет установлено.
Повторное подключение	Повторная попытка установки подключения (после первой неудачной попытки).
Исходящий	Сеанс находится в процессе подключения, создается подключение TCP.
Входящий	Сеанс находится в процессе подключения.
Отключение	Сеанс находится в процессе отключения.
Отключен	Локальный компьютер отправил запрос на отключение и ожидает подтверждения от удаленной системы.

Примеры

Чтобы вывести таблицу имен удаленного компьютера, имеющего имя NetBIOS CORP07, введите:

nbtstat -a CORP07

Чтобы вывести таблицу имен NetBIOS удаленного компьютера, имеющего IP-адрес 10.0.0.99, введите:

nbtstat -A 10.0.0.99

Чтобы вывести таблицу имен локального компьютера, введите:

nbtstat -n

Чтобы вывести содержимое кэша имен NetBIOS локального компьютера, введите:

nbtstat -c

Чтобы очистить кэш имен NetBIOS и перезагрузить записи #PRE из локального файла Lmhosts, введите:

nbtstat -R

Чтобы освободить имена NetBIOS, зарегистрированные на сервере WINS, и снова зарегистрировать их, введите:

nbtstat -RR

Чтобы просмотреть статистику сеанса NetBIOS по IP-адресу с обновлением каждые пять секунд, введите:

nbtstat -S 5

Netstat

Отображение активных подключений TCP, портов, прослушиваемых компьютером, статистики Ethernet, таблицы маршрутизации IP, статистики IPv4 (для протоколов IP, ICMP, TCP и UDP) и IPv6 (для протоколов IPv6, ICMPv6, TCP через IPv6 и UDP через IPv6). Запущенная без параметров, команда **nbtstat** отображает подключения TCP.

Синтаксис

netstat [**-a**] [**-e**] [**-n**] [**-o**] [**-p** *протокол*] [**-r**] [**-s**] [*интервал*]

Параметры

-a – вывод всех активных подключений TCP и прослушиваемых компьютером портов TCP и UDP;

-e – вывод статистики Ethernet, например количества отправленных и принятых байтов и пакетов. Этот параметр может комбинироваться с ключом **–s;**

-n – вывод активных подключений TCP с отображением адресов и номеров портов в числовом формате без попыток определения имен;

-o – вывод активных подключений TCP и включение кода процесса (PID) для каждого подключения. Код процесса позволяет найти приложение на вкладке **Процессы** диспетчера задач Windows. Этот параметр может комбинироваться с ключами **-a**, **-n** и **–p;**

-p *протокол* – вывод подключений для протокола, указанного параметром *протокол*. В этом случае параметр *протокол* может принимать значения **tcp**, **udp**, **tcpv6** или **udpv6**. Если данный параметр используется с ключом **-s** для вывода статистики по протоколу, параметр *протокол* может иметь значение **tcp**, **udp**, **icmp**, **ip**, **tcpv6**, **udpv6**, **icmpv6** или **ipv6;**

-s – вывод статистики по протоколу. По умолчанию выводится статистика для протоколов TCP, UDP, ICMP и IP. Если установлен протокол IPv6 для Windows XP, отображается статистика для протоколов TCP через IPv6, UDP через IPv6, ICMPv6 и IPv6. Параметр **-p** может использоваться для указания набора протоколов;

-r вывод содержимого таблицы маршрутизации IP. Эта команда эквивалентна команде **route print;**

интервал – обновление выбранных данных с интервалом, определенным параметром *интервал* (в секундах). Нажатие клавиш CTRL+C останавливает обновление. Если этот параметр пропущен, **netstat** выводит выбранные данные только один раз.

Заметки

- Параметрам, используемым с данной командой, должен предшествовать дефис (**-**), а не косая черта (/);
- команда **Netstat** выводит статистику для следующих объектов.
  - Протокол – имя протокола (TCP или UDP);
  - Локальные адреса – IP-адрес локального компьютера и номер используемого порта. Имя локального компьютера, соответствующее IP-адресу и имени порта, выводится только в том случае, если не указан параметр **-n**. Если порт не назначен, вместо номера порта будет выведена звездочка (*);
  - Внешние адреса – IP-адрес и номер порта удаленного компьютера, подключенного к данному сокету. Имена, соответствующие IP-адресу и порту, выводятся только в том случае, если не указан параметр **-n**. Если порт не назначен, вместо номера порта будет выведена звездочка (*).
  - (Состояние) – указание состояния подключения TCP. Возможные значения: CLOSE_WAIT; CLOSED; ESTABLISHED; FIN_WAIT_1; FIN_WAIT_2; LAST_ACK; LISTEN; SYN_RECEIVED; SYN_SEND; TIMED_WAIT.

Для получения дополнительных сведений о состояниях подключения TCP см. документ RFC 793.

Примеры

Для вывода статистики Ethernet и статистики по всем протоколам введите следующую команду:

netstat -e -s

Для вывода статистики только по протоколам TCP и UDP введите следующую команду:

netstat -s -p tcp udp

Для вывода активных подключений TCP и кодов процессов каждые 5 секунд введите следующую команду:

nbtstat -o 5

Для вывода активных подключений TCP и кодов процессов каждые с использованием числового формата введите следующую команду:

nbtstat -n –o

Nslookup

Предоставляет сведения, предназначенные для диагностики инфраструктуры DNS. Для использования этого средства необходимо быть знакомым с принципами работы системы DNS. Средство командной строки Nslookup доступно, только если установлен протокол TCP/IP.

Синтаксис

nslookup [**-***подкоманда ...*] [{*искомый_компьютер*| [**-***сервер*]}]

Параметры

-*подкоманда ...*

Задает одну или несколько подкоманд **nslookup** как параметры командной строки. Список подкоманд см. в разделе «См. также».

искомый_компьютер

Ищет данные для параметра *искомый_компьютер*, используя текущий, заданный по умолчанию сервер имен DNS, если никакого другого сервера не указано. Чтобы получить сведения о компьютере не из текущего домена DNS, в конец имени должна быть добавлена точка.

-*сервер*

Указывает, что данный сервер следует использовать в качестве сервера имен DNS. Если параметр *-сервер* не указан, используется сервер DNS, заданный по умолчанию.

{**help**|**?**}

Выводит краткое описание подкоманд **nslookup**.

Заметки

- Если *искомый_компьютер* задан IP-адресом, а запрашивается запись ресурса типа A или PTR, будет выведено имя компьютера. Если *искомый_компьютер* задан именем без замыкающей точки, имя домена DSN, используемого по умолчанию, будет добавлено к указанному имени. Поведение зависит от состояния следующих подкоманд команды **set**: **domain**, **srchlist**, **defname** и **search;**
- если в командной строке введен дефис (-) вместо параметра *искомый_компьютер*, команда **nslookup** перейдет в интерактивный режим;
- длина строки вызова команды не может превышать 256 символов;
- команда **nslookup** может работать в двух режимах: интерактивном и обычном (автономном);
- если требуется вывод только небольшой части информации, следует использовать обычный режим. В качестве первого параметра следует использовать имя или IP-адрес компьютера, о котором требуется получить данные. В качестве второго параметра введите имя или IP-адрес сервера имен DNS. Если второй параметр не задан, командой **nslookup** используется сервер имен DNS, установленный по умолчанию;
- если требуется получить более полные сведения, следует использовать интерактивный режим. В качестве первого параметра следует ввести знак дефиса (-) и имя или IP-адрес сервера имен DNS в качестве второго параметра. Если оба параметра не заданы, командой **nslookup** используется сервер имен DNS, установленный по умолчанию. Далее перечислено несколько советов по работе в интерактивном режиме:
  - для прерывания интерактивной команды в любой момент следует нажать CTRL+B;
  - для выхода необходимо ввести **exit;**

- для ввода имени компьютера, совпадающего с какой-либо командой, перед именем следует ввести обратную косую черту (\).
- нераспознанные команды воспринимаются как имена компьютеров;

- если при обработке запроса возникла ошибка, командой **nslookup** на экран будет выведено сообщение. В таблице 1.3 перечислены возможные сообщения об ошибках.

Примеры

Каждый параметр состоит из дефиса (-) и следующей за ним без пробелов команды, а также, в некоторых случаях, знака равенства (=) и значения. Например, чтобы изменить установленный по умолчанию тип запроса о сведениях для узла и установить начальное время ожидания равным 10 секундам, следует ввести команду:

nslookup -querytype=hinfo -timeout=10

Таблица 1.3 – Сообщения об ошибках команды **nslookup.**

Сообщение об ошибке	**Описание**
Timed out	Сервер не ответил на запрос в течение определенного времени и после определенного числа повторных попыток. Имеется возможность установить период ожидания с помощью подкоманды **set timeout**. Имеется возможность установить число повторных попыток с помощью подкоманды **set retry**.
No response from server	Сервер имен DNS не запущен на сервере
No records	Сервер имен DNS не содержит записей о ресурсах указанного типа, хотя имя сервера задано верно. Тип запроса задается командой **set querytype**.
Nonexistent domain	Заданный компьютер или имя домена DNS не существует.
Connection refused -или- Network is unreachable	Невозможно подключиться к серверу имен DNS или к серверу службы finger. Эта ошибка обычно возникает с запросами команд **ls** и **finger**.

Server failure	Сервер имен DNS обнаружил внутреннее несоответствие в своей базе данных и не может корректно ответить на запрос.
Refused	Отказано в обработке запроса сервером имен DNS.
Format error	Сервер DNS обнаружил ошибку в формате полученного пакета. Это может свидетельствовать об ошибке в команде **nslookup**.

Ping

С помощью отправки сообщений с эхо-запросом по протоколу ICMP проверяет соединение на уровне протокола IP с другим компьютером, поддерживающим TCP/IP. После каждой передачи выводится соответствующее сообщение с эхо-ответом. Ping – это основная TCP/IP-команда, используемая для устранения неполадки в соединении, проверки возможности доступа и разрешения имен. Команда **ping**, запущенная без параметров, выводит справку.

Синтаксис

ping [**-t**] [**-a**] [**-n** *счетчик*] [**-l** *размер*] [**-f**] [**-i** *TTL*] [**-v** *тип*] [**-r** *счетчик*] [**-s** *счетчик*] [{**-j** *список_узлов* | **-k** *список_узлов*}] [**-w** *интервал*] [*имя_конечного_компьютера*]

Параметры

-t – задает для команды ping отправку сообщений с эхо-запросом к точке назначения до тех пор, пока команда не будет прервана. Для прерывания команды и вывода статистики нажмите комбинацию CTRL-BREAK. Для прерывания команды ping и выхода из нее нажмите клавиши CTRL-C;

-a – задает разрешение обратного имени по IP-адресу назначения. В случае успешного выполнения выводится имя соответствующего узла;

-n *счетчик* – задает число отправляемых сообщений с эхо-запросом. По умолчанию — 4;

-l *размер* – задает длину (в байтах) поля данных в отправленных сообщениях с эхо-запросом. По умолчанию — 32 байта. Максимальный *размер* –65527;

-f – задает отправку сообщений с эхо-запросом с флагом «Don't Fragment» в IP-заголовке, установленном на 1. Сообщения с эхо-запросом не фрагментируются маршрутизаторами на пути к месту назначения. Этот параметр полезен для устранения проблем, возникающих с максимальным блоком данных для канала (Maximum Transmission Unit);

-i *TTL* – задает значение поля TTL в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию берется значение TTL, заданное по умолчанию для узла. Для узлов Windows XP это значение обычно равно 128. Максимальное значение *TTL* –255;

-v *тип* – задает значение поля типа службы (TOS) в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию это значение равно 0. *тип* – это десятичное значение от 0 до 255;

-r *счетчик* – задает параметр записи маршрута (Record Route) в IP-заголовке для записи пути, по которому проходит сообщение с эхо-запросом и соответствующее ему сообщение с эхо-ответом. Каждый переход в пути использует параметр записи маршрута. По возможности значение *счетчика* задается равным или большим, чем количество переходов между источником и местом назначения. Параметр *счетчик* имеет значение от 1 до 9;

-s *счетчик* – указывает вариант штампа времени Интернета (Internet Timestamp) в заголовке IP для записи времени прибытия сообщения с эхо-запросом и соответствующего ему сообщения с эхо-ответом для каждого перехода. Параметр *счетчик* имеет значение от 1 до 4;

-j *список_узлов* – указывает для сообщений с эхо-запросом использование параметра свободной маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в *списке_узлов*. При свободной маршрутизации последовательные промежуточные точки назначения могут быть разделены одним или несколькими маршрутизаторами. Максимальное число адресов или имен в списке узлов –9. Список узлов –это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами;

-k *список_узлов* – указывает для сообщений с эхо-запросом использование параметра строгой маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в *списке_узлов*. При строгой маршрутизации следующая промежуточная точка назначения должна быть доступной напрямую (она должна быть соседней в интерфейсе маршрутизатора). Максимальное число адресов или имен в списке узлов равно 9. Список узлов –это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами;

-w *интервал* – определяет в миллисекундах время ожидания получения сообщения с эхо-ответом, которое соответствует сообщению с эхо-запросом. Если сообщение с эхо-ответом не получено в пределах заданного интервала, то выдается сообщение об ошибке "Request timed out". Интервал по умолчанию равен 4000 (4 секунды);

имя_конечного_компьютера – задает точку назначения, идентифицированную IP-адресом или именем узла.

Заметки

Команда **ping** позволяет проверить имя и IP-адрес компьютера. Если проверка IP-адреса успешная, и проверка имени –нет, то имеет место проблема разрешения имен. В этом случае с помощью запросов DNS (Domain Name System) или с помощью методов разрешения имен NetBIOS проверьте, чтобы имя задаваемого компьютера было разрешено в локальном файле Hosts.

Примеры

Приведенный ниже пример (рисунок 1.1) содержит результаты работы команды **ping**:

```
C:\>ping example.microsoft.com
Pinging example.microsoft.com [192.168.239.132] with 32 bytes
of data:
Reply from 192.168.239.132: bytes=32 time=101ms TTL=124
Reply from 192.168.239.132: bytes=32 time=100ms TTL=124
Reply from 192.168.239.132: bytes=32 time=101ms TTL=124
Reply from 192.168.239.132: bytes=32 time=101ms TTL=124
```

Рисунок 1.1 – Результаты работы команды **ping**

Для отправки сообщения точке назначения 10.0.99.221 и сопоставления с ее узловым именем введите:

ping -a 10.0.99.221

Для отправки точке назначения 10.0.99.221 десяти сообщений с эхо-запросом, каждое из которых имеет поле данных из 1000 байт, введите:

ping -n 10 -l 1000 10.0.99.221

Для отправки сообщения точке назначения 10.0.99.221 и записи маршрута для 4 переходов введите:

ping -r 4 10.0.99.221

Для отправки сообщения точке назначения 10.0.99.221 и задания свободной маршрутизации для точек назначения 10.12.0.1-10.29.3.1-10.1.44.1 введите:

ping -j 10.12.0.1 10.29.3.1 10.1.44.1 10.0.99.221

Route

Выводит на экран и изменяет записи в локальной таблице IP-маршрутизации. Запущенная без параметров, команда **route** выводит справку.

Синтаксис

route [**-f**] [**-p**] [*команда* [*конечная_точка*] [**mask** *маска_сети*] [*шлюз*] [**metric** *метрика*]] [**if** *интерфейс*]]

Параметры

-f – очищает таблицу маршрутизации от всех записей, которые не являются узловыми маршрутами (маршруты с маской подсети 255.255.255.255), сетевым маршрутом замыкания на себя (маршруты с конечной точкой 127.0.0.0 и маской подсети 255.0.0.0) или маршрутом многоадресной рассылки (маршруты с конечной точкой 224.0.0.0 и маской подсети 240.0.0.0). При использовании данного параметра совместно с одной из команд (таких, как **add**, **change** или **delete**) таблица очищается перед выполнением команды;

-p – при использовании данного параметра с командой **add** указанный маршрут добавляется в реестр и используется для инициализации таблицы IP-маршрутизации каждый раз при запуске протокола TCP/IP. По умолчанию добавленные маршруты не сохраняются при запуске протокола TCP/IP. При использовании параметра с командой **print** выводит на экран список постоянных маршрутов. Все другие команды игнорируют этот параметр. Постоянные маршруты хранятся в реестре по адресу **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes**

команда – указывает команду, которая будет запущена на удаленной системе. В таблице 1.4 представлен список допустимых параметров.

Таблица 1.4 – Список допустимых параметров команды **route.**

Команда	Назначение
add	Добавление маршрута
change	Изменение существующего маршрута
delete	Удаление маршрута или маршрутов
print	Печать маршрута или маршрутов

конечная_точка – определяет конечную точку маршрута. Конечной точкой может быть сетевой IP-адрес (где разряды узла в сетевом адресе имеют значение 0), IP-адрес маршрута к узлу, или значение 0.0.0.0 для маршрута по умолчанию;

mask *маска_сети* – уазывает маску сети (также известной как маска подсети) в соответствии с точкой назначения. Маска сети может быть маской подсети соответствующей сетевому IP-адресу, например 255.255.255.255 для маршрута к узлу или 0.0.0.0. для маршрута по умолчанию. Если данный параметр пропущен, используется маска подсети 255.255.255.255. Конечная точка не может быть более точной, чем соответствующая маска подсети. Другими словами, значение разряда 1 в адресе конечной точки невозможно, если значение соответствующего разряда в маске подсети равно 0;

шлюз – указывает IP-адрес пересылки или следующего перехода, по которому доступен набор адресов, определенный конечной точкой и маской подсети. Для локально подключенных маршрутов подсети, адрес шлюза – это IP-адрес, назначенный интерфейсу, который подключен к подсети. Для удаленных маршрутов, которые доступны через один или несколько маршрутизаторов, адрес шлюза – непосредственно доступный IP-адрес ближайшего маршрутизатора;

metric *метрика* – задает целочисленную метрику стоимости маршрута (в пределах от 1 до 9999) для маршрута, которая используется при выборе в таблице маршрутизации одного из нескольких маршрутов, наиболее близко соответствующего адресу назначения пересылаемого пакета. Выбирается маршрут с наименьшей метрикой. Метрика отражает количество переходов, скорость прохождения пути, надежность пути, пропускную способность пути и средства администрирования;

if *интерфейс* – указывает индекс интерфейса, через который доступна точка назначения. Для вывода списка интерфейсов и их соответствующих индексов используйте команду **route print**. Значения индексов интерфейсов могут быть как десятичные, так и шестнадцатеричные. Перед шестнадцатеричными номерами вводится **0x**. В случае, когда параметр **if** пропущен, интерфейс определяется из адреса шлюза.

