

МГТУ им. Н. Э. Баумана

Кафедра «Системы обработки информации и управления»

Методические указания к лабораторным работам по дисциплине

Сети и телекоммуникации

Для студентов 3-го курса кафедры ИУ5

Разработали:

к.т.н., доцент Галкин В.А.

ст. преподаватель Аксенов А. Н.

ст. преподаватель Антонов А. И.

Москва 2013 г.

УДК 004.71-77

Рецензент: Недашковский В. М.

Галкин Валерий Александрович, Аксёнов Андрей Николаевич,
Антонов Артём Ильич.

Проектирование и анализ вычислительных сетей. Методические указания к лабораторным работам по дисциплине «Сети и телекоммуникации»

Данная работа предназначена для обучения студентов основам проектирования, построения, настройки и анализа локальных сетей передачи данных. В процессе работы студенты знакомятся с общими теоретическими основами передачи данных, учатся выбирать необходимое оборудование как по техническим, так и по экономическим критериям, приобретают навыки проектирования сетей в таких продуктах как NetCracker и Cisco Packet Tracer, изучают основы работы в командном режиме операционной системы IOS, а также получают навыки анализа сетевого трафика.

Рекомендуется Учебно-методической комиссией НУК «Информатика и системы управления» МГТУ им. Н.Э. Баумана

Содержание

Лабораторная работа №1

«Выбор состава оборудования передачи данных системы телекоммуникации по экономическому критерию с учетом качества каналов связи».....	4
1.1. Цель работы.....	4
1.2. Необходимое оборудование.....	4
1.3. Теоретическая часть.	4
1.4. Порядок выполнения лабораторной работы.	9
1.5. Содержание отчета.	10
1.5. Контрольные вопросы.	11
1.6. Варианты заданий.	11

Лабораторная работа №2

«Оптимизация пропускной способности составляющих маршрут каналов связи по критерию минимума затрат».....	14
2.1. Цель работы.....	14
2.2. Необходимое оборудование.....	14
2.3. Теоретическая часть.	14
2.4. Пример выполнения лабораторной работы.	16
2.5. Порядок выполнения лабораторной работы.	19
2.5. Контрольные вопросы.	20

Лабораторная работа №3 и №4

«Проектирование и анализ локальных вычислительных сетей в пакете NetCracker. Сети Ethernet. Сетевое оборудование».....	22
3.1. Цель работы.....	22
3.2. Теоретические сведения.....	22
3.3. Описание программы Net Cracker.	42
3.4. Пример выполнения задания.	46
3.5. Задания.....	62
3.6. Контрольные вопросы.....	64

3.7. Варианты параметров.....	64
Лабораторные работы №5, №6 и №7	
«Проектирование и анализ локальных вычислительных сетей в пакете Cisco Packet Tracer. Адресация. Статическая и динамическая маршрутизация».....	66
4.1. Цель лабораторных работ.....	66
4.2. Теоретическая часть.	66
4.3. Введение в пакет Cisco Packet Tracer 5.3.....	72
4.4. Командный режим операционной системы IOS	78
4.5. Пример выполнения задания.	97
4.6. Задания.....	101
4.7. Контрольные вопросы.	102
4.8. Варианты заданий.	103
Лабораторная работа №8 и №9	
«Исследование протоколов сетевого и транспортного уровней IP-сетей с помощью анализатора протоколов».....	104
5.1. Цель лабораторных работ.....	104
5.2. Необходимое оборудование.....	104
5.3. Теоретическая часть.	104
5.4. Изучение программы NetInfo.	120
5.5. Изучение пакетного анализатора Wireshark.	135
5.6. Порядок выполнения работы.....	141
5.7. Варианты заданий.....	142
5.8. Контрольные вопросы.	145
6. Литература.	146

Лабораторная работа №1.

«Выбор состава оборудования передачи данных системы телекоммуникации по экономическому критерию с учетом качества каналов связи».

1.1. Цель работы.

Закрепление теоретических знаний и развитие практических навыков проектирования сетей по экономическому критерию.

1.2. Необходимое оборудование.

Персональный компьютер, пакет Microsoft Office, LibreOffice или Open Office.

1.3. Теоретическая часть.

Как известно, применение типа оборудования передачи данных определяется характером решаемых задач каждым информационно-вычислительным пунктом (ИВП). В зависимости от типа ЭВМ, характера и режима работы с информацией можно выделить следующие классы ИВП и абонентских станций и рекомендуемое для них оборудование передачи данных.