Заметки

- Большие значения в столбце **metric** таблицы маршрутизации – результат возможности протокола TCP/IP автоматически определять метрики маршрутов таблицы маршрутизации на основании конфигурации IP-адреса, маски подсети и стандартного шлюза для каждого интерфейса ЛВС. Автоматическое определение метрики интерфейса, включенное по умолчанию, устанавливает скорость каждого интерфейса и метрики маршрутов для каждого интерфейса так, что самый быстрый интерфейс создает маршруты с наименьшей метрикой. Чтобы удалить большие метрики, отключите автоматическое определение метрики интерфейса в дополнительных свойствах протокола TCP/IP для каждого подключения по локальной сети;
- имена могут использоваться для параметра *конечная_точка*, если существует соответствующая запись в файле базы данных Networks и папке *системный_корневой_каталог*\System32\Drivers\Etc. В параметре *шлюз* можно указывать имена до тех пор, пока они разрешаются в IP-адреса с помощью стандартных способов разрешения узлов, таких как запрос службы DNS, использование локального файла Hosts, находящегося в папке *системный_корневой_каталог*\system32\drivers\etc, или разрешение имен NetBIOS;
- если команда –**print** или **delete**, параметр *шлюз* опускается и используются подстановочные знаки для указания точки назначения и шлюза. Значение *конечной_точки* может быть подстановочным значением, которое указывается звездочкой (*). При наличии звездочки (*) или вопросительного знака (?) в описании конечной точки, они рассматриваются как подстановки, тогда печатаются или удаляются только маршруты, соответствующие точке назначения. Звездочка соответствует любой последовательности символов, а вопросительный знак –любому одному символу. 10.*.1, 192.168.*,

127.* и *224* являются допустимыми примерами использования звездочки в качестве подстановочного символа;

- при использовании недопустимой комбинации значений конечной точки и маски подсети (маски сети) выводится следующее сообщение об ошибке: «Маршрут: неверная маска подсети адреса шлюза». Ошибка появляется, когда одно или несколько значений разрядов в адресе конечной точки равно 1, а значения соответствующих разрядов маски подсети – 1. Для проверки этого состояния выразите конечную точку и маску подсети в двоичном формате. Маска подсети в двоичном формате состоит из последовательности единичных битов, представляющей часть сетевого адреса конечной точки, и последовательности нулевых битов, обозначающей часть адреса узла конечной точки. Проверьте наличие единичных битов в части адреса точки назначения, которая является адресом узла (как определено маской подсети).

Примеры

Чтобы вывести на экран все содержимое таблицы IP-маршрутизации, введите команду:

route print

Чтобы вывести на экран маршруты из таблицы IP-маршрутизации, которые начинаются с *10.*, введите команду:

route print 10.*

Чтобы добавить маршрут по умолчанию с адресом стандартного шлюза 192.168.12.1, введите команду:

route add 0.0.0.0 mask 0.0.0.0 192.168.12.1

Чтобы добавить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0 и следующим адресом перехода 10.27.0.1, введите команду:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1

Чтобы добавить постоянный маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0 и следующим адресом перехода 10.27.0.1, введите команду:

route -p add 10.41.0.0 mask 255.255.0.0 10.27.0.1

Чтобы добавить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0 и следующим адресом перехода 10.27.0.1 и метрикой стоимости 7, введите команду:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 metric 7

Чтобы добавить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0 и следующим адресом перехода 10.27.0.1 и использованием индекса интерфейса 0x3, введите команду:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 if 0x3

Чтобы удалить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0, введите команду:

route delete 10.41.0.0 mask 255.255.0.0

Чтобы удалить все маршруты из таблицы IP-маршрутизации, которые начинаются с *10.*, введите команду:

route delete 10.*

Чтобы изменить следующий адрес перехода для маршрута с конечной точкой 10.41.0.0 и маской подсети 255.255.0.0 с 10.27.0.1 на 10.27.0.25, введите команду:

route change 10.41.0.0 mask 255.255.0.0 10.27.0.25

Tracert

Определяет путь до точки назначения с помощью посылки в точку назначения эхо-сообщений протокола Control Message Protocol (ICMP) с постоянным увеличением значений срока жизни (Time to Live, TTL). Выведенный путь –это список ближайших интерфейсов маршрутизаторов, находящихся на пути между узлом источника и точкой назначения. Ближний интерфейс представляют собой интерфейс маршрутизатора, который является ближайшим к узлу отправителя на пути. Запущенная без параметров, команда **tracert** выводит справку.

Синтаксис

tracert [**-d**] [**-h** *максимальное_число_переходов*] [**-j** *список_узлов*] [**-w** *интервал*] [*имя_конечного_компьютера*]

Параметры

-d – предотвращает попытки команды **tracert** разрешения IP-адресов промежуточных маршрутизаторов в имена. Увеличивает скорость вывода результатов команды **tracert;**

-h *максимальное_число_переходов* – задает максимальное количество переходов на пути при поиске конечного объекта. Значение по умолчанию равно 30;

-j *список_узов* – указывает для сообщений с эхо-запросом использование параметра свободной маршрутизации в заголовке IP с набором промежуточных мест назначения, указанных в *списке_узлов*.

При свободной маршрутизации успешные промежуточные места назначения могут быть разделены одним или несколькими маршрутизаторами. Максимальное число адресов или имен в списке – 9. *Список_адресов* представляет набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами;

-w *интервал* – определяет в миллисекундах время ожидания для получения эхо-ответов протокола ICMP или ICMP-сообщений об истечении времени, соответствующих данному сообщению эхо-запроса. Если сообщение не получено в течение заданного времени, выводится звездочка (*). Таймаут по умолчанию 4000 (4 секунды);

имя_конечного_компьютера – задает точку назначения, указанную IP-адресом или именем узла.

Заметки

- Диагностическое средство, предназначенное для определения маршрута до точки назначения с помощью посылки в точку назначения эхо-запросов протокола Internet Control Message Protocol (ICMP) с различными значениями срока жизни (TTL, Time-To-Live). Каждый маршрутизатор, через который проходит путь, обязан перед дальнейшей пересылкой пакета уменьшить значение его поля TTL по меньшей мере на 1. Фактически, TTL –счетчик узлов. Предполагается, что когда параметр TTL становится равен 0, маршрутизатор посылает системе-источнику сообщение ICMP об истечении времени. Команда tracert определяет маршрут, посылая первый эхо-запрос с полем TTL, равным 1, и увеличивая значение этого поля на единицу для каждого последующего отправляемого эхо-пакета до тех пор, пока конечный узел не ответит или пока не будет достигнуто максимальное значение поля TTL. Максимальное количество переходов по умолчанию равно 30 и может быть изменено с помощью параметра **-h**. Путь определяется из анализа сообщений ICMP об истечении времени, полученных от промежуточных маршрутизаторов, и это-ответов точки назначения. Однако некоторые маршрутизаторы не посылают сообщений об истечении времени для пакетов с нулевыми значениями TTL и не видны для команды tracert. В этом случае для перехода отображается ряд звездочек (*);
- чтобы выполнить трассировку маршрута, вывести значение задержки распространения по сети и потерь пакета на каждом маршрутизаторе и узле в пути, используйте команду **pathping.**

Примеры

Чтобы выполнить трассировку пути к узлу corp7.microsoft.com, введите команду:

tracert corp7.microsoft.com

Чтобы выполнить трассировку пути к узлу corp7.microsoft.com и предотвратить разрешение каждого IP-адреса в имя, введите:

tracert -d corp7.microsoft.com

Чтобы выполнить трассировку пути к узлу corp7.microsoft.com и использовать узлы 10.12.0.1-10.29.3.1-10.1.44.1 для свободной маршрутизации, введите следующую команду:

tracert -j 10.12.0.1 10.29.3.1 10.1.44.1 corp7.microsoft.com

Net session

Служит для управления подключениями к серверу. Команда **net session** без параметров выводит сведения обо всех сеансах локального компьютера.

Синтаксис

net session [*имя_компьютера*] [**/delete**]

Параметры

имя_компьютера – имя компьютера, сеансы которого требуется просмотреть или отключить;

/delete – завершение сеанса с *компьютером* и закрытие всех открытых файлов данного сеанса. Если *имя_компьютера* не задано, закрываются все сеансы на локальном компьютере;

net help *команда* – отображение справки для указанной команды **net**.

Внимание!

- Использование команды **net session** может привести к потере данных. Рекомендуется уведомлять пользователей перед принудительным завершением сеанса.

Заметки

- Для вызова команды **net session** также можно использовать синтаксис **net sessions** или **net sess;**
- Команда **net session** служит для вывода имен пользователей и компьютеров, имеющих доступ к серверу, со сведениями об открытых файлах и о времени холостого хода сеанса.

Эти сведения выводятся в формате, представленном на рисунке 1.2.

```
Компьютер      Пользователь      Тип клиента  Ожидание открытия
-------------------------------------------------------------
\\BASSETT      CHRISDR           Windows 2000   1      00:00:13
\\SHARONCA     Администратор     DOS LM 2.1     0      01:05:13
```

Рисунок 1.2 – Формат вывода команды **net session**

• Чтобы вывести сведения о сеансе одного пользователя, задайте *имя_компьютера*. Сведения об одном пользователе включают список общих ресурсов, к которым подключен пользователь;

• запись о сеансе появляется, когда пользователь компьютера-клиента успешно соединяется с сервером. Успешный сеанс возможен в случае, если два компьютера находятся в одной сети, а имя и пароль пользователя приняты сервером. Прежде чем клиент сможет использовать ресурсы сервера, он должен установить сеанс с сервером. Сеанс будет длиться до тех пор, пока пользователь подключен к ресурсу. Клиент и сервер могут иметь только один сеанс, однако допускается несколько подключений к ресурсам;

• чтобы задать время простоя сеанса до автоматического отключения, включите режим **автоматического отключения**, используя команду **net config server /autodisconnect**. Для получения дополнительных сведений о команде **net config server** щелкните ссылку «См. также». Автоматическое отключение незаметно для пользователя, поскольку сеанс автоматически восстанавливается, когда пользователь снова обращается к ресурсу;

• чтобы завершить сеанс с сервером, введите команду **net session** *имя_компьютера* **/delete**.

Примеры

Чтобы вывести сведения о сеансе для локального сервера, введите:

net session

Чтобы вывести сведения о сеансе для клиента с компьютера Shepherd, введите:

net session \\shepherd

Чтобы завершить все сеансы между сервером и подключенными к нему клиентами, введите:

net session /delete

Net share

Управление общими ресурсами. При вызове команды **net share** без параметров выводятся сведения обо всех общих ресурсах локального компьютера.

Синтаксис

net share [*имя_ресурса*] **net share** [*имя_ресурса*=*диск*:*путь* [{**/users:***число*|**/unlimited**}] [**/remark:"***текст***"**] [**/cache:** {**manual**|**automatic**|**no**}]]**net share** [*имя_ресурса* [{**/users:***число*|**unlimited**}] [**/remark:"***текст***"**] [**/cache:** {**manual**|**automatic**|**no**}]] **net share** [{*имя_ресурса*|*диск*:*путь*} **/delete**]

Параметры

имя_ресурса – сетевое имя общего ресурса. Команда **net share** *имя_ресурса* выводит сведения об отдельном ресурсе;

диск:*путь* – абсолютный путь к папке, которую требуется сделать общей;

/users:*число* – максимальное количество пользователей, которым разрешен одновременный доступ к общему ресурсу;

/unlimited – отмена ограничения на число пользователей, которым разрешен одновременный доступ к общему ресурсу;

/remark:"*текст***"** – добавление описательного комментария к ресурсу. Текст следует заключать в кавычки;

/cache:automatic – включение автономного кэширования клиентов с автоматической реинтеграцией;

/cache:manual – включение автономного кэширования клиентов с реинтеграцией вручную;

/cache:no – оповещение клиента о невозможности автономного кэширования;

/delete – тмена общего доступа к ресурсу;

net help *команда* – отображение справки для указанной команды **net**.

Заметки

- Чтобы предоставить общий доступ к папке, имя которой содержит пробелы, заключите диск и путь к папке в кавычки (например **"C:\Новая папка"**);
- при запросе списка всех общих ресурсов компьютера выводятся (Рисунок 1.3): имя общего ресурса, имена устройств или путь, связанный с устройством, а также комментарий к этому ресурсу.

```
-------------------------------------------------------
ADMIN$      C:\WINNT                        Удаленный Admin
Общее имя   Ресурс                          Заметки
C$          C:\                             Стандартный общий ресурс
print$      C:\WINNT\SYSTEM\SPOOL
IPC$                                        Удаленный IPC
LASER       LPT1                Очередь     Лазерный принтер
```

Рисунок 1.3 – Формат вывода команды **net share**

- когда общий ресурс создается на сервере, его конфигурация сохраняется. После остановки службы «Сервер» все общие ресурсы отключаются, но после следующего запуска службы «Сервер» они будут восстановлены. Дополнительные сведения о службах содержатся в разделе Службы;
- имена общих ресурсов, заканчивающиеся знаком **$**, не отображаются при обзоре локального компьютера с удаленного компьютера.

Примеры

Чтобы вывести сведения об общих ресурсах компьютера, введите:

net share

Чтобы сделать папку «C:\Данные» общим ресурсом Данные и включить примечание к нему, введите:

net share ОбщиеДанные=c:\Данные /remark:"Для отдела 123"

Чтобы отменить общий доступ к ресурсу ОбщиеДанные, созданному в предыдущем примере, введите:

net share ОбщиеДанные /delete

Чтобы сделать папку «C:\Список рисунков» общим ресурсом Список, введите:

net share Список="c:\Список рисунков"

Net use

Подключение к общим сетевым ресурсам или вывод информации о подключениях компьютера. Команда также управляет постоянными сетевыми соединениями. Вызванная без параметров, команда **net use** извлекает список сетевых подключений.

Синтаксис

net use [{*имя_устройства* | ***}] [*имя_компьютера**ресурс*[*том*]] [{*пароль* | ***}]] [**/user:**[*имя_домена*\]] [**/user:**[*имя_домена_с_точкой*\]*имя_пользователя*] [**/user:**[*имя_пользователя*@*имя_домена_с_точкой*] [**/savecred**] [**/smartcard**] [{**/delete** | **/persistent:**{**yes** | **no**}}]

net use [*имя_устройства* [**/home**[{*пароль* | ***}] [**/delete:**{**yes** | **no**}]]

net use [**/persistent:**{**yes** | **no**}]

Параметры

Имя_устройства – задает имя ресурса при подключении или имя устройства при отключении. Существует два вида имен устройств: имена для дисковых устройств (то есть, диски с буквенными обозначениями от D: до Z:) и для принтеров (соответственно, от LPT1: до LPT3:). Ввод звездочки (*) вместо имени определенного устройства обеспечит присвоение такому устройству ближайшего доступного имени;

*имя_компьютера**имя_ресурса* –указывает имя сервера и общего ресурса. Если параметр *имя_компьютера* содержит пробелы, все имя компьютера от двойной обратной черты (\\) до конца (например, "*Computer Name**Share Name*") должно быть заключено в прямые кавычки ("). Имя компьютера может иметь длину от 1 до 15 знаков;

том – задает имя тома системы NetWare. Для подключения к серверам Netware необходимо установить и запустить клиент для сетей NetWare;

пароль – задает пароль, необходимый для подключения к общему ресурсу. Введите звездочку (*) для вывода приглашения на ввод пароля. При вводе с клавиатуры символы пароля не выводятся на экран;

/user – задает другое имя пользователя для подключения к общему ресурсу;

имя_домена – задает имя другого домена. Пропуск параметра *имя_домена* приводит к тому, что команда **net use** использует имя домена, заданное при входе в систему;

имя_пользователя – указывает имя пользователя для подключения;

имя_домена _с_точкой –указывает полное имя домена, в котором присутствует учетная запись пользователя;

/savecred – сохраняет введенные учётные данные для дальнейшего использования;

/smartcard – указывает необходимость считывания учетных данных со смарт-карты для сетевого подключения. При наличии нескольких смарт-карт появится запрос на указание одной из них;

/delete – отменяет указанное сетевое подключение. Если подключение задано с символом звездочки (*), будут отменены все сетевые подключения;

/persistent:{**yes** | **no**} – управляет постоянными сетевыми подключениями. По умолчанию берется последнее использованное значение. Подключения без устройства не являются постоянными. Выбор значения **Yes** приводит к сохранению всех существующих соединений и восстанавлению их при следующем подключении. При выборе значения **No** выполняемые и последующие подключения не сохраняются. Существующие подключения восстанавливаются при следующем входе в систему. Для удаления постоянных подключений используется ключ **/delete;**

/home – подключает пользователя к его основному каталогу;

net help *команда* – отображение справки для указанной команды **net**.

Заметки

- Подключение и отключение от сетевого ресурса. Команда **net use** используется для подключения и отключения от сетевых ресурсов и для вывода сведений о текущих подключениях к таким ресурсам. Если сетевой ресурс является текущим диском или его использует какое-либо работающее приложение, отключиться от такого ресурса невозможно;
- просмотр сведений о подключениях. Чтобы просмотреть сведения о подключении, можно использовать любой из следующих способов:
  - введите команду **net use** *имя_устройства* для получения сведений о конкретном подключении;
  - введите команду **net use** для получения списка всех подключений компьютера.
- использование подключений без устройств. Подключения без устройств не являются постоянными;

• подключение к серверам NetWare. Установка и запуск клиента для сетей NetWare дает возможность подключаться к серверам NetWare или сети Novell. При этом используется тот же синтаксис, что и при подключении к серверам сети Windows, с добавлением имени тома для подключения;

• использование кавычек. Если вводимое *имя_сервера* содержит пробелы, его следует заключать в кавычки (т. е. **"***имя_сервера***"**). Пропуск кавычек влечет за собой появление сообщения об ошибке.

Примеры

Чтобы назначить относящееся к дисковому устройству имя E: общему каталогу Letters на сервере \\Financial, следует ввести:

net use e: \\financial\letters

Чтобы назначить относящееся к дисковому устройству имя M: каталогу Mike тома Letters на сервере \\Financial Netware, следует ввести:

net use m: \\financial\letters\mike

Чтобы подключить пользователя с идентификатором Dan так, как если бы он подключался из домена Accounts, следует ввести:

net use d:\\server\share /user:Accounts\Dan

Для отключения от каталога \\Financial\Public служит команда:

net use f: \\financial\public /delete

Для подключения к совместно используемым запискам ресурса на сервере \\Financial 2 служит команда:

net use k: "\\financial 2" \memos

Для восстановления текущих подключений при следующих входах в сеть, независимо от будущих изменений, служит команда:

net use /persistent:yes

Net view

Выводит список доменов, компьютеров или общих ресурсов на данном компьютере. Вызванная без параметров, команда **net view** выводит список компьютеров в текущем домене.

Синтаксис

net view [*имя_компьютера*] [**/domain**[**:***имя_домена*]]

net view /network:nw [*имя_компьютера*]

Параметры

имя_компьютера – задает имя компьютера для просмотра расположенных на нем общих ресурсов;

/domain[**:***имя_домена*] – задает домен, для которого выводится список компьютеров. Если параметр *имя_домена* не задан, команда выводит список всех доменов сети;

/network:nw – выводит список всех доступных серверов сети NetWare. При указании имени компьютера команда отображает все доступные ресурсы на данном компьютере. Кроме того, можно указать дополнительные сети;

net help *команда* – отображение справки для указанной команды **net**.

Заметки

Команда **net view** выводит список компьютеров.

Примеры

Список общих ресурсов компьютера \\Production может быть получен с помощью команды:

net view \\production

Для просмотра ресурсов сервера NetWare с именем \\Marketing служит команда:

net view /network:nw \\marketing

Для вывода списка компьютеров в домене или рабочей группе sales служит команда:

net view /domain:sales

Для вывода списка всех серверов в сети NetWare можно использовать следующую команду:

net view /network:nw

1.2 Задание на лабораторную работу

Для выполнения лабораторной работы необходимо:

1. Изучить основные команды ОС Windows для работы с сетевыми ресурсами
2. Научиться определять IP адрес компьютера и сетевые настройки в ОС Windows.
3. Научиться проверять наличие соединения с удалённым узлом. Научиться определять по имени компьютера его IP-адрес
4. Научиться осуществлять мониторинг использования сети и анализ сетевого взаимодействия в ОС Windows

1.3 Индивидуальные задания

Необходимо написать файл сценариев в ОС Windows, осуществляющий решение задачи согласно варианта в таблице 1.5. Результаты решения сохранить в текстовый файл.

Отчёт должен содержать описание всех изученных команд и подробное описание действий для выполнения п.1-5, листинги решения индивидуального задания и результаты проведённой верификации. Результаты выполнения лабораторной работы должны быть **обязательно** продемонстрированы на компьютере.

1.4 Контрольные вопросы

1. Команды ОС Windows для работы с сетевыми ресурсами. Основные параметры.
2. Проверка наличия соединения с удалённым узлом.
3. Программные средства мониторинга и анализа использования сети в ОС Windows.

Таблица 1.5 – Варианты задач

№ варианта	Условие задачи
1	Определить IP-адрес компьютеров, с заданной маской имени.
2	Посчитать количество компьютеров, видимых с данного компьютера.
3	Определить компьютер в сети, скорость взаимодействия с котором наибольшая.
4	Определить компьютер в сети, до которого самый длинный маршрут
5	Определить IP-адреса всех компьютеров, связь с которыми осуществляется через указанный шлюз
6	Определить компьютеры, имеющие более одного IP-адреса
7	Найти компьютер в сети, скорость взаимодействия с которым наименьшая
8	Сформировать список всех доступных сетевых ресурсов в заданном сегменте
9	Определить IP-адреса всех доступных DHCP-серверов
10	Подключить все доступные сетевые ресурсы из указанного списка компьютеров

11	Определить IP-адреса всех доступных DNS-серверов
12	Определить IP-адреса всех доступных WINS-серверов
13	Определить MAC-адреса компьютеров, из указанного списка (задан ip адрес)
14	Определить компьютер в сети, скорость взаимодействия с котором наименьшая.
15	Определить MAC-адреса всех доступных DHCP-серверов
16	Определить MAC-адреса всех доступных DNS-серверов
17	Определить MAC-адреса всех доступных WINS-серверов
18	Сформировать список имен компьютеров в заданном сегменте
19	Определить ip адреса компьютеров установивших подключения с данным компьютером
20	Подключить все доступные сетевые ресурсы со всех компьютеров, установивших подключения с данным компьютером
21	Определить самый короткий участок на пути к указанному узлу
22	Определить IP-адреса всех компьютеров, связь с которыми осуществляется через шлюз по умолчанию
23	Вывести список доступных сетевых ресурсов со всех компьютеров, установивших подключения с данным компьютером
24	Посчитать количество компьютеров, видимых через шлюз по умолчанию.
25	Определить количество маршрутизаторов на пути к указанному узлу
26	Вывести ip адреса маршрутизаторов на пути к указанному узлу, отсортированных по возрастанию времени задержки.
27	Определить скорости доступа к компьютерам из списка по ip адресам
28	Определить имя домена в котором находиться данный компьютер
29	Определить самый длинный участок на пути к указанному узлу
30	Определить компьютеры, не имеющие имен

2 Лабораторная работа №2. Сети Ethernet: используемое оборудование. Топологии сетей

2.1 Теоретические сведения. Описание программы моделирования работы компьютерных сетей *Network Emulator*

Возможности и используемые технологии ***Network Emulator (NE)*:** маршрутизация, система моделирования каналов, IP-фильтрация (также в формате для роутеров Cisco), типы пакетов: ICMP, UDP, TCP, низкоуровневые ARP запросы, концепция интерфейсов, концепция сокетов (простой, дейтаграммный и потоковый), эмуляция хостов, свитчей и хабов, процессы: traceroute, talkd, talk, echoer, gated (с BGP), уровень помех на канале, система демонстрации сцены, возможность связывания нескольких ***NE*** через реальную сеть TCP/IP.

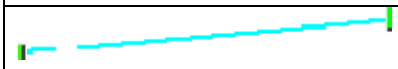
NE разрабатывалась как чисто визуальная среда создания виртуальных IP сетей путем сборки их из виртуальных компьютеров с виртуальными интерфейсами и виртуальными каналами связи между ними. Основным средством манипулирования объектами являются контекстные меню, появляющиеся при нажатии на них правой клавиши мыши.

Визуальные объекты ***NE*** представлены в таблице 2.1.

Интерфейсы здесь трех типов: Точка-Точка (point-to-point), Ethernet (или любая среда широковещания) и виртуальный интерфейс loopback, именуемый "заглушка". Так как Ethernet строится на коаксиале или на витой паре, то интерфейсы поддерживают один или несколько типов, и нужно при настройке указать, какой именно тип активен.

Хост – это компьютер, подключенный к сети. Предположим, мы имеем настоящий компьютер. В него можно, в принципе, установить до пяти сетевых карт или модемов. Плюс к ним добавить виртуальные интерфейсы. Получаем, что для среднего компьютера подходит ограничение в 8 интерфейсов. Поэтому, в ***NE*** на хост положено ограничение по количеству интерфейсов: 1 <= Ifaces <= 8. Далее. На хосте можно запускать программы. Ввиду редкого запуска программ, максимальное работающее их количество на хосте составляет 5 штук. Максимальное количество сокетов – тоже в пределах 5-10.

Таблица 2.1 – Визуальные объекты NE

Изображение объекта	Функция объекта
	хост – компьютер или сервер с сетевой картой или сетевыми картами (интерфейсами)
	свитч (коммуникационное устройство)
	хаб (коммуникационное устройство)
	интерфейс (сетевая карта)
	канал связи (универсальный)
	коаксиальный кабель Ethernet – отрезок коаксиального кабеля с терминаторами на его концах. К нему могут подключаться до десяти каналов связи. В реальных примерах таким каналом связи можно считать сочленение между коаксиальным кабелем и сетевой картой.

Свитчи и хабы имеют ограничение на количество интерфейсов 1 <= Ifaces <= 24. Свитчи могут иметь несколько специальных интерфейсов (Module), через которые происходит связь между самими свитчами. Все интерфейсы свитчей поддерживают только витую пару. В отличие от свитчей, хабы имеют один (единственный) интерфейс (первый, сверху слева), поддерживающий коаксиал. Этот интерфейс служит хабу для связи с общим кабелем Ethernet.

Коаксиальный кабель Ethernet также является объектом в структуре ***NE***, так как он все же выполняет некоторый набор действий. Кабель Ethernet – это просто кусок коаксиального кабеля с терминаторами на его концах. К нему могут подключаться до десяти каналов связи. В реальных примерах таким каналом связи можно считать сочленение между коаксиальным кабелем и сетевой картой.

Канал связи – универсальный. Он связывает всех со всеми, при условии, что в реальности разъемы объектов совместимы. Например,

можно соединить первый интерфейс хаба (BNC), поддерживающий соединение через коаксиал, с кабелем Ethernet (так как он коаксиальный). Но нельзя соединить, например, любой интерфейс свича (UTP) с кабелем Ethernet, или интерфейс Ethernet с интерфейсом Точка-Точка.

Интерфейс к удаленной сети. На него положено ограничение по использованию портов: от 8300 до 8400. Интерфейс подключается только к интерфейсу Точка-Точка.

В примере, приведенном ниже (рисунок 2.1), IP-адрес сетевой карты компьютер – 10.180.0.4, а маска подсети требуемого вида (255.255.255.0) определяется путем задания значения после наклонной черты, равного 24.

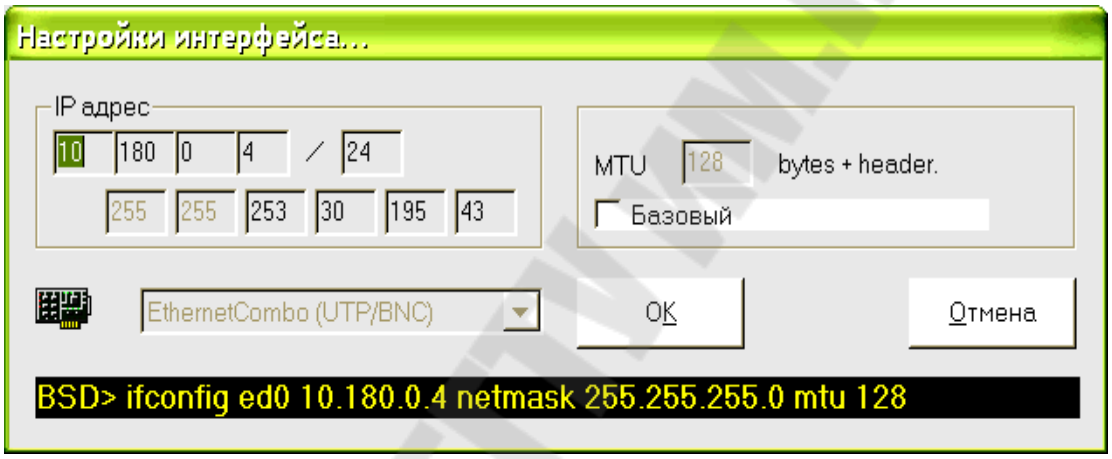


Рисунок 2.1 –Пример задания сетевой карты хоста

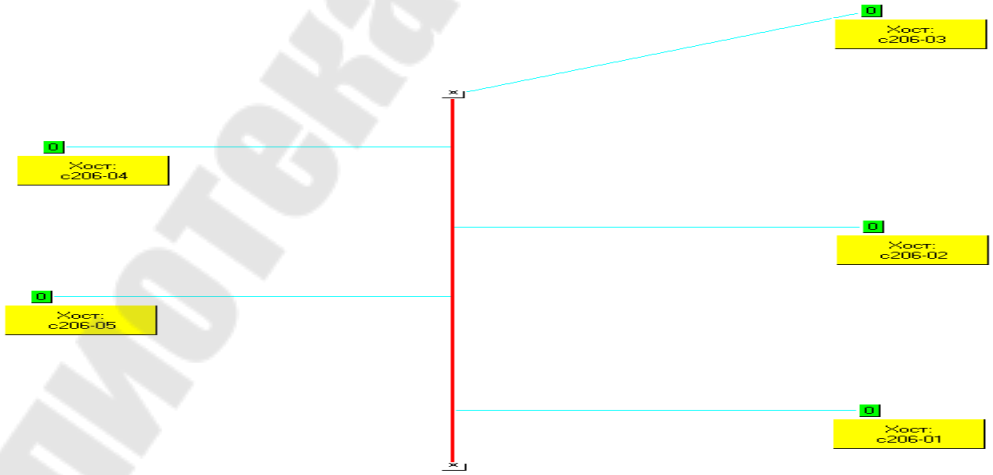


Рисунок 2.2 – Пример реализации модели локальной вычислительной сети с топологией типа "общая шина"

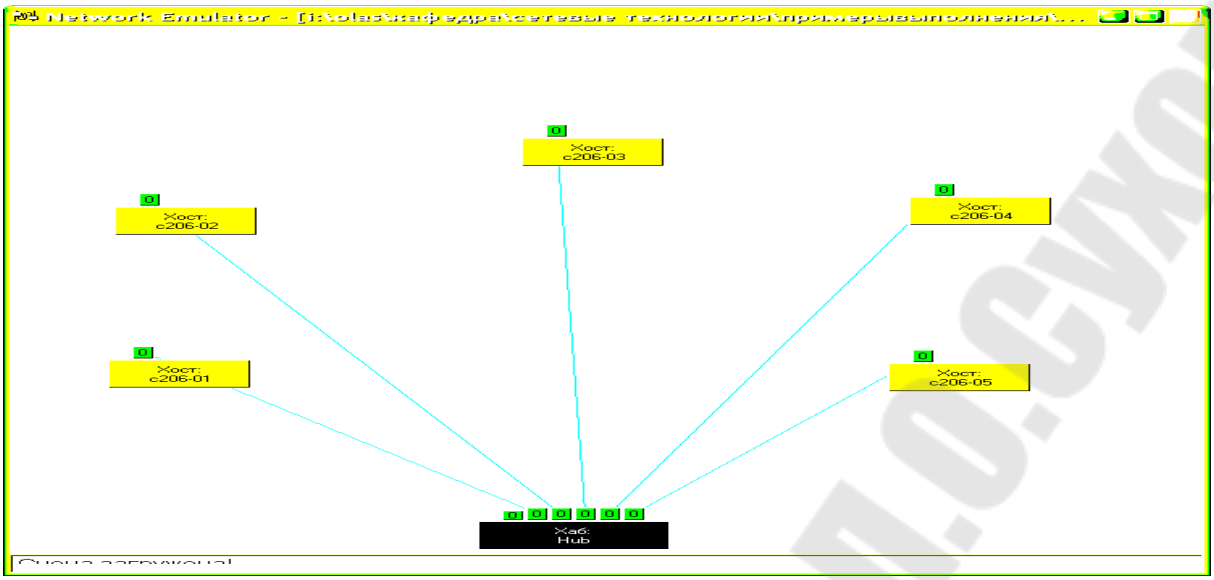


Рисунок 2.3 – Пример реализации модели локальной вычислительной сети с топологией типа "звезда":

На рисунке 2.2 показан пример реализации модели локальной вычислительной сети с топологией типа "общая шина", а на рисунке 2.3 – с топологией типа "звезда":

Проверка правильности функционирования созданной модели осуществляется с помощью функции **Быстрое создание пакета** (выбирается из контекстного меню, которое вызывается при нажатии на правую кнопку мыши). При создании IP-пакета соответствующее сообщение посылается от одного выделенного компьютера к другому. Сообщение этого IP-пакета изображается сначала в виде движущихся желтых шариков (установка связи), а затем в виде белых шариков (прохождение самого сообщения) Если пакеты распространяются между выбранными компьютерами и во всей сети (рисунок 2.4), то модель сети корректна.

2.2 Задание на лабораторную работу

Для выполнения лабораторной работы необходимо:

1. смоделировать в NetWork Emulator'e несколько вариантов сети Ethernet с использованием различного сетевого оборудования: коаксиальный кабель, концентратор, коммутируемый концентратор;
2. смоделировать несколько вариантов сети Ethernet с совместным использованием различного сетевого оборудования: коаксиальный кабель, концентратор, коммутируемый концентратор;

3. изучить основные принципы работы используемого оборудования, его особенности и различия (тезисно описать в отчёте по лабораторной работе).

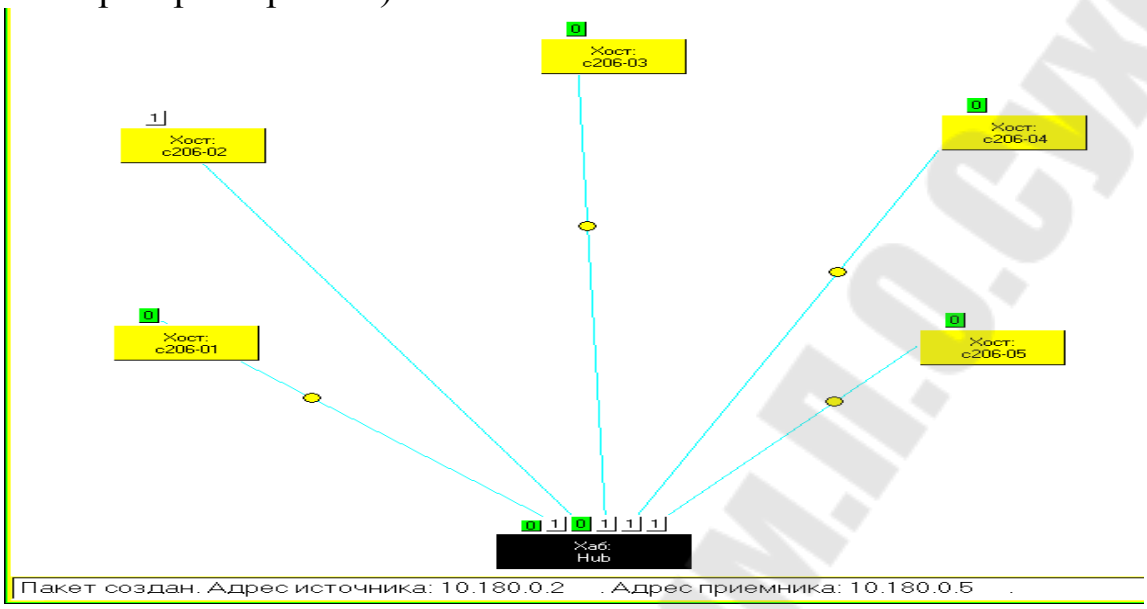


Рисунок 2.4 – Пример прохождения пакетов посети

Отчёт должен содержать схемы моделируемых ЛВС с указанием адресов компьютеров в сети и подробное описание функционирования ЛВС.

Результаты моделирования обязательно должны быть продемонстрированы на компьютере.

2.3 Контрольные вопросы

1. Что такое и для чего предназначена сетевая карта?
2. Что такое и для чего предназначен хаб?
3. Что такое и для чего предназначен свитч?
4. Какие кабельные среды передачи данных вы знаете?
5. Охарактеризуйте топологию типа "звезда".
6. Охарактеризуйте топологию типа "общая шина".
7. Какова структура IP адреса. Что такое маска подсети?
8. Что такое MAC адрес?
9. Опишите алгоритм передачи данных между двумя компьютерами по протоколу TCP/IP. Для чего предназначен ARP протокол?