Первый класс. Главные вычислительные центры (ГВЦ), оснащенные ЭВМ высокой производительности. Рекомендуемым оборудованием передачи данных для этого класса могут служить мультиплексоры передачи данных, а также процессоры телеобработки данных (ПТД).

Второй класс. Кустовые вычислительные центры (КВЦ), оснащенные ЭВМ средней производительности, осуществляющие обмен

данными с ГВЦ в режиме сеансовой связи и имеющие возможность обмена данными с абонентскими станциями, подключенными к данному КВЦ.

Третий класс. Абонентские станции, оснащенные программируемыми абонентскими пунктами с дисковыми накопителями, предназначенные для работы с ЭВМ в режиме диалога и пакетной передачи данных.

Четвертый класс. Абонентские станции, оснащенные персональными ЭВМ и предназначенные для обмена данными с ГВЦ и КВЦ в режиме пакетной передачи и диалога, а также для решения информационно-вычислительных задач ограниченного объема.

Пятый класс. Абонентские станции, осуществляющие сбор и регистрацию больших объемов входных данных в режиме пакетной передачи на ВЗУ.

Шестой класс. Абонентские станции, осуществляющие передачу исходящей информации в режиме пакетной передачи и регистрацию входящей информации на бумаге. Оборудованием таких АС может служить АЦПУ, подключаемое к сети через модем.

Седьмой класс. Абонентские станции, работающие в режиме диалога без регистрации данных на ВЗУ. Оборудованием станций являются терминальные пульта, имеющие интерфейс RS232C для подключения к сети через модем.

Восьмой класс. Абонентские станции, работающие в режиме запрос-ответ по телеграфным каналам связи с выводом информации на печать, могут быть оснащены телеграфными аппаратами.

Исходными данными для решения задачи выбора сетевого оборудования является заданная иерархическая топология информационной сети, которая определяет количество вычислительных центров (ГВЦ и КВЦ) - **К** и подмножества абонентских станций **А_к**, относящихся к каждому из них, причем каждая абонентская станция

может быть подключена к своему центру.

$A = \{ A_1, A_2, \dots, A_k, \dots, A_m \}$ - множество абонентских станций в сети;

где m - мощность множества A .

Каждый элемент множества A - $A(ik)$, принадлежащий подмножеству абонентских станций, подключенных к k -ому ВЦ, характеризуется внутренними параметрами:

C - класс абонентской станции;

V - объем суточной исходящей информации;

W - объем суточной входящей информации;

T - продолжительность рабочего дня;

G - количество рабочих дней в году;

L - расстояние до КВЦ или ГВЦ, к которому подключается станция;

$A(ik) = (C, V, W, T, G, L)$.

Оборудование выбирается из множества типов D , разбитого на подмножества рекомендуемого оборудования для каждого l -го класса $D(l)$:

$D(l) = \{d(1), \dots, d(j), \dots, d(r)\}$, $l = 1, N_c$,

где N_c - количество классов абонентских станций.

Каждый тип оборудования обладает набором характеристик, образующих вектор $d(j)$:

$d(j) = (N, v, S, H, P_w)$, $j = 1, r$,

где N - наименование ;

v - скорость передачи по каналу связи;

S - стоимость;

H - занимаемая площадь;

P_w - потребляемая мощность;

r - количество типов оборудования передачи данных **l**-го класса.

При выборе сетевого оборудования необходимо учитывать и ряд обобщенных параметров, значения которых могут изменяться в зависимости от объемов передаваемой информации, квалификации операторов абонентских станций, принятого в сети размера информационного кадра, а также совокупность условно-постоянных параметров, значения которых остаются постоянными на протяжении длительного интервала времени: тарифы на аренду телефонных каналов связи или затраты на их прокладку, стоимость квадратного метра занимаемой оборудованием площади и единицы потребляемой им энергии, заработная плата обслуживающего персонала абонентских станций и т.п.

Задача состоит в том, чтобы выбрать из множества оборудования передачи данных **D** такое **d(j)** из **D(l)** для **A_k:i=1,N_k** и в таком количестве, чтобы обеспечить своевременную передачу объемов исходящей и входящей информации каждой абонентской станцией, и при этом суммарные приведенные затраты на приобретение и эксплуатацию оборудования всей сети были бы минимальными:

$$t(i,j) \leq T(i)_d, i=1,N_k; j=1,M;$$

где

N_k - количество абонентских станций, подключенных к **k**-ому КВЦ,
k=1,K;

t(i,j) - время, затрачиваемое **j**-ым оборудованием передачи данных **i**-ой абонентской станции на передачу требуемых объемов информации;

T(i)_d - допустимое время на передачу для **i**-ой абонентской станции;

$n(j)$ - количество сетевого оборудования j -го типа;

$C(i,k) = Kz \cdot E_n + Z_{\text{экспл}}$ - приведенные затраты на единицу оборудования i -ой абонентской станции, подключенной к k -ому КВЦ;

здесь

Kz - капитальные затраты;

E_n - нормативный коэффициент эффективности капитальных вложений - величина, обратная нормативному сроку окупаемости капитальных вложений;

$Z_{\text{экспл}}$ - эксплуатационные затраты на сеть за год.

Сформулированная задача представляет собой комбинаторную задачу.

Существенное влияние на время передачи информации оказывает качество каналов связи. При больших скоростных возможностях ОПД может оказаться, что текущее состояние канала связи не обеспечит своевременную передачу требуемых объемов информации. Затраты на аренду каналов связи существенно зависят от скоростных возможностей канала. Поэтому при решении задачи выбора оборудования будем определять минимально допустимую эффективную скорость передачи для каждого типа ОПД, и следовательно, минимизировать затраты на аренду канала связи.

Эффективная скорость передачи определяется как произведение максимально допустимой скорости оборудования передачи данных на понижающий коэффициент качества канала.

При выполнении лабораторной работы определяется критическое значение коэффициента качества канала, при котором коэффициент использования канала достигает максимального значения для того же количества выбранного оборудования передачи данных.

Под коэффициентом использования понимается отношение времени занятости канала передачей данных - $t(i,j)$ ко времени работы

абонентской станции **T(i)**д.

После определения критического значения коэффициента качества канала вычисляется минимально допустимая эффективная скорость передачи данных:

$$V_{эфф} = V_j * K_{кс}$$

В параметрах выбранного для данного абонентского пункта оборудования передачи данных значения максимальной скорости заменяются на ближайшее большее из стандартного ряда (100, 300, 600, 1200, 2400, 4800, 9600, 19200) вычисленного значения эффективной скорости передачи.

В заключении вновь решается задача выбора ОПД, но уже для найденной эффективной скорости передачи.

1.4. Порядок выполнения лабораторной работы.

Требуемое время для выполнения: 4 часа.

Изучить теоретическую часть. Получить у преподавателя индивидуальное задание. Номер варианта определяется по списку группы. Для пакета Microsoft Office запустить файл «Задача выбора ОПД.xls», для пакета LibreOffice и OpenOffice версии 2 и ниже — «ОПД OpenOffice.xls», версии 3 — «ОПД OpenOffice3.ods». В пакете должны быть разрешены макросы.

Этап 1. Выбор оборудования передачи данных для АП с параметром максимальной скорости V_j .

Оборудование для ГВЦ (АП 1) выбирается в последнюю очередь (нагрузка на ГВЦ считается автоматически и вводить ее не надо).

Ввести нагрузку на АП и выполнить действия в соответствии с приглашениями программы.

Просмотреть и перенести в отчет зависимость капитальных и приведенных затрат от типа ОПД.

Этап 2. Определение эффективной скорости передачи.

Просмотреть и перенести в отчет зависимость $K_{исп} = f(K_{кс})$;

Определить по графику $K_{исп} = f(K_{кс})$ ближайшую к $K_{кс}=1$ точку максимума и по ней $K_{кс}$, для которого $V_{эфф} = V_j * K_{кс}$. Выбрать для $V_{эфф}$ ближайшее большее значение из стандартного ряда (100, 300, 600, 1200, 2400, 4800, 9600)

Заменить значение скоростей на $V_{эфф}$ для **всех типов оборудования** (Лист Расчет. Столбец В)

Этап 3. Выбор оборудования передачи данных для АП (с параметром $V_{эфф}$).

После завершения расчетов результаты (N° АП, Тип ОПД, V_j , K , $L_{пр}$, $V_{эфф}$) зафиксировать в итоговой.