3 **Лабораторная работа №3. Сетевое оборудование Ethernet**

3.1 Теоретические сведения

Сетевые адаптеры

Функции и характеристики сетевых адаптеров

Сетевой адаптер (Network Interface Card, NIC) вместе со своим драйвером реализует второй, канальный уровень модели открытых систем в конечном узле сети -компьютере. Более точно, в сетевой операционной системе пара адаптер и драйвер выполняет только функции физического и MAC-уровней, в то время как LLC-уровень обычно реализуется модулем операционной системы, единым для всех драйверов и сетевых адаптеров. Собственно так оно и должно быть в соответствии с моделью стека протоколов IEEE 802. Например, в ОС Windows NT уровень LLC реализуется в модуле NDIS, общем для всех драйверов сетевых адаптеров, независимо от того, какую технологию поддерживает драйвер.

Сетевой адаптер совместно с драйвером выполняют две операции: передачу и прием кадра.

Передача кадра из компьютера в кабель состоит из перечисленных ниже этапов (некоторые могут отсутствовать, в зависимости от принятых методов кодирования).

- Прием кадра данных LLC через межуровневый интерфейс вместе с адресной информацией MAC-уровня. Обычно взаимодействие между протоколами внутри компьютера происходит через буферы, расположенные в оперативной памяти. Данные для передачи в сеть помещаются в эти буферы протоколами верхних уровней, которые извлекают их из дисковой памяти либо из файлового кэша с помощью подсистемы ввода/вывода операционной системы;
- оформление кадра данных MAC-уровня, в который инкапсулируется кадр LLC (с отброшенными флагами 01111110). Заполнение адресов назначения и источника, вычисление контрольной суммы;
- формирование символов кодов при использовании избыточных кодов типа 4B/5B. Скрэмблирование кодов для получения более равномерного спектра сигналов. Этот этап

используется не во всех протоколах – например, технология Ethernet 10 Мбит/с обходится без него;

- выдача сигналов в кабель в соответствии с принятым линейным кодом – манчестерским, NRZI, MLT-3 и т. п. Прием кадра из кабеля в компьютер включает следующие действия;
- прием из кабеля сигналов, кодирующих битовый поток;
- выделение сигналов на фоне шума. Эту операцию могут выполнять различные специализированные микросхемы или сигнальные процессоры DSP. В результате в приемнике адаптера образуется некоторая битовая последовательность, с большой степенью вероятности совпадающая с той, которая была послана передатчиком;
- если данные перед отправкой в кабель подвергались скрэмблированию, то они пропускаются через дескрэмблер, после чего в адаптере восстанавливаются символы кода, посланные передатчиком;
- проверка контрольной суммы кадра. Если она неверна, то кадр отбрасывается, а через межуровневый интерфейс наверх, протоколу LLC передается соответствующий код ошибки. Если контрольная сумма верна, то из MAC – кадра извлекается кадр LLC и передается через межуровневый интерфейс наверх, протоколу LLC. Кадр LLC помещается в буфер оперативной памяти.

Распределение обязанностей между сетевым адаптером и его драйвером стандартами не определяется, поэтому каждый производитель решает этот вопрос самостоятельно. Обычно сетевые адаптеры делятся на адаптеры для клиентских компьютеров и адаптеры для серверов.

В адаптерах для клиентских компьютеров значительная часть работы перекладывается на драйвер, тем самым адаптер оказывается проще и дешевле. Недостатком такого подхода является высокая степень загрузки центрального процессора компьютера рутинными работами по передаче кадров из оперативной памяти компьютера в сеть. Центральный процессор вынужден заниматься этой работой вместо выполнения прикладных задач пользователя.

Поэтому адаптеры, предназначенные для серверов, обычно снабжаются собственными процессорами, которые самостоятельно выполняют большую часть работы по передаче кадров из оперативной памяти в сеть и в обратном направлении. Примером

такого адаптера может служить сетевой адаптер SMS EtherPower со встроенным процессором Intel i960.

В зависимости от того, какой протокол реализует адаптер, адаптеры делятся на Ethernet-адаптеры, Token Ring-адаптеры, FDDI-адаптеры и т. д. Так как протокол Fast Ethernet позволяет за счет процедуры автопереговоров автоматически выбрать скорость работы сетевого адаптера в зависимости от возможностей концентратора, то многие адаптеры Ethernet сегодня поддерживают две скорости работы и имеют в своем названии приставку 10/100. Это свойство некоторые производители называют авточувствительностью.

Сетевой адаптер перед установкой в компьютер необходимо конфигурировать. При конфигурировании адаптера обычно задаются номер прерывания IRQ, используемого адаптером, номер канала прямого доступа к памяти DMA (если адаптер поддерживает режим DMA) и базовый адрес портов ввода/вывода.

Если сетевой адаптер, аппаратура компьютера и операционная система поддерживают стандарт Plug-and-Play, то конфигурирование адаптера и его драйвера осуществляется автоматически. В противном случае нужно сначала сконфигурировать сетевой адаптер, а затем повторить параметры его конфигурации для драйвера. В общем случае, детали процедуры конфигурирования сетевого адаптера и его драйвера во многом зависят от производителя адаптера, а также от возможностей шины, для которой разработан адаптер.

Классификация сетевых адаптеров

В качестве примера классификации адаптеров используем подход фирмы 3Com, имеющей репутацию лидера в области адаптеров Ethernet. Фирма 3Com считает, что сетевые адаптеры Ethernet прошли в своем развитии три поколения.

Адаптеры первого поколения были выполнены на дискретных логических микросхемах, в результате чего обладали низкой надежностью. Они имели буферную память только на один кадр, что приводило к низкой производительности адаптера, так как все кадры передавались из компьютера в сеть или из сети в компьютер последовательно. Кроме этого, задание конфигурации адаптера первого поколения происходило вручную, с помощью перемычек. Для каждого типа адаптеров использовался свой драйвер, причем интерфейс между драйвером и сетевой операционной системой не был стандартизирован.

В сетевых адаптерах второго поколения для повышения производительности стали применять метод многокадровой буферизации. При этом следующий кадр загружается из памяти компьютера в буфер адаптера одновременно с передачей предыдущего кадра в сеть. В режиме приема, после того как адаптер полностью принял один кадр, он может начать передавать этот кадр из буфера в память компьютера одновременно с приемом другого кадра из сети.

В сетевых адаптерах второго поколения широко используются микросхемы с высокой степенью интеграции, что повышает надежность адаптеров. Кроме того, драйверы этих адаптеров основаны на стандартных спецификациях. Адаптеры второго поколения обычно поставляются с драйверами, работающими как в стандарте NDIS (спецификация интерфейса сетевого драйвера), разработанном фирмами 3Com и Microsoft и одобренном IBM, так и в стандарте ODI (интерфейс открытого драйвера), разработанном фирмой Novell.

В сетевых адаптерах третьего поколения (к ним фирма 3Com относит свои адаптеры семейства EtherLink III) осуществляется конвейерная схема обработки кадров. Она заключается в том, что процессы приема кадра из оперативной памяти компьютера и передачи его в сеть совмещаются во времени. Таким образом, после приема нескольких первых байт кадра начинается их передача. Это существенно (на 25-55 %) повышает производительность цепочки оперативная память – адаптер – физический канал – адаптер – оперативная память. Такая схема очень чувствительна к порогу начала передачи, то есть к количеству байт кадра, которое загружается в буфер адаптера перед началом передачи в сеть. Сетевой адаптер третьего поколения осуществляет самонастройку этого параметра путем анализа рабочей среды, а также методом расчета, без участия администратора сети. Самонастройка обеспечивает максимально возможную производительность для конкретного сочетания производительности внутренней шины компьютера, его системы прерываний и системы прямого доступа к памяти.

Адаптеры третьего поколения базируются на специализированных интегральных схемах (ASIC), что повышает производительность и надежность адаптера при одновременном снижении его стоимости. Компания 3Com назвала свою технологию

конвейерной обработки кадров Parallel Tasking, другие компании также реализовали похожие схемы в своих адаптерах. Повышение производительности канала «адаптер-память» очень важно для повышения производительности сети в целом, так как производительность сложного маршрута обработки кадров, включающего, например, концентраторы, коммутаторы, маршрутизаторы, глобальные каналы связи и т. п., всегда определяется производительностью самого медленного элемента этого маршрута. Следовательно, если сетевой адаптер сервера или клиентского компьютера работает медленно, никакие быстрые коммутаторы не смогут повысить скорость работы сети.

Выпускаемые сегодня сетевые адаптеры можно отнести к четвертому поколению. В эти адаптеры обязательно входит ASIC, выполняющая функции MAC-уровня, а также большое количество высокоуровневых функций. В набор таких функций может входить поддержка агента удаленного мониторинга RMON, схема приоритезации кадров, функции дистанционного управления компьютером и т. п. В серверных вариантах адаптеров почти обязательно наличие мощного процессора, разгружающего центральный процессор. Примером сетевого адаптера четвертого поколения может служить адаптер компании 3Com Fast EtherLink XL 10/100.

Концентраторы

Характеристики сетевых концентраторов

- *Количество портов* – разъёмов для подключения сетевых линий, обычно выпускаются концентраторы с 4, 5, 6, 8, 16, 24 и 48 портами (наиболее популярны с 4, 8 и 16). Концентраторы с большим количеством портов значительно дороже. Однако концентраторы можно соединять каскадно друг к другу, наращивая количество портов сегмента сети. В некоторых для этого предусмотрены специальные порты;
- *скорость передачи данных* – измеряется в Мбит/с, выпускаются концентраторы со скоростью 10, 100 и 1000. Кроме того, в основном распространены концентраторы с возможностью изменения скорости, обозначаются как 10/100/1000 Мбит/с. Скорость может переключаться как автоматически, так и с помощью перемычек или переключателей. Обычно, если хотя бы одно устройство

присоединено к концентратору на скорости нижнего диапазона, он будет передавать данные на все порты с этой скоростью;

- *тип сетевого носителя* – обычно это витая пара или оптоволокно, но существуют концентраторы и для других носителей, а также смешанные, например, для витой пары и коаксиального кабеля.

Основные и дополнительные функции концентраторов

Практически во всех современных технологиях локальных сетей определено устройство, которое имеет несколько равноправных названий – концентратор (concentrator), хаб (hub), повторитель (repeator). В зависимости от области применения этого устройства в значительной степени изменяется состав его функций и конструктивное исполнение. Неизменной остается только основная функция – это повторение кадра либо на всех портах (как определено в стандарте Ethernet), либо только на некоторых портах, в соответствии с алгоритмом, определенным соответствующим стандартом.

Концентратор обычно имеет несколько портов, к которым с помощью отдельных физических сегментов кабеля подключаются конечные узлы сети – компьютеры. Концентратор объединяет отдельные физические сегменты сети в единую разделяемую среду, доступ к которой осуществляется в соответствии с одним из рассмотренных протоколов локальных сетей – Ethernet, Token Ring и т. п. Так как логика доступа к разделяемой среде существенно зависит от технологии, то для каждого типа технологии выпускаются свои концентраторы – Ethernet, Token Ring, FDDI и 100VG-AnyLAN. Для конкретного протокола иногда используется свое, узкоспециализированное название этого устройства, более точно отражающее его функции или же использующееся в силу традиций, например, для концентраторов Token Ring характерно название MSAU.

Каждый концентратор выполняет некоторую основную функцию, определенную в соответствующем протоколе той технологии, которую он поддерживает. Хотя эта функция достаточно детально определена в стандарте технологии, при ее реализации концентраторы разных производителей могут отличаться такими деталями, как количество портов, поддержка нескольких типов кабелей и т. п.

Кроме основной функции концентратор может выполнять некоторое количество дополнительных функций, которые либо в стандарте вообще не определены, либо являются факультативными. Например, концентратор Token Ring может выполнять функцию отключения некорректно работающих портов и перехода на резервное кольцо, хотя в стандарте такие его возможности не описаны. Концентратор оказался удобным устройством для выполнения дополнительных функций, облегчающих контроль и эксплуатацию сети.

Рассмотрим особенности реализации основной функции концентратора на примере концентраторов Ethernet.

В технологии Ethernet устройства, объединяющие несколько физических сегментов коаксиального кабеля в единую разделяемую среду, использовались давно и получили название «повторителей» по своей основной функции – повторению на всех своих портах сигналов, полученных на входе одного из портов. В сетях на основе коаксиального кабеля обычными являлись двухпортовые повторители, соединяющие только два сегмента кабеля, поэтому термин концентратор к ним обычно не применялся.

С появлением спецификации 10Base-T для витой пары повторитель стал неотъемлемой частью сети Ethernet, так как без него связь можно было организовать только между двумя узлами сети. Многопортовые повторители Ethernet на витой паре стали называть концентраторами или хабами, так как в одном устройстве действительно концентрировались связи между большим количеством узлов сети. Концентратор Ethernet обычно имеет от 8 до 72 портов, причем основная часть портов предназначена для подключения кабелей на витой паре

Многопортовый повторитель-концентратор Ethernet может по-разному рассматриваться при использовании правила 4-х хабов. В большинстве моделей все порты связаны с единственным блоком повторения, и при прохождении сигнала между двумя портами повторителя блок повторения вносит задержку всего один раз. Поэтому такой концентратор нужно считать одним повторителем с ограничениями, накладываемыми правилом 4-х хабов. Но существуют и другие модели повторителей, в которых на несколько портов имеется свой блок повторения. В таком случае каждый блок

повторения нужно считать отдельным повторителем и учитывать его отдельно в правиле 4-х хабов.

Некоторые отличия могут демонстрировать модели концентраторов, работающие на одномодовый волоконно-оптический кабель. Дальность сегмента кабеля, поддерживаемого концентратором FDDI, на таком кабеле может значительно отличаться в зависимости от мощности лазерного излучателя – от 10 до 40 км.

Однако если существующие различия при выполнении основной функции концентраторов не столь велики, то их намного превосходит разброс в возможностях реализации концентраторами дополнительных функций.

Отключение портов

Очень полезной при эксплуатации сети является способность концентратора отключать некорректно работающие порты, изолируя тем самым остальную часть сети от возникших в узле проблем. Эту функцию называют автосегментацией (autopartitioning). Для концентратора FDDI эта функция для многих ошибочных ситуаций является основной, так как определена в протоколе. В то же время для концентратора Ethernet или Token Ring функция автосегментации для многих ситуаций является дополнительной, так как стандарт не описывает реакцию концентратора на эту ситуацию. Основной причиной отключения порта в стандартах Ethernet и Fast Ethernet является отсутствие ответа на последовательность импульсов link test, посылаемых во все порты каждые 16 мс. В этом случае неисправный порт переводится в состояние «отключен», но импульсы link test будут продолжать посылаться в порт с тем, чтобы при восстановлении устройства работа с ним была продолжена автоматически.

Рассмотрим ситуации, в которых концентраторы Ethernet и Fast Ethernet выполняют отключение порта.

- Ошибки на уровне кадра. Если интенсивность прохождения через порт кадров, имеющих ошибки, превышает заданный порог, то порт отключается, а затем, при отсутствии ошибок в течение заданного времени, включается снова. Такими ошибками могут быть: неверная контрольная сумм, неверная длина кадра (больше 1518 байт или меньше 64 байт), неоформленный заголовок кадра;

• множественные коллизии. Если концентратор фиксирует, что источником коллизии был один и тот же порт 60 раз подряд, то порт отключается. Через некоторое время порт снова будет включен.

• затянувшаяся передача (jabber). Как и сетевой адаптер, концентратор контролирует время прохождения одного кадра через порт. Если это время превышает время передачи кадра максимальной длины в 3 раза, то порт отключается.

Поддержка резервных связей

Так как использование резервных связей в концентраторах определено только в стандарте FDDI, то для остальных стандартов разработчики концентраторов поддерживают такую функцию с помощью своих частных решений. Например, концентраторы Ethernet/Fast Ethernet могут образовывать только иерархические связи без петель. Поэтому резервные связи всегда должны соединять отключенные порты, чтобы не нарушать логику работы сети. Обычно при конфигурировании концентратора администратор должен определить, какие порты являются основными, а какие по отношению к ним – резервными. Если по какой-либо причине порт отключается (срабатывает механизм автосегментации), концентратор делает активным его резервный порт. В некоторых моделях концентраторов разрешается использовать механизм назначения резервных портов только для оптоволоконных портов, считая, что нужно резервировать только наиболее важные связи, которые обычно выполняются на оптическом кабеле. В других же моделях резервным можно сделать любой порт.

Защита от несанкционированного доступа

Разделяемая среда предоставляет очень удобную возможность для несанкционированного прослушивания сети и получения доступа к передаваемым данным. Для этого достаточно подключить компьютер с программным анализатором протоколов к свободному разъему концентратора, записать на диск весь проходящий по сети трафик, а затем выделить из него нужную информацию.

Разработчики концентраторов предоставляют некоторый способ защиты данных в разделяемых средах.

Наиболее простой способ – назначение разрешенных MAC-адресов портам концентратора. В стандартном концентраторе Ethernet порты MAC-адресов не имеют. Защита заключается в том, что администратор вручную связывает с каждым портом концентратора

некоторый MAC-адрес. Этот MAC-адрес является адресом станции, которой разрешается подключаться к данному порту.

Заметим, что для реализации описанного метода защиты данных концентратор нужно предварительно сконфигурировать. Для этого концентратор должен иметь блок управления. Такие концентраторы обычно называют интеллектуальными. Блок управления представляет собой компактный вычислительный блок со встроенным программным обеспечением. Для взаимодействия администратора с блоком управления концентратор имеет консольный порт (чаще всего RS-232), к которому подключается терминал или персональный компьютер с программой эмуляции терминала. При присоединении терминала блок управления организует на его экране диалог, с помощью которого администратор вводит значения MAC-адресов. Блок управления может поддерживать и другие операции конфигурирования, например ручное отключение или включение портов и т. д. Для этого при подключении терминала блок управления выдает на экран некоторое меню, с помощью которого администратор выбирает нужное действие.

Другим способом защиты данных от несанкционированного доступа является их шифрация. Однако процесс истинной шифрации требует большой вычислительной мощности, и для повторителя, не буферизующего кадр, выполнить шифрацию «на лету» весьма сложно. Вместо этого в концентраторах применяется метод случайного искажения поля данных в пакетах, передаваемых портам с адресом, отличным от адреса назначения пакета. Этот метод сохраняет логику случайного доступа к среде, так как все станции видят занятость среды кадром информации, но только станция, которой послан этот кадр, может понять содержание поля данных кадра. Для реализации этого метода концентратор также нужно снабдить информацией о том, какие MAC-адреса имеют станции, подключенные к его портам. Обычно поле данных в кадрах, направляемых станциям, отличным от адресата, заполняется нулями.

Многосегментные концентраторы

При рассмотрении некоторых моделей концентраторов возникает вопрос – зачем в этой модели имеется такое большое количество портов, например 192 или 240? Имеет ли смысл разделять среду в 10 или 16 Мбит/с между таким большим количеством станций? Возможно, десять – пятнадцать лет назад ответ в некоторых

случаях мог бы быть и положительным, например, для тех сетей, в которых компьютеры пользовались сетью только для отправки небольших почтовых сообщений или для переписывания небольшого текстового файла. Сегодня таких сетей осталось крайне мало, и даже 5 компьютеров могут полностью загрузить сегмент Ethernet или Token Ring, а в некоторых случаях – и сегмент Fast Ethernet. Для чего же тогда нужен концентратор с большим количеством портов, если ими практически нельзя воспользоваться из-за ограничений по пропускной способности, приходящейся на одну станцию? Ответ состоит в том, что в таких концентраторах имеется несколько несвязанных внутренних шин, которые предназначены для создания нескольких разделяемых сред.

Многосегментные концентраторы нужны для создания разделяемых сегментов, состав которых может легко изменяться. Большинство многосегментных концентраторов, например System 5000 компании Nortel Networks или PortSwitch Hub компании 3Com, позволяют выполнять операцию соединения порта с одной из внутренних шин чисто программным способом, например с помощью локального конфигурирования через консольный порт, В результате администратор сети может присоединять компьютеры пользователей к любым портам концентратора, а затем с помощью программы конфигурирования концентратора управлять составом каждого сегмента. Если завтра сегмент 1 станет перегруженным, то его компьютеры можно распределить между оставшимися сегментами концентратора.

Возможность многосегментного концентратора программно изменять связи портов с внутренними шинами называется конфигурационной коммутацией (configuration switching).

Многосегментные концентраторы – это программируемая основа больших сетей. Для соединения сегментов между собой нужны устройства другого типа – мосты/коммутаторы или маршрутизаторы. Такое межсетевое устройство должно подключаться к нескольким портам многосегментного концентратора, подсоединенным к разным внутренним шинам, и выполнять передачу кадров или пакетов между сегментами точно так же, как если бы они были образованы отдельными устройствами-концентраторами.

Для крупных сетей многосегментный концентратор играет роль интеллектуального кроссового шкафа, который выполняет новое

соединение не за счет механического перемещения вилки кабеля в новый порт, а за счет программного изменения внутренней конфигурации устройства.

Управление концентратором по протоколу SNMP

Как видно из описания дополнительных функций, многие из них требуют конфигурирования концентратора. Это конфигурирование может производиться локально, через интерфейс RS-232C, который имеется у любого концентратора, имеющего блок управления. Кроме конфигурирования в большой сети очень полезна функция наблюдения за состоянием концентратора: работоспособен ли он, в каком состоянии находятся его порты.

При большом количестве концентраторов и других коммуникационных устройств в сети постоянное наблюдение за состоянием многочисленных портов и изменением их параметров становится очень обременительным занятием, если оно должно выполняться с помощью локального подключения терминала. Поэтому большинство концентраторов, поддерживающих интеллектуальные дополнительные функции, могут управляться централизованно по сети с помощью популярного протокола управления SNMP (Simple Network Management Protocol) из стека TCP/IP.

В блок управления концентратором встраивается так называемый SNMP-агент. Этот агент собирает информацию о состоянии контролируемого устройства и хранит ее в так называемой базе данных управляющей информации – Management In formation Base, MIB. Эта база данных имеет стандартную структуру, что позволяет одному из компьютеров сети, выполняющему роль центральной станции управления, запрашивать у агента значения стандартных переменных базы MIB. В базе MIB хранятся не только данные о состоянии устройства, но и управляющая информация, воздействующая на это устройство. Например, в MIB есть переменная, управляющая состоянием порта, имеющая значения «включить» и «выключить». Если станция управления меняет значение управляющей переменной, то агент должен выполнить это указание и воздействовать на устройство соответствующим образом, например выключить порт или изменить связь порта с внутренними шинами концентратора.

Взаимодействие между станцией управления (по-другому – менеджером системы управления) и встроенными в коммуникационные устройства агентами происходит по протоколу SNMP. Концентратор, который управляется по протоколу SNMP, должен поддерживать основные протоколы стека TCP/IP и иметь IP- и MAC – адреса. Точнее, эти адреса относятся к агенту концентратора. Поэтому администратор, который хочет воспользоваться преимуществами централизованного управления концентраторами по сети, должен знать стек протоколов TCP/IP и сконфигурировать IP-адреса их агентов.

Коммутаторы

Характеристики коммутаторов

Производительность коммутатора – то свойство, которое сетевые интеграторы и администраторы ждут от этого устройства в первую очередь.

Основными показателями коммутатора, характеризующими его производительность, являются:

1. скорость фильтрации кадров;
2. скорость продвижения кадров;
3. пропускная способность;
4. задержка передачи кадра.

Кроме того, существует несколько характеристик коммутатора, которые в наибольшей степени влияют на указанные характеристики производительности. К ним относятся:

1. тип коммутации – «на лету» или с полной буферизацией;
2. размер буфера (буферов) кадров;
3. производительность внутренней шины;
4. производительность процессора или процессоров;
5. размер внутренней адресной таблицы.

Скорость фильтрации и скорость продвижения

Скорость фильтрации и продвижения кадров – это две основные характеристики производительности коммутатора. Эти характеристики являются интегральными показателями, они не зависят от того, каким образом технически реализован коммутатор.

Скорость фильтрации (filtering) определяет скорость, с которой коммутатор выполняет следующие этапы обработки кадров:

1. прием кадра в свой буфер;

2. просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра;

3. уничтожение кадра, так как его порт назначения и порт источника принадлежат одному логическому сегменту.

Скорость фильтрации практически у всех коммутаторов является неблокирующей – коммутатор успевает отбрасывать кадры в темпе их поступления.

Скорость продвижения (forwarding) определяет скорость, с которой коммутатор выполняет следующие этапы обработки кадров.

1. прием кадра в свой буфер;

2. просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра;

3. передача кадра в сеть через найденный по адресной таблице порт назначения.

Как скорость фильтрации, так и скорость продвижения измеряются обычно в кадрах в секунду. Если в характеристиках коммутатора не уточняется, для какого протокола и для какого размера кадра приведены значения скоростей фильтрации и продвижения, то по умолчанию считается, что эти показатели даются для протокола Ethernet и кадров минимального размера, то есть кадров длиной 64 байт (без преамбулы) с полем данных в 46 байт. Если скорости указаны для какого-либо определенного протокола, например Token Ring или FDDI, то они также даны для кадров минимальной длины этого протокола (например, кадров длины 29 байт для протокола FDDI). Применение в качестве основного показателя скорости работы коммутатора кадров минимальной длины объясняется тем, что такие кадры всегда создают для коммутатора наиболее тяжелый режим работы по сравнению с кадрами другого формата при равной пропускной способности переносимых пользовательских данных. Поэтому при проведении тестирования коммутатора режим передачи кадров минимальной длины используется как наиболее сложный тест, который должен проверить способность коммутатора работать при наихудшем сочетании параметров трафика. Кроме того, для пакетов минимальной длины скорость фильтрации и продвижения максимальна, что имеет немаловажное значение при рекламе коммутатора.

Пропускная способность коммутатора

Пропускная способность коммутатора измеряется количеством пользовательских данных (в мегабитах в секунду), переданных в единицу времени через его порты. Так как коммутатор работает на канальном уровне, для него пользовательскими данными являются те данные, которые переносятся в поле данных кадров протоколов канального уровня – Ethernet, Token Ring, FDDI и т. п. Максимальное значение пропускной способности коммутатора всегда достигается на кадрах максимальной длины, так как при этом доля накладных расходов на служебную информацию кадра гораздо ниже, чем для кадров минимальной длины, а время выполнения коммутатором операций по обработке кадра, приходящееся на один байт пользовательской информации, существенно меньше. Поэтому коммутатор может быть блокирующим для кадров минимальной длины, но при этом иметь очень хорошие показатели пропускной способности.

Задержка передачи кадра

Задержка передачи кадра измеряется как время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на его выходном порту. Задержка складывается из времени, затрачиваемого на буферизацию байт кадра, а также времени, затрачиваемого на обработку кадра коммутатором, – просмотра адресной таблицы, принятия решения о фильтрации или продвижении и получения доступа к среде выходного порта.

Величина вносимой коммутатором задержки зависит от режима его работы. Если коммутация осуществляется «на лету», то задержки обычно невелики и составляют от 5 до 40 мкс, а при полной буферизации кадров – от 50 до 200 мкс (для кадров минимальной длины).

Коммутатор – это многопортовое устройство, поэтому для него принято все приведенные выше характеристики (кроме задержки передачи кадра) давать в двух вариантах. Первый вариант – суммарная производительность коммутатора при одновременной передаче трафика по всем его портам, второй вариант – производительность, приведенная в расчете на один порт. Обычно производители коммутаторов указывают общую максимальную пропускную способность устройства.

Тип коммутации

На производительности коммутатора сказывается способ передачи пакетов – «на лету» или с буферизацией. Коммутаторы, передающие пакеты «на лету», вносят меньшие задержки передачи кадров на каждом промежуточном коммутаторе, поэтому общее уменьшение задержки доставки данных может быть значительным, что важно для мультимедийного трафика. Кроме того, выбранный способ коммутации оказывает влияние на возможности реализации некоторых полезных дополнительных функций, например трансляцию протоколов канального уровня.

Средняя величина задержки коммутаторов, работающих «на лету», при высокой нагрузке объясняется тем, что в этом случае выходной порт часто бывает занят приемом другого пакета, поэтому вновь поступивший пакет для данного порта все равно приходится буферизовать.

Коммутатор, работающий «на лету», может выполнять проверку некорректности передаваемых кадров, но не может изъять плохой кадр из сети, так как часть его байт (и, как правило, большая часть) уже переданы в сеть.

Так как каждый способ имеет свои достоинства и недостатки, в тех моделях коммутаторов, которым н нужно транслировать протоколы, иногда применяется механизм адаптивной смены режима работы коммутатора. Основной режим такого коммутатора – коммутация «на лету», но коммутатор постоянно контролирует трафик и при превышении интенсивности появления плохих кадров некоторого порога переходит на режим полной буферизации. Затем коммутатор может вернуться к коммутации «на лету».

Размер адресной таблицы

Максимальная емкость адресной таблицы определяет предельное количество MAC-адресов, с которыми может одновременно оперировать коммутатор. Так как коммутаторы чаще всего используют для выполнения операций каждого порта выделенный процессорный блок со своей памятью для хранения экземпляра адресной таблицы, то размер адресной таблицы для коммутаторов обычно приводится в расчете на один порт. Экземпляры адресной таблицы разных процессорных модулей не обязательно содержат одну и ту же адресную информацию – скорее всего, повторяющихся адресов будет не так много, если только

распределение трафика каждого порта между остальными портами не полностью равновероятно. Каждый порт хранит только те наборы адресов, с которыми он работал в последнее время.

Значение максимального числа MAC-адресов, которое может запомнить процессор порта, зависит от области применения коммутатора. Коммутаторы рабочих групп обычно поддерживают всего несколько адресов на порт, так как они предназначены для образования микросегментов. Коммутаторы отделов должны поддерживать несколько сотен адресов, а коммутаторы магистралей сетей – до нескольких тысяч, обычно 4000-8000 адресов.

Недостаточная емкость адресной таблицы может служить причиной замедления работы коммутатора и засорения сети избыточным трафиком. Если адресная таблица процессора порта полностью заполнена, а он встречает новый адрес источника в поступившем пакете, процессор должен вытеснить из таблицы какой-либо старый адрес и поместить на его место новый. Эта операция сама по себе отнимет у процессора часть времени, но главные потери производительности будут наблюдаться при поступлении кадра с адресом назначения, который пришлось удалить из адресной таблицы. Так как адрес назначения кадра неизвестен, то коммутатор должен передать этот кадр на все остальные порты. Эта операция будет создавать лишнюю работу для многих процессоров портов, кроме того, копии этого кадра будут попадать и на те сегменты сети, где они совсем не обязательны.

Некоторые производители коммутаторов решают эту проблему за счет изменения алгоритма обработки кадров с неизвестным адресом назначения. Один из портов коммутатора конфигурируется как магистральный порт, на который по умолчанию передаются все кадры с неизвестным адресом. В маршрутизаторах такой прием применяется давно, позволяя сократить размеры адресных таблиц в сетях, организованных по иерархическому принципу.

Передача кадра на магистральный порт производится в расчете на то, что этот порт подключен к вышестоящему коммутатору при иерархическом соединении коммутаторов в крупной сети, который имеет достаточную емкость адресной таблицы и знает, куда нужно передать любой кадр.

Объем буфера кадров

Внутренняя буферная память коммутатора нужна для временного хранения кадров данных в тех случаях, когда их невозможно немедленно передать на выходной порт. Буфер предназначен для сглаживания кратковременных пульсаций трафика. Ведь даже если трафик хорошо сбалансирован и производительность процессоров портов, а также других обрабатывающих элементов коммутатора достаточна для передачи средних значений графика, это не гарантирует, что их производительности хватит при пиковых значениях нагрузок. Например, трафик может в течение нескольких десятков миллисекунд поступать одновременно на все входы коммутатора, не давая ему возможности передавать принимаемые кадры на выходные порты.

Для предотвращения потерь кадров при кратковременном многократном превышении среднего значения интенсивности трафика (а для локальных сетей часто встречаются значения коэффициента пульсации трафика в диапазоне 50-100) единственным средством служит буфер большого объема. Как и в случае адресных таблиц, каждый процессорный модуль порта обычно имеет свою буферную память для хранения кадров. Чем больше объем этой памяти, тем менее вероятны потери кадров при перегрузках, хотя при несбалансированности средних значений трафика буфер все равно рано или поздно переполнится.

Обычно коммутаторы, предназначенные для работы в ответственных частях сети, имеют буферную память в несколько десятков или сотен килобайт на порт. Хорошо, когда эту буферную память можно перераспределять между несколькими портами, так как одновременные перегрузки по нескольким портам маловероятны. Дополнительным средством защиты может служить общий для всех портов буфер в модуле управления коммутатором. Такой буфер обычно имеет объем в несколько мегабайт.

Дополнительные функции коммутаторов

Так как коммутатор представляет собой сложное вычислительное устройство, имеющее несколько процессорных модулей, то естественно нагрузить его помимо выполнения основной функции передачи кадров с порта на порт по алгоритму моста и некоторыми дополнительными функциями, полезными при построении надежных и гибких сетей. Ниже описываются наиболее

распространенные дополнительные функции коммутаторов, которые поддерживаются большинством производителей коммуникационного оборудования.

Поддержка алгоритма Spanning Tree

Алгоритм покрывающего дерева – Spanning Tree Algorithm (STA) позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединения портов между собой. Как уже отмечалось, для нормальной работы коммутатора требуется отсутствие замкнутых маршрутов в сети. Эти маршруты могут создаваться администратором специально для образования резервных связей или же возникать случайным образом, что вполне возможно, если сеть имеет многочисленные связи, а кабельная система плохо структурирована или документирована.

Поддерживающие алгоритм STA коммутаторы автоматически создают активную древовидную конфигурацию связей (то есть связную конфигурацию без петель) на множестве всех связей сети. Такая конфигурация называется покрывающим деревом – Spanning Tree (иногда ее называют основным деревом), и ее название дало имя всему алгоритму. Алгоритм Spanning Tree описан в стандарте IEEE 802.1D, том же стандарте, который определяет принципы работы прозрачных мостов.

Коммутаторы находят покрывающее дерево адаптивно, с помощью обмена служебными пакетами. Реализация в коммутаторе алгоритма STA очень важна для работы в больших сетях – если коммутатор не поддерживает этот алгоритм, то администратор должен самостоятельно определить, какие порты нужно перевести в заблокированное состояние, чтобы исключить петли. К тому же при отказе какого-либо кабеля, порта или коммутатора администратор должен, во-первых, обнаружить факт отказа, а во-вторых, ликвидировать последствия отказа, переведя резервную связь в рабочий режим путем активизации некоторых портов. При поддержке коммутаторами сети протокола Spanning Tree отказы обнаруживаются автоматически, за счет постоянного тестирования связности сети служебными пакетами. После обнаружения потери связности протокол строит новое покрывающее дерево, если это возможно, и сеть автоматически восстанавливает работоспособность.

Алгоритм Spanning Tree определяет активную конфигурацию сети за три этапа.

Сначала в сети определяется корневой коммутатор (root switch), от которого строится дерево. Корневой коммутатор может быть выбран автоматически или назначен администратором. При автоматическом выборе корневым становится коммутатор с меньшим значением MAC-адреса его блока управления.

Затем, на втором этапе, для каждого коммутатора определяется корневой порт (root port) – это порт, который имеет по сети кратчайшее расстояние до корневого коммутатора (точнее, до любого из портов корневого коммутатора).

И наконец, на третьем этапе для каждого сегмента сети выбирается так называемый назначенный порт (designated port) – это порт, который имеет кратчайшее расстояние от данного сегмента до корневого коммутатора. После определения корневых и назначенных портов каждый коммутатор блокирует остальные порты, которые не попали в эти два класса портов. Можно математически доказать, что при таком выборе активных портов в сети исключаются петли и оставшиеся связи образуют покрывающее дерево (если оно может быть построено при существующих связях в сети).

Понятие расстояния играет важную роль в построении покрывающего дерева. Именно по этому критерию выбирается единственный порт, соединяющий каждый коммутатор с корневым коммутатором, и единственный порт, соединяющий каждый сегмент сети с корневым коммутатором.

Трансляция протоколов канального уровня

Коммутаторы могут выполнять трансляцию одного протокола канального уровня в другой, например Ethernet в FDDI, Fast Ethernet в Token Ring и т. п. При этом они работают по тем же алгоритмам, что и транслирующие мосты, то есть в соответствии со спецификациями IEEE 802.1H и RFC 1042, определяющими правила преобразования полей кадров разных протоколов.

Трансляцию протоколов локальных сетей облегчает тот факт, что наиболее сложную работу, которую при объединении гетерогенных сетей часто выполняют маршрутизаторы и шлюзы, а именно работу по трансляции адресной информации, в данном случае выполнять не нужно. Все конечные узлы локальных сетей имеют уникальные адреса одного и того же формата независимо от

поддерживаемого протокола. Поэтому адрес сетевого адаптера Ethernet понятен сетевому адаптеру FDDI, и они могут использовать эти адреса в полях своих кадров не задумываясь о том, что узел, с которым они взаимодействуют, принадлежит сети, работающей по другой технологии.

Поэтому при согласовании протоколов локальных сетей коммутаторы не строят таблиц соответствия адресов узлов, а переносят адреса назначения и источника из кадра одного протокола в кадр другого.

Фильтрация трафика

Многие коммутаторы позволяют администраторам задавать дополнительные условия фильтрации кадров наряду со стандартными условиями их фильтрации в соответствии с информацией адресной таблицы. Пользовательские фильтры предназначены для создания дополнительных барьеров на пути кадров, которые ограничивают доступ определенных групп пользователей к определенным службам сети.