Повторить выполнение этапов 1 - 3 для всех номеров АП варианта задания. Для АП1 (ГВЦ) выполняется только этап 1. Перенести в отчет итоговую таблицу выбранного оборудования для всех АП варианта задания.

1.5. Содержание отчета.

Отчет о лабораторной работе должен содержать:

1. Для каждого АП графики зависимостей:

1.1. капитальных и приведенных затрат от типа ОПД;

- 1.2.коэфф. использования канала от $K_{кс}$ - $K_{исп}=f(K_{кс})$;
- 1.3.капитальных и приведенных затрат от типа ОПД для $V_{эфф}$.
2. Значение эффективной скорости передачи для выбранного типа ОПД.
3. Итоговую таблицу выбранного типа ОПД для каждого АП.

1.5. Контрольные вопросы.

1. Каким параметром задается качество канала связи?
2. Как вычисляется значение эффективной скорости передачи?
3. Почему уменьшаются приведенные затраты при $V_{эфф}$?
4. Как задать загрузку АП1?

1.6. Варианты заданий.

№ варианта	Номер АП	Нагрузка [Кбайт]	№ варианта	Номер АП	Нагрузка [Кбайт]
1	2	3850	16	2	3750
	7	4000		7	4050
	4	4750		4	4740
2	12	3500	17	12	3475
	15	5800		15	5780
	14	2350		14	2340
3	6	7100	18	6	7150
	13	1800		13	1785
	16	9100		16	9175

4	2	3800	19	3	6985
	11	3050		8	5300
	9	3125		16	9000
5	12	3500	20	4	4700
	7	4000		6	7150
	4	4750		9	3050
6	6	7100	21	6	7100
	7	4000		13	1800
	9	3125		9	3125
7	6	7160	22	5	3200
	13	1785		8	5250
	9	3120		10	8500
8	5	3250	23	2	3850
	8	5245		11	3000
	10	8750		14	2350
9	9	3100	24	2	3900
	11	2950		13	1800
	13	1800		16	9100
10	2	3860	25	5	3200
	13	1755		11	3000
	16	9245		14	3250
11	5	3200	26	5	3210
	11	3000		11	2975
	8	5250		8	5150
12	3	6500	27	3	6675
	7	4000		7	3985
	4	4750		4	4725

13	2	3850	28	12	3485
	15	5800		15	5750
	14	2350		14	2345
14	7	4000	29	7	3985
	8	5250		8	5150
	10	8500		10	8750
15	8	5250	30	10	8400
	9	3100		3	6790
	10	8450		15	5725

Нагрузка АП1 (ГВт) вычисляется автоматически как сумма нагрузок подключенных к нему АП, поэтому значение нагрузки для АП1 не задано.

Лабораторная работа №2.

«Оптимизация пропускной способности составляющих маршрут каналов связи по критерию минимума затрат».

2.1. Цель работы.

Закрепление теоретических знаний по курсу «Основы телекоммуникаций» и развитие практических навыков компьютерного моделирования.

2.2. Необходимое оборудование.

Персональный компьютер с установленной средой Java Runtime Environment, система для вычисления пропускной способности составляющих маршрут — каналов связи при критерии минимума затрат «ОПСК» (разработана студенткой кафедры ИУ5 Егоровой Ольгой).

2.3. Теоретическая часть.

Рассмотрим следующую модель сети передачи данных, которая состоит из N узлов коммутации и M линий связи. Предполагается, что:

- Все линии связи абсолютно надежны
- Все линии связи помехоустойчивы
- Узлы коммутации имеют бесконечную память
- Время обработки в узлах коммутации отсутствует
- Длины всех сообщений независимы и распределены по

$\frac{1}{\mu}$
показательному закону со средним значением μ [бит]

- Трафик, поступающий в сеть, состоит из сообщений, имеющих одинаковый приоритет, и образует пуассоновский поток со средним значением λ_{ij} [пакетов/сек] для сообщений, возникающих в узле i и предназначенных узлу j .

Обозначим:

$\gamma = \sum_{i=1}^N \sum_{j=1}^N \lambda_{ij}$ -полный трафик в сети, т.е. полное число пакетов в секунду, поступающих в сеть (и покидающих ее).

Каждая линия связи состоит из единственного дуплексного канала связи с пропускной способностью C_k

При обозначении d_k – стоимость единицы канальной емкости, стоимость канала будет $C_k d_k$

А так же будем иметь в виду, что увеличение пропускной способности уменьшает среднюю загрузку в сети, но увеличивает стоимость. Пропускная способность канала ограничена.

Введем некоторые упрощения:

1. Все очереди связываются с линиями, выходящими из узла (иначе говоря, со входом в каждый канал).
2. С каждой k -ой линией будем сопоставлять среднюю задержку этой линии T_k (ожидание обслуживания + время передачи)

Тогда, если T – средняя задержка пакета, то γT – среднее число пакетов, находящихся в сети. Если просуммируем по всем каналам в сети, то получим:

$$\gamma T = \sum_{k=1}^N \lambda_k T_k$$

или формулу Л. Клейнрока:

$$T = \frac{1}{\gamma} \sum_{k=1}^N \lambda_k T_k$$

В данном случае сеть очередей сводится к модели, впервые изученной Джексоном, в которой каждая линия связи рассматривается

как независимая СМО типа М/М/1.

На вход k -ой очереди поступает пуассоновский поток пакетов с интенсивностью λ_k пакетов/с. Средняя задержка в этом канале выражается формулой:

$$T_k = \frac{1}{\mu C_k - \lambda_k}$$

Подставляя данную формулу в предыдущую, получаем:

$$T = \frac{1}{\gamma} \sum \frac{\lambda_k}{\mu C_k - \lambda_k}$$

Поиск минимума затрат D (с учетом время обработки пакета в узле и задержку при распространении сигнала по линии связи) и соответствующих ему значений C_k осуществляется с помощью метода неопределенных множителей Лагранжа. При этом делается допущения, что стоимость канала линейно зависит от его емкости.

Для заданного потока в канале необходима определенная минимальная пропускная способность - $\frac{\lambda_k}{\mu}$. Если каналы будут обладать в точности такой емкостью, очереди на передачу бесконечно возрастут, что означает наступление состояния насыщения сети.

Поэтому, для определения предельных параметров сети значение пропускной способности канала должно превышать значение минимальной емкости на один шаг и выбирается

1. либо из стандартного ряда аналоговых линий 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 бит/с;
2. либо кратно 64 кбит/с для цифровых (до 2048 кбит/с).

2.4. Пример выполнения лабораторной работы.

Программа OPSK написана на Java, и, следовательно, может быть запущена на любой (практически) платформе: например Windows, Mac

OS, Linux, Solaris и т.д. Для запуска программы необходима виртуальная машина Java, которая входит в состав Java Runtime Environment (JRE), распространяемого компанией SUN, эта виртуальная машина уже установлена на большинстве компьютеров, но если она не установлена загрузить её можно с сайта: <http://java.com/ru/download/index.jsp>.

В лабораторной работе нужно определить предельные параметры сети (средняя длина пакета, канальные емкости, среднее время задержки пакета в сети), при которых сохраняется единственный маршрут из узла-источка в узел-сток.

После запуска программы OPSK и выбора из меню «Файл»-> «Новая модель», вы получите рабочую область для создания модели сети. Используя компоненты «Узел» и «Канал связи» постройте граф сети, изображенный на рисунке 1. В окне «Свойства» задайте параметры в соответствии с таблицей 1, для этого кликом левой клавиши мыши выделите последовательно каждый из каналов связи сети. Последовательно выбрав 1 и 4 узлы, укажите в контекстном меню узел исток – 1, а узел сток – 4. Кликнув на свободном поле рабочей области, в окне «Свойства» задайте обобщенные параметры сети в соответствии с рис.2.1.

Таблица 2.1. Характеристики каналов связи

Канал связи	Длина	Интенсивность
1-3	100	3.0
1-5	1	2000
2-1	1	1.0
3-4	50	1200
4-2	1	1.0
5-3	1	1.0
5-2	1	1.0
5-4	1	1.0

Граф сети и параметры сети будут выглядеть следующим образом (Рис.2.1.):