Наиболее простыми являются пользовательские фильтры на основе MAC-адресов станций. Так как MAC-адреса – это та информация, с которой работает коммутатор, то он позволяет задавать такие фильтры в удобной для администратора форме, возможно, проставляя некоторые условия в дополнительном поле адресной таблицы, например, отбрасывать кадры с определенным адресом. При этом пользователю, работающему на компьютере с данным MAC-адресом, полностью запрещается доступ к ресурсам другого сегмента сети.

Часто администратору требуется задать более тонкие условия фильтрации, например запретить некоторому пользователю печатать свои документы на определенном сервере печати NetWare чужого сегмента, а остальные ресурсы этого сегмента сделать доступными. Для реализации такого фильтра нужно запретить передачу кадров с определенным MAC-адресом, в которых вложены пакеты IPX, в поле «номер сокета» которых будет указано значение, соответствующее службе печати NetWare. Коммутаторы не анализируют протоколы верхних уровней, такие как IPX, поэтому администратору приходится для задания условий такой фильтрации вручную определять поле, по значению которого нужно осуществлять фильтрацию, в виде пары «смещение – размер» относительно начала поля данных кадра

канального уровня, а затем еще указать в шестнадцатеричном формате значение этого поля для службы печати.

Обычно условия фильтрации записываются в виде булевых выражений, формируемых с помощью логических операторов AND и OR.

Наложение дополнительных условий фильтрации может снизить производительность коммутатора, так как вычисление булевых выражений требует проведения дополнительных вычислений процессорами портов.

Приоритетная обработка кадров

Построение сетей на основе коммутаторов позволяет использовать приоритезацию трафика, причем делать это независимо от технологии сети. Эта новая возможность (по сравнению с сетями, построенными целиком на концентраторах) является следствием того, что коммутаторы буферизуют кадры перед их отправкой на другой порт. Коммутатор обычно ведет для каждого входного и выходного порта не одну, а несколько очередей, причем каждая очередь имеет свой приоритет обработки. При этом коммутатор может быть сконфигурирован, например, так, чтобы передавать один низкоприоритетный пакет на каждые 10 высокоприоритетных пакетов.

Поддержка приоритетной обработки может особенно пригодиться для приложений, предъявляющих различные требования к допустимым задержкам кадров и к пропускной способности сети для потока кадров.

Приоритезация трафика коммутаторами сегодня является одним из основных механизмов обеспечения качества транспортного обслуживания в локальных сетях. Это, естественно, не гарантированное качество обслуживания, а только механизм best effort – «с максимальными усилиями». К каким уровням задержек приводит приписывание того или иного уровня приоритета кадру, какую пропускную способность обеспечивает приоритет потоку кадров – схема приоритезации не говорит. Выяснить последствия ее применения можно только путем проведения натурных экспериментов или же с помощью имитационного моделирования. Ясно только одно – более приоритетные кадры будут обрабатываться раньше менее приоритетных, поэтому все показатели качества обслуживания у них будут выше, чем у менее приоритетных.

Остается вопрос – насколько? Гарантии качества обслуживания дают другие схемы, которые основаны на предварительном резервировании качества обслуживания. Например, такие схемы используются в технологиях глобальных сетей frame relay и ATM или в протоколе RSVP для сетей TCP/IP. Однако для коммутаторов такого рода протоколов нет, так что гарантий качества обслуживания они пока дать не могут.

Основным вопросом при приоритетной обработке кадров коммутаторами является вопрос назначения кадру приоритета. Так как не все протоколы канального уровня поддерживают поле приоритета кадра, например у кадров Ethernet оно отсутствует, то коммутатор должен использовать какой-либо дополнительный механизм для связывания кадра с его приоритетом. Наиболее распространенный способ – приписывание приоритета портам коммутатора. При этом способе коммутатор помещает кадр в очередь кадров соответствующего приоритета в зависимости от того, через какой порт поступил кадр в коммутатор. Способ несложный, но недостаточно гибкий – если к порту коммутатора подключен не отдельный узел, а сегмент, то все узлы сегмента получают одинаковый приоритет.

Многие компании, выпускающие коммутаторы, реализовали в них ту или иную схему приоритетной обработки кадров. Примером фирменного подхода к назначению приоритетов на основе портов является технология PACE компании 3Com.

Более гибким является назначение приоритетов кадрам в соответствии с достаточно новым стандартом IEEE 802.1p. Этот стандарт разрабатывался совместно со стандартом 802.10, который рассматривается в следующем разделе, посвященном виртуальным локальным сетям. В обоих стандартах предусмотрен общий дополнительный заголовок для кадров Ethernet, состоящий из двух байт. В этом дополнительном заголовке, который вставляется перед полем данных кадра, 3 бита используются для указания приоритета кадра. Существует протокол, по которому конечный узел может запросить у коммутатора один из восьми уровней приоритета кадра. Если сетевой адаптер не поддерживает стандарт 802.1p, то коммутатор может назначать приоритеты кадрам на основе порта поступления кадра. Такие помеченные кадры будут обслуживаться в соответствии с их приоритетом всеми коммутаторами сети, а не

только тем коммутатором, который непосредственно принял кадр от конечного узла. При передаче кадра сетевому адаптеру, не поддерживающему стандарт 802.1p, дополнительный заголовок должен быть удален.

Виртуальные локальные сети

Кроме своего основного назначения – повышения пропускной способности связей в сети – коммутатор позволяет локализовывать потоки информации в сети, а также контролировать эти потоки и управлять ими, опираясь на механизм пользовательских фильтров. Однако пользовательский фильтр может запретить передачи кадров только по конкретным адресам, а широковещательный трафик он передает всем сегментам сети. Так требует алгоритм работы моста, который реализован в коммутаторе, поэтому сети, созданные на основе мостов и коммутаторов, иногда называют плоскими – из-за отсутствия барьеров на пути широковещательного трафика.

Технология виртуальных локальных сетей (Virtual LAN, VLAN), которая появилась несколько лет тому назад в коммутаторах, позволяет преодолеть указанное ограничение. Виртуальной сетью называется группа узлов сети, трафик которой, в том числе и широковещательный, на канальном уровне полностью изолирован от других узлов сети . Это означает, что передача кадров между разными виртуальными сетями на основании адреса канального уровня невозможна, независимо от типа адреса – уникального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра. Виртуальные сети могут пересекаться, если один или несколько компьютеров входят в состав более чем одной виртуальной сети.

Маршрутизаторы

Основная задача маршрутизатора – выбор наилучшего маршрута в сети – часто является достаточно сложной с математической точки зрения. Особенно интенсивных вычислений требуют протоколы, основанные на алгоритме состояния связей, вычисляющие оптимальный путь на графе, – OSPF, NLSP, IS-IS. Кроме этой основной функции в круг ответственности маршрутизатора входят и другие задачи, такие как буферизация, фильтрация и фрагментация перемещаемых пакетов. При этом очень

важна производительность, с которой маршрутизатор выполняет эти задачи.

Поэтому типичный маршрутизатор является мощным вычислительным устройством с одним или даже несколькими процессорами, часто специализированными или построенными на RISC-архитектуре, со сложным программным обеспечением. То есть сегодняшний маршрутизатор – это специализированный компьютер, имеющий скоростную внутреннюю шину или шины (с пропускной способностью 600-2000 Мбит/с), часто использующий симметричное или асимметричное мультипроцессирование и работающий под управлением специализированной операционной системы, относящейся к классу систем реального времени. Многие разработчики маршрутизаторов построили в свое время такие операционные системы на базе операционной системы Unix, естественно, значительно ее переработав.

Маршрутизаторы могут поддерживать как один протокол сетевого уровня (например, IP, IPX или DECnet), так и множество таких протоколов. В последнем случае они называются многопротокольными маршрутизаторами. Чем больше протоколов сетевого уровня поддерживает маршрутизатор, тем лучше он подходит для корпоративной сети.

Большая вычислительная мощность позволяет маршрутизаторам наряду с основной работой по выбору оптимального маршрута выполнять и ряд вспомогательных высокоуровневых функций.

Основные технические характеристики маршрутизатора

Основные технические характеристики маршрутизатора связаны с тем, как он решает свою главную задачу – маршрутизацию пакетов в составной сети. Именно эти характеристики прежде всего определяют возможности и сферу применения того или иного маршрутизатора.

Перечень поддерживаемых сетевых протоколов

Магистральный маршрутизатор должен поддерживать большое количество сетевых протоколов и протоколов маршрутизации, чтобы обеспечивать трафик всех существующих на предприятии вычислительных систем (в том числе и устаревших, но все еще успешно эксплуатирующихся, так называемых унаследованных – legacy), а также систем, которые могут появиться на предприятии в ближайшем будущем. Если центральная сеть образует отдельную

автономную систему Internet, то потребуется поддержка и специфических протоколов маршрутизации этой сети, таких как EGP и BGP. Программное обеспечение магистральных маршрутизаторов обычно строится по модульному принципу, поэтому при возникновении потребности можно докупать и добавлять программные модули, реализующие недостающие протоколы.

Перечень поддерживаемых сетевых протоколов обычно включает протоколы IP, CONS и CLNS OSI, IPX, AppleTalk, DECnet, Banyan VINES, Xerox XNS.

Перечень протоколов маршрутизации составляют протоколы IP RIP, IPX RIP, NLSP, OSPF, IS-IS OSI, EGP, BGP, VINES RTP, AppleTalk RTMP.

Перечень поддерживаемых интерфейсов локальных и глобальных сетей

Для локальных сетей – это интерфейсы, реализующие физические и канальные протоколы сетей Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN иATM.

Для глобальных связей – это интерфейсы физического уровня для связи с аппаратурой передачи данных, а также протоколы канального и сетевого уровней, необходимые для подключения к глобальным сетям с коммутацией каналов и пакетов.

Поддерживаются интерфейсы последовательных линий (serial lines) RS-232, RS-449/422, V.35 (для передачи данных со скоростями до 2-6 Мбит/с), высокоскоростной интерфейс HSSI, обеспечивающий скорость до 52 Мбит/с, а также интерфейсы с цифровыми каналами T1/E1, T3/E3 и интерфейсами BRI и PRI цифровой сети ISDN. Некоторые маршрутизаторы имеют аппаратуру связи с цифровыми глобальными каналами, что исключает необходимость использования внешних устройств сопряжения с этими каналами.

В набор поддерживаемых глобальных технологий обычно входят технологии X.25, frame relay, ISDN и коммутируемых аналоговых телефонных сетей, сетей ATM, а также поддержка протокола канального уровня PPP.

Общая производительность маршрутизатора

Высокая производительность маршрутизации важна для работы с высокоскоростными локальными сетями, а также для поддержки новых высокоскоростных глобальных технологий, таких как frame relay, T3/E3, SDH и ATM. Общая производительность

маршрутизатора зависит от многих факторов, наиболее важными из которых являются: тип используемых процессоров, эффективность программной реализации протоколов, архитектурная организация вычислительных и интерфейсных модулей. Общая производительность маршрутизаторов колеблется от нескольких десятков тысяч пакетов в секунду до нескольких миллионов пакетов в секунду. Наиболее производительные маршрутизаторы имеют мультипроцессорную архитектуру, сочетающую симметричные и асимметричные свойства – несколько мощных центральных процессоров по симметричной схеме выполняют функции вычисления таблицы маршрутизации, а менее мощные процессоры в интерфейсных модулях занимаются передачей пакетов на подключенные к ним сети и пересылкой пакетов на основании части таблицы маршрутизации, кэшированной в локальной памяти интерфейсного модуля.

Магистральные маршрутизаторы обычно поддерживают максимальный набор протоколов и интерфейсов и обладают высокой общей производительностью в один-два миллиона пакетов в секунду. Маршрутизаторы удаленных офисов поддерживают один-два протокола локальных сетей и низкоскоростные глобальные протоколы, общая производительность таких маршрутизаторов обычно составляет от 5 до 20-30 тысяч пакетов в секунду.

Маршрутизаторы региональных отделений занимают промежуточное положение, поэтому их иногда не выделяют в отдельный класс устройств.

Наиболее высокой производительностью обладают коммутаторы 3-го уровня, особенности которых рассмотрены ниже.

Дополнительные функциональные возможности маршрутизаторов

Наряду с функцией маршрутизации многие маршрутизаторы обладают следующими важными дополнительными функциональными возможностями, которые значительно расширяют сферу применения этих устройств.

Поддержка одновременно нескольких протоколов маршрутизации

В протоколах маршрутизации обычно предполагается, что маршрутизатор строит свою таблицу на основе работы только этого одного протокола. Деление Internet на автономные системы также

направлено на исключение использования в одной автономной системе нескольких протоколов маршрутизации. Тем не менее иногда в большой корпоративной сети приходится поддерживать одновременно несколько таких протоколов, чаще всего это складывается исторически. При этом таблица маршрутизации может получаться противоречивой – разные протоколы маршрутизации могут выбрать разные следующие маршрутизаторы для какой-либо сети назначения. Большинство маршрутизаторов решает эту проблему за счет придания приоритетов решениям разных протоколов маршрутизации. Высший приоритет отдается статическим маршрутам (администратор всегда прав), следующий приоритет имеют маршруты, выбранные протоколами состояния связей, такими как OSPF или NLSP, а низшим приоритетов обладают маршруты дистанционно-векторных протоколов, как самых несовершенных.

Приоритеты сетевых протоколов

Можно установить приоритет одного протокола сетевого уровня над другими. На выбор маршрутов эти приоритеты не оказывают никакого влияния, они влияют только на порядок, в котором многопротокольный маршрутизатор обслуживает пакеты разных сетевых протоколов. Это свойство бывает полезно в случае недостаточной полосы пропускания кабельной системы и существования трафика, чувствительного к временным задержкам, например, трафика SNA или голосового трафика, передаваемого одним из сетевых протоколов.

Поддержка политики маршрутных объявлений

В большинстве протоколов обмена маршрутной информации (RIP, OSPF, NLSP) предполагается, что маршрутизатор объявляет в своих сообщениях обо всех сетях, которые ему известны. Аналогично предполагается, что маршрутизатор при построении своей таблицы учитывает все адреса сетей, которые поступают ему от других маршрутизаторов сети. Однако существуют ситуации, когда администратор хотел бы скрыть существование некоторых сетей в определенной части своей сети от других администраторов, например, по соображениям безопасности. Или же администратор хотел бы запретить некоторые маршруты, которые могли бы существовать в сети. При статическом построении таблиц маршрутизации решение таких проблем не составляет труда.

Динамические же протоколы маршрутизации не позволяют стандартным способом реализовывать подобные ограничения. Существует только один широко используемый протокол динамической маршрутизации, в котором описана возможность существования *правил (policy)*, ограничивающих распространение некоторых адресов в объявлениях, – это протокол BGP. Необходимость поддержки таких правил в протоколе BGP понятна, так как это протокол обмена маршрутной информацией между автономными системами, где велика потребность в административном регулировании маршрутов (например, некоторый поставщик услуг Internet может не захотеть, чтобы через него транзитом проходил трафик другого поставщика услуг). Разработчики маршрутизаторов исправляют этот недостаток стандартов протоколов, вводя в маршрутизаторы поддержку правил передачи и использования маршрутной информации, подобных тем, которые рекомендует BGP.

Защита от широковещательных штормов (broadcast storm)

Одна из характерных неисправностей сетевого программного обеспечения – самопроизвольная генерация с высокой интенсивностью широковещательных пакетов. Широковещательным штормом считается ситуация, в которой процент широковещательных пакетов превышает 20 % от общего количества пакетов в сети. Обычный коммутатор или мост слепо передает такие пакеты на все свои порты, как того требует его логика работы, засоряя, таким образом, сеть. Борьба с широковещательным штормом в сети, соединенной коммутаторами, требует от администратора отключения портов, генерирующих широковещательные пакеты. Маршрутизатор не распространяет такие поврежденные пакеты, поскольку в круг его задач не входит копирование широковещательных пакетов во все объединяемые им сети. Поэтому маршрутизатор является прекрасным средством борьбы с широковещательным штормом, правда, если сеть разделена на достаточное количество подсетей.

Поддержка немаршрутизируемых протоколов

таких как NetBIOS, NetBEUI или DEC LAT, которые не оперируют с таким понятием, как сеть. Маршрутизаторы могут обрабатывать пакеты таких протоколов двумя способами.

1. В первом случае они могут работать с пакетами этих протоколов как мосты, то есть передавать их на основании изучения

MAC – адресов. Маршрутизатор необходимо сконфигурировать особым способом, чтобы по отношению к некоторым немаршрутизируемым протоколам на некоторых портах он выполнял функции моста, а по отношению к маршрутизируемым протоколам – функции маршрутизатора. Такой мост/маршрутизатор иногда называют brouter (bridge плюс router).

2. Другим способом передачи пакетов немаршрутизируемых протоколов является инкапсуляция этих пакетов в пакеты какого-либо сетевого протокола. Некоторые производители маршрутизаторов разработали собственные протоколы, специально предназначенные для инкапсуляции немаршрутизируемых пакетов. Кроме того, существуют стандарты для инкапсуляции некоторых протоколов в другие, в основном в IP. Примером такого стандарта является протокол DLSw, определяющий методы инкапсуляции пакетов SDLC и NetBIOS в IP-пакеты, а также протоколы PPTP и L2TP, инкапсулирующие кадры протокола PPP в IP-пакеты. Более подробно технология инкапсуляции рассматривается в главе, посвященной межсетевому взаимодействию.

Разделение функций построения и использования таблицы маршрутизации

Основная вычислительная работа проводится маршрутизатором при составлении таблицы маршрутизации с маршрутами ко всем известным ему сетям. Эта работа состоит в обмене пакетами протоколов маршрутизации, такими как RIP или OSPF, и вычислении оптимального пути к каждой целевой сети по некоторому критерию. Для вычисления оптимального пути на графе, как того требуют протоколы состояния связей, необходимы значительные вычислительные мощности. После того как таблица маршрутизации составлена, функция продвижения пакетов происходит весьма просто – осуществляется просмотр таблицы и поиск совпадения полученного адреса с адресом целевой сети. Если совпадение есть, то пакет передается на соответствующий порт маршрутизатора. Некоторые маршрутизаторы поддерживают только функции продвижения пакетов по готовой таблице маршрутизации. Такие маршрутизаторы являются усеченными маршрутизаторами, так как для их полноценной работы требуется наличие полнофункционального маршрутизатора, у которого можно взять готовую таблицу маршрутизации. Этот маршрутизатор часто называется сервером

маршрутов. Отказ от самостоятельного выполнения функций построения таблицы маршрутизации резко удешевляет маршрутизатор и повышает его производительность. Примерами такого подхода являются маршрутизаторы NetBuilder компании 3Com, поддерживающие фирменную технологию Boundary Routing, маршрутизирующие коммутаторы Catalyst 5000 компании Cisco Systems.

3.1 Задание на лабораторную работу

По согласованию с преподавателям выбрать один тип коммуникационного оборудования и рассмотреть его варианты не менее, чем от 3-х производителей. Провести сравнительный анализ данного оборудования. Сделать выводы и рекомендации.

Отчёт должен содержать:

1. Основные характеристики каждого сетевого оборудования.
2. Описание значимости указанных характеристик.
3. Рекомендации по использованию данного оборудования.

3.2 Контрольные вопросы

1. Как влияет на производительность сети пропускная способность сетевого адаптера и пропускная способность порта концентратора?
2. Имеются ли отличия в работе сетевых адаптеров, соединяющих компьютер с коммутатором или с мостом, или с концентратором?
3. Как концентратор поддерживает резервные связи?
4. В соответствии с основной функцией концентратора – повторением сигнала -его относят к устройствам, работающим на физическом уровне модели OSI. Приведите примеры дополнительных функций концентратора, для выполнения которых концентратору требуется информация протоколов более высоких уровней?
5. Чем модульный концентратор отличается от стекового?
6. О чем говорит размер внутренней адресной таблицы коммутатора? Что произойдет, если таблица переполнится?

7. Что нужно сделать администратору сети, чтобы коммутаторы, не поддерживающие алгоритм Spanning Tree, правильно работали в сети с петлями?

8. Что произойдет, если в сети, построенной на концентраторах, имеются замкнутые контуры ?

a) сеть будет работать нормально;
b) кадры не будут доходить до адресата;
c) в сети при передаче кадра будет возникать коллизия;
d) произойдет зацикливание кадров.

9. Какие дополнительные возможности имеют коммутаторы, поддерживающие алгоритм Spanning Tree?

10. В чем отличие между резервированием связей маршрутизаторами, с одной стороны, и коммутаторами, поддерживающими алгоритм Spanning Tree, с другой стороны?

11. Почему недорогие коммутаторы, выполняющие ограниченное число функций, обычно работают по быстрому алгоритму обработки пакетов «на лету», а дорогие коммутаторы, с большим числом функций – по более медленному алгоритму буферизации пакетов?

12. Какая информация содержится в таблицах коммутаторов и маршрутизаторов?

13. В каких случаях появляется необходимость в создании виртуальных сегментов? Приведите примеры.

14. Какие из следующих утверждений верны всегда?

a) Каждый порт моста/коммутатора имеет MAC-адрес.
b) Каждый мост/коммутатор имеет сетевой адрес.
c) Каждый порт моста/коммутатора имеет сетевой адрес.
d) Каждый маршрутизатор имеет сетевой адрес.
e) Каждый порт маршрутизатора имеет MAC-адрес.
f) Каждый порт маршрутизатора имеет сетевой адрес.

4 Лабораторная работа №4. Стек протоколов TCP/IP. Создание правил маршрутизации

4.1 Теоретические сведения

Маршрутизация

Маршрутизация (Routing) –процесс определения маршрута следования информации в сетях связи.

Маршруты могут задаваться административно (статические маршруты), либо вычисляться с помощью алгоритмов маршрутизации, базируясь на информации о топологии и состоянии сети, полученной с помощью протоколов маршрутизации (динамические маршруты).

Статическими маршрутами могут быть:

- маршруты, не изменяющиеся во времени;
- маршруты, изменяющиеся по расписанию;
- маршруты, изменяющиеся по ситуации –административно в момент возникновения стандартной ситуации.

Маршрутизация в компьютерных сетях типично выполняется специальными программно-аппаратными средствами– маршрутизаторами; в простых конфигурациях может выполняться и компьютерами общего назначения, соответственно настроенными.

Маршрутизируемые протоколы

Протокол маршрутизации может работать только с пакетами, принадлежащими к одному из маршрутизируемых протоколов, например, IP, IPX или Xerox Network System, AppleTalk. Маршрутизируемые протоколы определяют формат пакетов (заголовков), важнейшей информацией из которых для маршрутизации является адрес назначения. Протоколы, не поддерживающие маршрутизацию, могут передаваться между сетями с помощью туннелей. Подобные возможности обычно предоставляют программные и некоторые аппаратные маршрутизаторы.

Программная и аппаратная маршрутизация

Первые маршрутизаторы представляли из себя специализированное ПО, обрабатывающее приходящие IP-пакеты специфичным образом. Это ПО работало на компьютерах, у которых было несколько сетевых интерфейсов, входящих в состав различных сетей (между которыми осуществляется маршрутизация). В

дальнейшем появились маршрутизаторы в форме специализированных устройств. Компьютеры с маршрутизирующим ПО называют программные маршрутизаторы, оборудование - аппаратные маршрутизаторы.

В современных аппаратных маршрутизаторах для построения таблиц маршрутизации используется специализированное ПО ("прошивка"), для обработки же IP-пакетов используется коммутационная матрица (или другая технология аппаратной коммутации), расширенная фильтрами адресов в заголовке IP-пакета.

Аппаратная маршрутизация

Есть два типа аппаратной маршрутизации: со статическими шаблонами потоков и с динамически адаптируемыми таблицами.

Статические шаблоны потоков подразумевают разделение всех входящих в маршрутизатор IP-пакетов на виртуальные потоки; каждый поток характеризуется набором признаков для пакета: IP-адресами отправителя/получателя, TCP/UDP-порт отправителя/получателя (в случае поддержки маршрутизации на основании информации 4 уровня), порт, через который пришёл пакет. Оптимизация маршрутизации при этом строится на идее, что все пакеты с одинаковыми признаками должны обрабатываться одинаково (по одинаковым правилам), при этом правила проверяются только для первого пакета в потоке (при появлении пакета с набором признаков, не укладывающимся в существующие потоки, создаётся новый поток), по результатам анализа этого пакета формируется статический шаблон, который и используется для определения правил коммутации приходящих пакетов (внутри потока). Обычно время хранения не использующегося шаблона ограничено (для освобождения ресурсов маршрутизатора). Ключевым недостатком подобной схемы является инерционность по отношению к изменению таблицы маршрутизации (в случае существующего потока изменение правил маршрутизации пакетов не будет "замечено" до момента удаления шаблона).

Динамически адаптируемые таблицы используют правила маршрутизации "напрямую", используя маску и номер сети из таблицы маршрутизации для проверки пакета и определения порта, на который нужно передать пакет. При этом изменения в таблице маршрутизации (в результате работы, например, протоколов маршрутизации/резервирования) сразу же влияют на обработку всех

ново пришедших пакетов. Динамически адаптируемые таблицы также позволяют легко реализовывать быструю проверку списков доступа.

Программная маршрутизация

Программная маршрутизация выполняется либо специализированным ПО маршрутизаторов (в случае, когда аппаратные методы не могут быть использованы, например, в случае организации туннелей), либо программным обеспечением на компьютере. В общем случае, любой компьютер осуществляет маршрутизацию своих собственных исходящих пакетов (как минимум, для разделения пакетов, отправляемых на шлюз по умолчанию и пакетов, предназначенных узлам в локальном сегменте сети). Для маршрутизации чужих IP-пакетов, а также построения таблиц маршрутизации используется различное ПО:

- Сервис RRAS (routing and remote access service) в Windows Server;
- Демоны routed, gated, quagga в Unix-подобных операционных системах (Linux, FreeBSD и т.д..).

Таблица маршрутизации

Таблица маршрутизации–электронная таблица (файл) или база данных, хранящаяся на маршрутизаторе или сетевом компьютере, описывающая соответствие между адресами назначения и интерфейсами, через которые следует отправить пакет данных до следующего маршрутизатора. Является простейшей формой правил маршрутизации.

Таблица маршутизации обычно содержит:

- адрес сети или узла назначения, либо указание, что маршрут является маршрутом по умолчанию
- маску сети назначения (для IPv4-сетей маска /32 (255.255.255.255) позволяет указать единичный узел сети)
- шлюз, обозначающий адрес маршрутизатора в сети, на который необходимо отправить пакет, следующий до указанного адреса назначения
- интерфейс (в зависимости от системы это может быть порядковый номер, GUID или символьное имя устройства)
- метрику – числовой показатель, задающий предпочтительность маршрута. Чем меньше число, тем более предпочтителен маршрут (интуитивно представляется как расстояние).

В таблице может быть один, а в некоторых операционных системах и несколько шлюзов по умолчанию. Такой шлюз используется для сетей, для которых нет более конкретных маршрутов в таблице маршрутизации.

Пример таблицы маршрутизации при четырёх интерфейсах (loopback, две сетевые карты, VPN-соединение) представлен на рисунке 4.1. Типы записей в таблице маршрутизации: маршрут до сети; маршрут до компьютера и маршрут по умолчанию.

```
===========================================================
Interface List
0x1 .  ................... MS TCP Loopback interface
0x2  .00 14 2a 8b a1 b5 . NVIDIA nForce Networking Controller
0x3 ..00 50 56 c0 00 01 .VMware Virtual Ethernet Adapter for
VMnet1
0xd0005 ...00 53 45 00 00 00 ...... WAN (PPP/SLIP) Interface
===========================================================
===========================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    89.223.67.129   89.223.67.131      20
     60.48.85.155  255.255.255.255    89.223.67.129   89.223.67.131      20
      60.48.105.1  255.255.255.255    89.223.67.129   89.223.67.131      20
    60.48.172.103  255.255.255.255    89.223.67.129   89.223.67.131      20
    60.48.203.116  255.255.255.255    89.223.67.129   89.223.67.131      20
     60.49.71.132  255.255.255.255    89.223.67.129   89.223.67.131      20
    66.36.138.228  255.255.255.255    89.223.67.129   89.223.67.131      20
    66.36.152.228  255.255.255.255    89.223.67.129   89.223.67.131      20
   74.108.102.130  255.255.255.255    89.223.67.129   89.223.67.131      20
    89.223.67.128  255.255.255.192    89.223.67.131   89.223.67.131      20
    89.223.67.131  255.255.255.255        127.0.0.1       127.0.0.1      20
   89.255.255.255  255.255.255.255    89.223.67.131   89.223.67.131      20
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
   164.77.239.153  255.255.255.255    89.223.67.129   89.223.67.131      20
     192.168.23.0    255.255.255.0     192.168.23.1    192.168.23.1      20
     192.168.23.1  255.255.255.255        127.0.0.1       127.0.0.1      20
   192.168.23.255  255.255.255.255     192.168.23.1    192.168.23.1      20
    192.168.192.0    255.255.255.0  192.168.192.251  192.168.192.251       1
  192.168.192.251  255.255.255.255        127.0.0.1       127.0.0.1      50
  192.168.192.255  255.255.255.255  192.168.192.251  192.168.192.251      50
   212.113.96.250  255.255.255.255    89.223.67.129   89.223.67.131      20
   219.95.153.243  255.255.255.255    89.223.67.129   89.223.67.131      20
        224.0.0.0        240.0.0.0    89.223.67.131   89.223.67.131      20
        224.0.0.0        240.0.0.0     192.168.23.1    192.168.23.1      20
        224.0.0.0        240.0.0.0  192.168.192.251  192.168.192.251      50
  255.255.255.255  255.255.255.255    89.223.67.131   89.223.67.131       1
  255.255.255.255  255.255.255.255     192.168.23.1    192.168.23.1       1
  255.255.255.255  255.255.255.255  192.168.192.251  192.168.192.251       1
Default Gateway:     89.223.67.129
===========================================================================
```

Рисунок 4.1 – Пример таблицы маршрутизации

Добавление маршрута в Network Emulator

Для добавления маршрута в Network Emulatore (см. рисунок 4.2) необходимо выделить объект, таблицу маршрутизации которого необходимо изменить, нажать правую кнопку мыши и выбрать «Настройки». После чего ввести маршрут и нажать «Добавить».

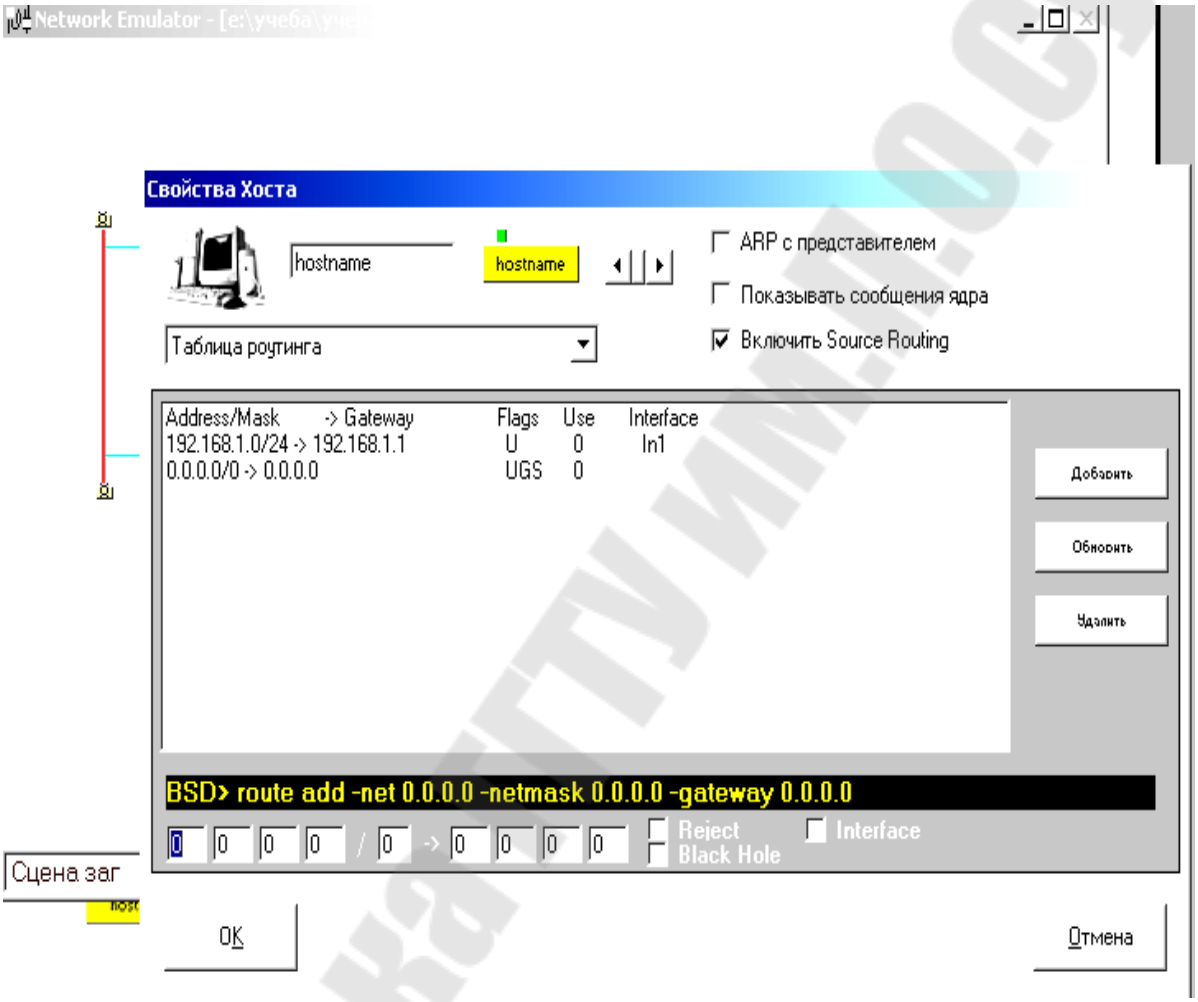


Рисунок 4.2 – Добавление маршрута

4.2 Задание на лабораторную работу

Согласно варианту (см. таблицу 4.1) смоделировать ЛВС в NetWork Emulator'e. Задать правила маршрутизации и проверить доставку пакетов между компьютерами сети.

Отчёт должен содержать схему моделируемой ЛВС с указанием адресов компьютеров в сети и правил маршрутизации для каждого компьютера и подробное описание функционирования ЛВС.

Результаты моделирования обязательно должны быть продемонстрированы на компьютере.

Таблица 4.1 – Варианты заданий

Вариант	**Топология ЛВС (протокол TCP/IP)**
1, 6, 9, 12, 19, 25	2 подсети объединённые с использованием 2 шлюзов. Между шлюзами соединение «точка-точка»
2, 13, 16, 21, 26, 30	3 подсети объединённые между собой с использованием 1 шлюза.
3, 8, 14, 18, 24, 27	2 подсети объединённые с использованием 2 шлюзов. Между шлюзами соединение с использованием общей шины.
4, 7, 10, 15, 22, 28	3 шлюза объединены по кольцу между собой. К каждому шлюзу подключены локальные подсети.
5, 11, 17, 20, 23, 29	3 подсети последовательно соединены между собой с использованием двух шлюзов

4.3 Контрольные вопросы

1. Таблица маршрутизации содержит записи о сетях назначения. Должна ли она содержать записи обо всех сетях составной сети или только о некоторых? Если только о некоторых, то о каких именно?
2. Может ли в таблице маршрутизации иметься несколько записей о маршрутизаторах по умолчанию?
3. Пусть IP-адрес некоторого узла подсети равен 198.65.12.67, а значение маски для этой подсети - 255.255.255.240. Определите номер подсети. Какое максимальное число узлов может быть в этой подсети?
4. Какое максимальное количество подсетей теоретически возможно организовать, если в вашем распоряжении имеется сеть класса С? Какое значение должна при этом иметь маска?
5. Почему даже в тех случаях, когда используются маски, в IP-пакете маска не передается?
6. Почему в записи о маршрутизаторе по умолчанию в качестве адреса сети назначения указывается 0.0.0.0 с маской 0.0.0.0?
7. Сравните функции маршрутизаторов, которые поддерживают маршрутизацию от источника, с функциями маршрутизаторов, поддерживающих протоколы адаптивной маршрутизации.
8. Какие метрики расстояния могут быть использованы в алгоритмах сбора маршрутной информации?

5 Лабораторная работа №5. Конфигурирование сетевых интерфейсов

5.1 Теоретические сведения

NET-Simulator позволяет строить виртуальные вычислительные сети из виртуальных сетевых устройств: маршрутизаторов, настольных компьютеров, концентраторов и т.п. Устройствами можно управлять при помощи интерфейса командной строки из виртуальных терминалов.

Сетевое ядро

В NET-Simulator реализованы только два уровня ISO OSI: канальный и сетевой. Таким образом NET-Simulator позволяет решать следующие образовательные задачи:

1) изучение принципов работы коммуникаторов второго и третьего уровня, пассивных концентраторов;

2) отработка практических навыков статической маршрутизации в IP-сетях;

3) изучение принципов работы протоколов канального уровня, ARP, IP4, ICMP;

4) отработка практических навыков поисков неисправностей в IP-сетях.

Физическая природа сети не учитывается. Предполагается, что пакеты канального уровня распространяются в среде аналогичной локальной сети на основе Ethernet.

На канальном уровне используется простейший Ethrnet-образный протокол, который предусматривает адресацию по 6-ти байтовым MAC-адресам. Уникальность MAC-адресов обеспечивает ядро NET-Simulator. Пакет канального протокола представляет собой объект Java и не имеет аналогов в реальных сетях.

На сетевом уровне используется ограниченная реализация IP в соответствии с RFC791. Для преобразования IP-адресов в MAC реализована служба ARP на основе широковещательных запросов.

Для работы служебных утилит, таких как ping, используется ограниченная реализация ICMP в соответствии с RFC792.

Графический интерфейс

В главном окне NET-Simulator (рисунок 5.1) отображается поле, в которое можно добавлять различные устройства из меню Устройства.

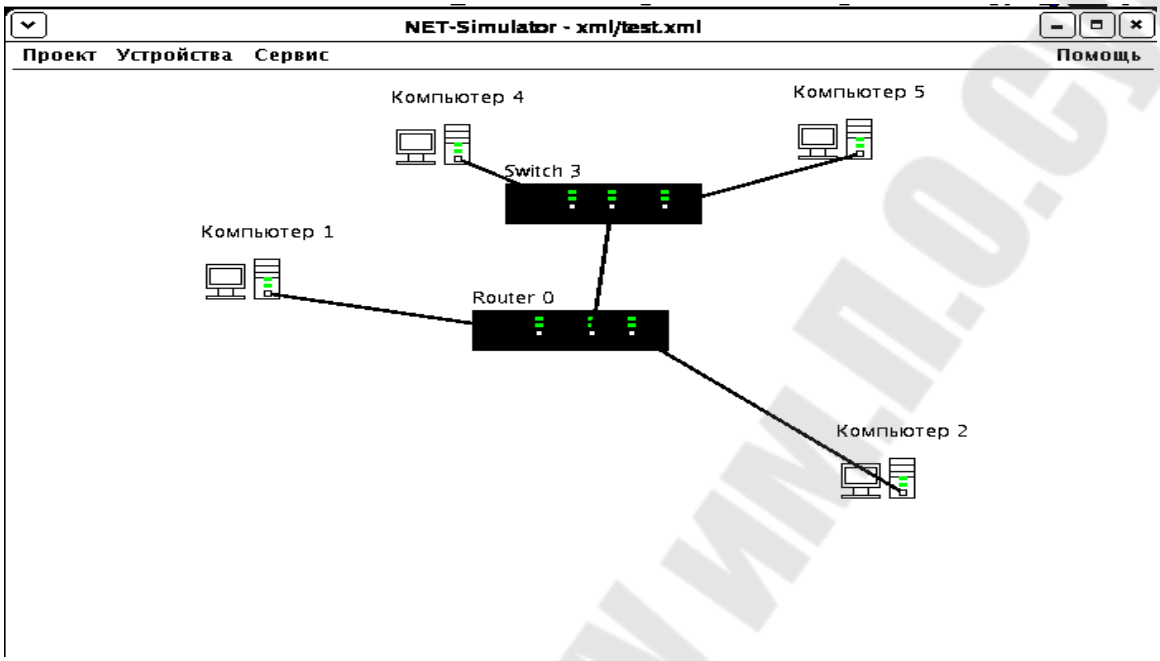


Рисунок 5.1 – Внешний вид главного окна симулятора

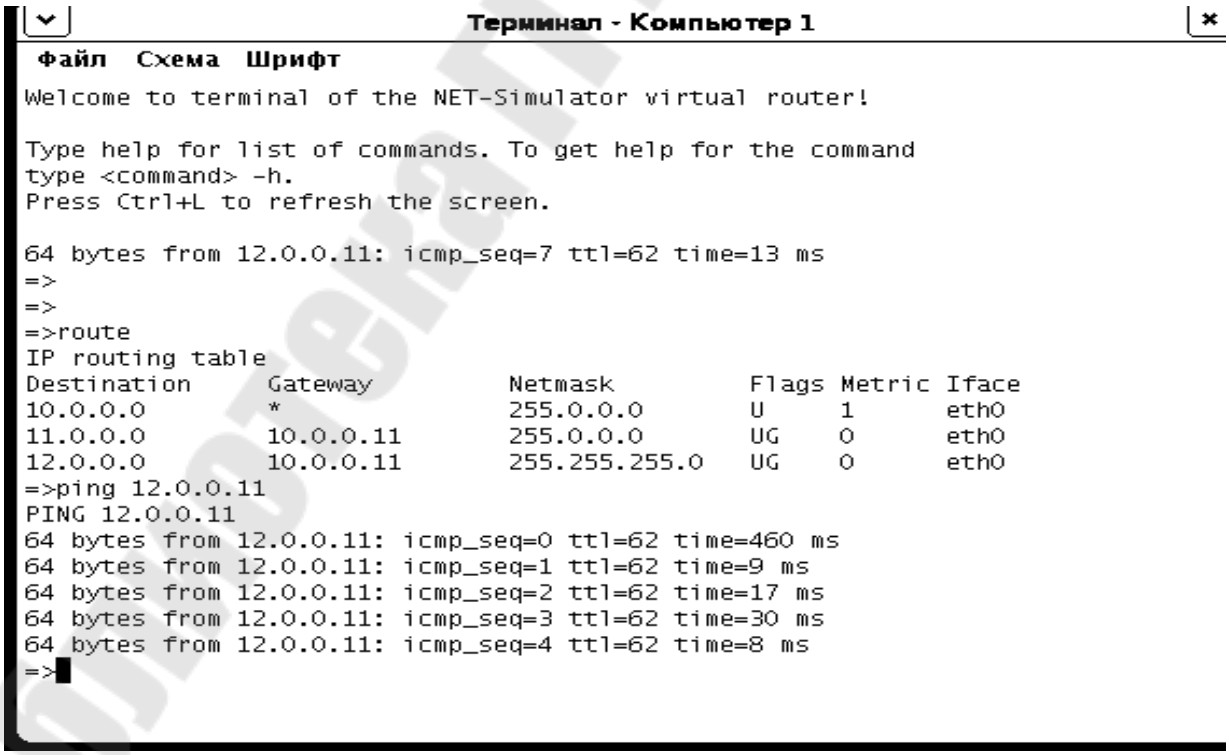


Рисунок.5.2 – Внешний вид окна настройки компьютера

Поддерживаются следующие типы устройств:

1) маршрутизатор, коммутатор третьего уровня с восемью интерфейсами и поддержкой IP4;

2) настольный компьютер – фактически маршрутизатор с одним интерфейсом;

3) концентратор (Hub) – простейшее устройство, ретранслирующее пакеты канального уровня на свои интерфейсы; не имеет терминала и соответственно никак не управляется.

4) коммутатор (Switch) – коммутатор второго уровня с восемью интерфейсами. Коммутирует пакеты канального уровня на основе таблиц MAC-адресов, по аналогии с известными алгоритмами используемыми в Ethernet-свитчах.

Устройства соединяются с помощью универсальной среды передачи данных, виртуального патчкорда. При прохождении пакета через патчкорд, он подсвечивается для визуального отслеживания активности в сети.

Вновь добавленные устройств появляются в верхнем левом углу, после чего их можно перетаскивать мышкой в удобное место. Вилки патчкордов «приклеиваются» к розеткам интерфейсов устройств. Нажатие правой кнопки мыши на устройстве открывает контекстное меню, которое позволяет просмотреть свойства, открыть терминал или удалить устройство. Двойной щелчок левой кнопкой мыши открывает терминал.

Сохранение/загрузка проектов

Проекты сохраняются в формате xml. DTD для проектов NET-Simulator находиться в каталоге dtd –net_simulator.dtd.

Виртуальные терминалы и интерфейс командной строки

Виртуальные устройства в NET-Simulator управляются при помощи интерфейса командной строки из виртуальных терминалов. Терминал устройства можно открыть двойным кликом на значке устройства или через контекстное меню. Поддерживается история команд, клавиши вверх/вниз позволяют просматривать историю команд.

Список команд доступных на данном устройстве можно посмотреть командой help. Сочетание клавиш Ctrl+L очищает терминал. Краткая справка по любой команде выводится при вызове команды с опцией -h.

Справочник команд

help

Выводит список доступных команд.

Параметры

-h –краткая справка.

route

Позволяет управлять таблицей маршрутизации устройств поддерживающих протокол IP4.

Синтаксис

route [-h] [{-add|-del} <target> [-netmask <address>] [-gw <address>] [-metric <M>] [-dev <If>]]

Параметры

Описание опций команды **route** представлено в таблице 5.1.

Таблица 5.1 – Описание параметров команды **route**

Опции	Описание
-h	Краткая справка.
target	Адрес назначения. Назначением может быть подсеть или отдельный узел в зависимости от значения маски подсети. Если маска равна 255.255.255.255 или отсутствует совсем назначением будет узел, иначе назначением будет сеть.
-add	Добавляет новый маршрут в таблицу маршрутизации.
-del	Удаляет маршрут из таблицы маршрутизации.
-dev <If>	Принудительно присоединяет маршрут к определенному интерфейсу. If –имя интерфейса.
-gw <address>	Направляет пакеты по этому маршруту через заданный шлюз. address –адрес шлюза.
-netmask <address>	Маска подсети используемая совместно с адресом назначения при добавлении маршрута. address –маска. Если маска не задана явно подразумевается 255.255.255.255.
-metric <M>	Метрика используемая в данном маршруте. M –целое число большее или равное нулю.

Если **route** вызывается без параметров, то команда выводит на экран (рисунок 5.3) таблицу маршрутизации:

```
=>route
IP routing table
Destination     Gateway     Netmask          Flags  Metric Iface
10.0.0.0        *           255.0.0.0        U      1       eth0
11.0.0.0        10.0.0.10   255.0.0.0        UG     1       eth0
192.168.120.1   10.0.0.10   255.255.255.255  UGH    1       eth0
```

Рисунок 5.3 – Вывод таблицы маршрутизации

Если маршрут не использует шлюз, вместо адреса шлюза выводиться *. Flags может содержать значение: U – маршрут активен, G – маршрут использует шлюз, H – назначением является узел.

Примеры

Примеры реализации команды **route** показаны на рисунке 5.4.

```
=>route -add 192.168.120.0 -netmask 255.255.255.0 -dev eth0
=>route
IP routing table
Destination   Gateway       Netmask         Flags Metric Iface
192.168.120.0 *            255.255.255.0    U     1       eth0
=>
=>route -add 192.168.121.10 -gw 192.168.120.10
=>route
IP routing table
Destination    Gateway      Netmask         Flags Metric Iface
192.168.120.0  *            255.255.255.0   U     1       eth0
192.168.121.10192.168.120.1 255.255.255.255 UGH   1       eth0
=>
```

Рисунок 5.4 – Примеры реализации команды route

ifconfig

Конфигурирует сетевые интерфейсы.

Синтаксис

ifconfig [-h] [-a] [<interface>] [<address>] [-broadcast <address>] [-netmask <address>] [-up|-down]

Параметры

Описание опций команды **ifconfig** представлено в таблице 5.2.

Если **ifconfig** вызывается без параметров, то команда выводит на экран (рисунок 5.5) данные о состоянии всех активных интерфейсов.

Таблица 5.2 – Описание параметров команды **ifconfig**

Опции	Описание
-h	Краткая справка.
-a	Показывать информацию о всех интерфейсах. Если данная опция отсутствует выводится информация только об активных интерфейсах.
interface	Конфигурировать или показать информацию только о заданном интерфейсе.
address	IP-адрес присваиваемый интерфейсу.
- broadcast <address>	Широковещательный адрес присваиваемый интерфейсу. address –широковещательный адрес.
-netmask <address>	Маска подсети используемая совместно с адресом. address –маска. Если маска не задана явно, маска принимается равной стандартным значения для стандартных классов подсетей A, B и C.
-up	Активирует интерфейс. При активизации интерфейса для него автоматически добавляется соответствующий маршрут в таблице маршрутизации.
-down	Деактивирует интерфейс. При деактивации интерфейса соответствующий маршрут автоматически удаляется из таблицы маршрутизации.

```
=>ifconfig
eth0 Link encap:Ethernet  HWaddr 0:0:0:0:CF:0
inet addr:192.168.120.1 Bcast:192.168.120.255
Mask:255.255.255.0
UP
RX packets:23 errors:0 dropped:0
TX packets:23 errors:0 dropped:0
RX bytes:0 TX bytes:0
```

Рисунок 5.5 – Состояние активных интерфейсов команды **ifconfig**

Примеры

Пример реализации команды **ifconfig** показаны на рисунке 5.6.

HWaddr – уникальный шестибайтовый адрес интерфейса, аналогичный MAC-адресу в Ethernet сетях. Назначается автоматически.

ping

Использует ICMP протокол что бы проверить достижимость интерфейса удаленного узла. ping посылает удаленному узлу ICMP ECHO_REQUEST и ожидает в течении определенного промежутка времени ICMP ECHO_RESPONSE. В случае получения ответа выводит данные о прохождении ICMP-пакета по сети.

```
=>ifconfig eth0 192.168.120.1 -up
=>ifconfig
eth0    Link encap:Ethernet  HWaddr 0:0:0:0:CF:0
inet addr:192.168.120.1 Bcast:192.168.120.255
Mask:255.255.255.0
UP
RX packets:0 errors:0 dropped:0
TX packets:0 errors:0 dropped:0
RX bytes:0 TX bytes:0
```

Рисунок 5.6 – Пример реализации команды **ifconfig**

Синтаксис

ping [-h] [-i <interval>] [-t <ttl>] <destination>

Параметры

Описание опций команды **ping** представлено в таблице 5.3.

Таблица 5.3 – Описание параметров команды **ping**

Опции	Описание
-h	Краткая справка.
-i <interval>	Задает частоту ICMP-запросов. interval – интервал между запросами в секундах. По умолчанию отсылается один пакет в секунду.
-t <ttl>	Задает значение атрибута Time to Live в генерируемых IP-пакетах. ttl – целое число 0-255. По умолчанию TTL равно 64.
destination	IP-адрес исследуемого узла

Примеры

Пример реализации команды **ping** показаны на рисунке 5.7.

Команда ping выводит результат исследования удаленного узла в следующем формате: 64 bytes from 192.168.120.1 – размер полученного ответа и адрес источника ответа. В NET-Simulator размер пакета имеет условное значение и всегда равен 64B. icmp_seq=0 – номер пакета. Каждый запрос содержит свой номер, как правило формируется инкрементно. ping выводит номер пакета

из каждого полученного ответа. ttl=62 – значение TTL из полученного ответа. time=48 ms – время прохождения пакетом полного маршрута (туда и обратно, round-trip time) в миллисекундах.

```
=>ping 192.168.120.1
PING 192.168.120.1
64 bytes from 192.168.120.1: icmp_seq=0 ttl=62 time=477 ms
64 bytes from 192.168.120.1: icmp_seq=1 ttl=62 time=435 ms
64 bytes from 192.168.120.1: icmp_seq=2 ttl=62 time=234 ms
64 bytes from 192.168.120.1: icmp_seq=3 ttl=62 time=48 ms
64 bytes from 192.168.120.1: icmp_seq=4 ttl=62 time=87 ms
64 bytes from 192.168.120.1: icmp_seq=5 ttl=62 time=56 ms
```

Рисунок 5.7 – Пример реализации команды **ping**

arp

Показывает ARP-таблицу устройства. Кроме того опция -r позволяет сформировать запрос для определения MAC-адреса по явно заданному IP-адресу. Эта функция обычно отсутствует в реальных устройствах, в NET-Simulator она добавлена для наглядности при изучении протоколов канального и сетевого уровня.

Синтаксис

arp [-h] [-r <IP-address> <interface>]

Параметры

Описание опций команды представлено в таблице 5.4.

Таблица 5.4 – Описание параметров команды **arp**

Опции	Описание
-h	Краткая справка.
-r <IP-address> <interface>	Прежде чем вывести ARP-таблицу программа предпринимает попытку найти MAC-адрес по явно заданному IP-адресу. IP-address – IP-адрес, для которого определяется MAC-адрес. interface – имя интерфейса в сети, применительно к подсоедию к которому будет происходить поиск.

Если arp вызывается без параметров, то команда выводит на экран ARP-таблицу:

Примеры

Пример реализации команды **arp** показан на рисунке 5.8.

```
=>arp -r 192.168.120.12 eth1
Address           HWaddress       iface
10.0.0.10         0:0:0:0:BC:0    eth0
10.0.0.11         0:0:0:0:1F:2    eth0
192.168.120.12    0:0:0:0:12:1    eth1
```

Рисунок 5.8 – Примеры реализации команды **arp**

5.2 Задание на лабораторную работу

Изучить команды операционной системы для управления и тестирования сетевых ресурсов. Выполнить задания лабораторной работы №4 с использованием командного режима (Net Simulator). Проверить спроектированную сеть.

Отчёт должен содержать перечень всех используемых команд с кратким описанием, последовательность выполненных команд с указанием полученного результата.

Результаты моделирования обязательно должны быть продемонстрированы на компьютере.

5.3 Контрольные вопросы

1. Возможности сетевого ядра программы NET-Simulator.
2. Типы устройств поддерживаемые программой NET-Simulator.
3. Команды программы NET-Simulator работы с сетью. Основные параметры.
4. Проверка наличия соединения с удалённым узлом.
5. Программные средства мониторинга и анализа использования сети в NET-Simulator.

Список использованных источников

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Ученик для вузов. 3-е изд. ./ СПб.: Питер, 2006. – 958с.

2. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. Учебник./ СПб.: Питер, 2006. – 544с.

3. Таненбаум Э. Компьютерные сети. / М. «Вильямс», 2003. – 992с.

4. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. / Москва: Техносфера, 2003. – 512с.

5. Уэнделл О. Компьютерные сети. Первый шаг. / Москва: Издательский дом «Вильямс», 2006. – 432с.

6. Сосновский О.А. Компьютерные сети и сетевые технологии. Курс лекций. / БГЭУ.: Минск, 2003. – 133с.

7. Курочка К.С. Практическое пособие по теме «Компьютерные сети» курсов «Сетевые технологии», «Компьютерные информационные технологии», «Информатика» для студентов дневного и заочного отделений. Гомель, 2005 (м/у 3105).

Осипенко Александр Николаевич

КОМПЬЮТЕРНЫЕ СЕТИ

Практикум
по одноименному курсу
для слушателей специальностей
1-40 01 74 «Web-дизайн и компьютерная графика»
и 1-40 01 73 «Программное обеспечение информационных систем»
заочной формы обучения

Подписано в печать 21.12.15.
Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».
Ризография. Усл. печ. л. 5,35. Уч.-изд. л. 5,06.
Изд. № 12.
http://www.gstu.by

Отпечатано на цифровом дуплекаторе
с макета оригинала авторского для внутреннего использования.
Учреждение образования «Гомельский государственный
технический университет имени П.О. Сухого».
246746, г. Гомель, пр. Октября, 48.