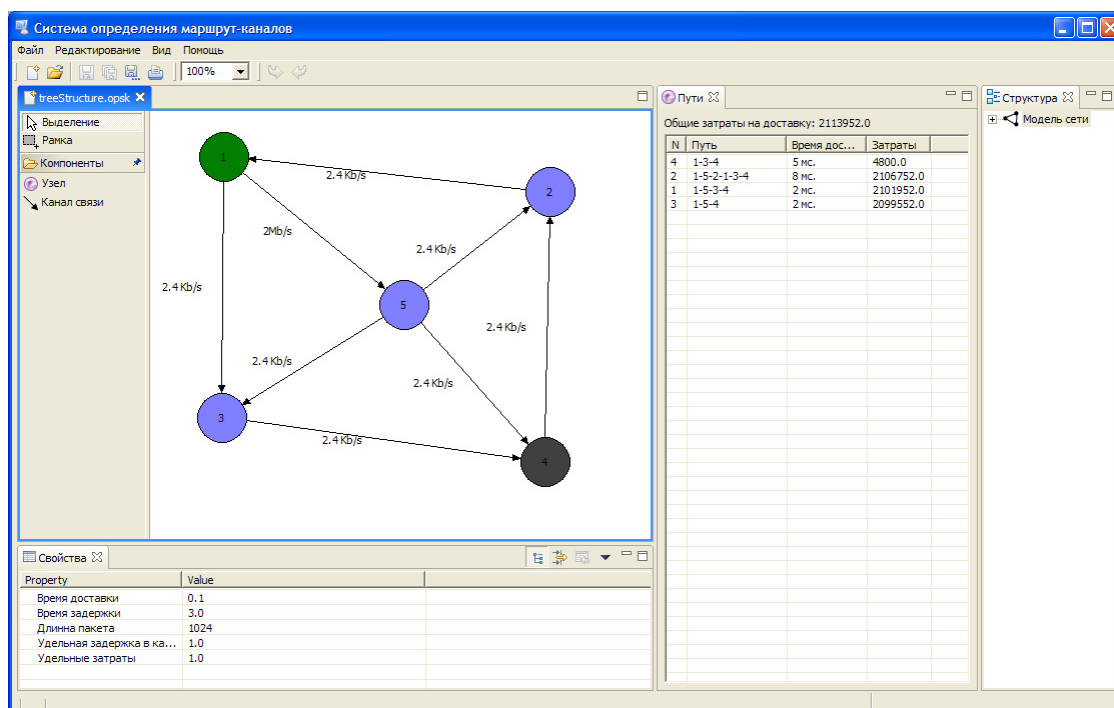


Рис.2.1. Система определения маршрут-каналов

После этого система автоматически заполнит таблицу «Пути», в которой будут указаны возможные пути, время прохождения пакета по каждому из путей (время доставки) и стоимость путей (затраты). Так же вы можете увидеть совокупную стоимость всей сети (общие затраты на доставку). Отчет можно сохранить в формате *.scv или в формате *.xml, там будут сохранены данные только из таблицы «Пути». Модель самой сети можно распечатать в pdf-формате или прямо вывести на принтер. Согласно заданию, мы должны получить единственный маршрут из узла-источка в узел-сток, попутно указав 2-3 итерации. Для данного примера опущены итерации, а приведен лишь конечный результат. Если изменить длину пакета на 1728, то получим следующий результат:

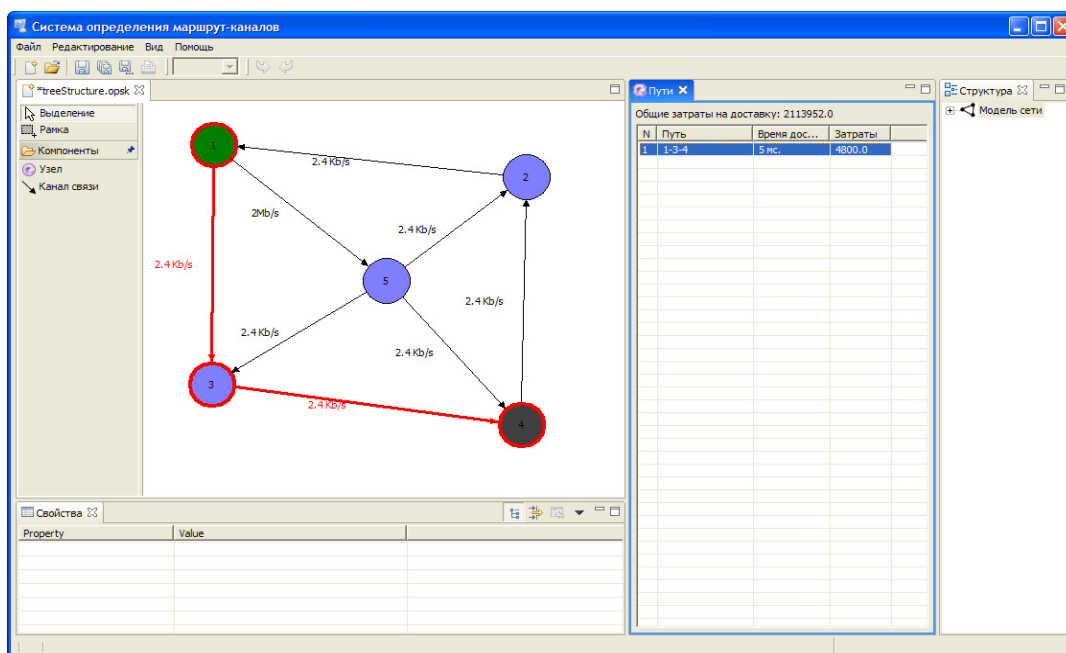


Рис.2.2. Предельные характеристики сети

Это значение длины пакета является предельным для данной сети с заданными ранее обобщенными параметрами, так как если мы увеличим это значение на 32 бита, то с заданными временными характеристиками в сети не будет ни одного пути, по которому мог бы пройти пакет размером 1760 бит.

2.5. Порядок выполнения лабораторной работы.

1. Ознакомится до конца с методическими указаниями по выполнению данной лабораторной работы.
2. Внимательно прочитать теоретическую часть методических указаний.
3. Запустить программу OPSK.
4. Изучить и выполнить приведенный пример.
5. Создать рисунок сети в виде графа произвольной топологии и связности с числом узлов не более 10 (оптимально 7-8 узлов), определив узел исток (исходная точка маршрута) и узел сток (конечная точка маршрута).

6. Задать параметры сети.
7. Задать характеристики каждого канала связи.
8. Определить возможные маршруты из узла истока в узел сток.
9. Изменяя параметры сети (включая значение удельных затрат на канальную емкость), добиться, чтобы в сети существовал один единственный маршрут из узла истока в узел сток. Далее, продолжая насыщать сеть, изменять параметры до тех пор, пока не найдутся критические значения.
10. Зафиксировать 2-3 шага итерации с указанием возможных маршрутов или их числа (для большого числа возможных маршрутов).
11. Для предельных параметров сети зафиксировать и представить в отчете взвешенный граф сети с указанием значений канальных емкостей, интенсивности и расстояния. Предельные параметры сети, включая затраты, указать в таблице. Отчет должен содержать так же промежуточные маршруты на каждом шаге итерации с указанием затрат.
12. Составить отчет по лабораторной работе.
13. Ответить на контрольные вопросы.
14. Защитить лабораторную работу.

2.5. Контрольные вопросы.

1. Как влияет удельная задержка в канале на число возможных маршрутов?
2. Как влияет задержка в узле на среднее время доставки сообщения?
3. Перечислите варьируемые обобщенные параметры сети?
4. Как влияет средняя длина пакета на количество возможных маршрутов?
5. Чем определяются затраты на доставку сообщения по

определенному маршруту?

Лабораторная работа №3 и №4.

«Проектирование и анализ локальных вычислительных сетей в пакете NetCracker. Сети Ethernet. Сетевое оборудование».

3.1. Цель работы.

Закрепление теоретических знаний в области конструирования и исследования характеристик сетей ЭВМ. Изучение программы Net Cracker Professional 4.1, а также приобретение практических навыков проектирования и моделирования работы сети, а также оценки принятых проектных решений.

С помощью программы Net Cracker Professional 4.1 необходимо построить модель вычислительной сети заданной топологии. В соответствии с топологией сети произвести подбор необходимого сетевого оборудования конкретного производителя в базе данных программы.

Задать сетевой трафик между компьютерами и произвести анализ полученных результатов. Добиться безошибочной работы модели.

3.2. Теоретические сведения.

Сети часто условно делят на три большие категории: *глобальные сети* (WAN, Wide Area Network), *городские сети* (MAN, Metropolitan Area Network) и *локальные сети* (LAN, Local Area Network). В нашей стране локальные сети распространены гораздо больше, чем городские или глобальные. Традиционное русское сокращение для локальных сетей — ЛВС (локальная вычислительная сеть).

- Глобальные сети позволяют организовать взаимодействие между абонентами на больших расстояниях. Эти сети работают на относительно низких скоростях и могут вносить значительные задержки в передачу информации. Протяженность глобальных сетей может составлять тысячи километров. Поэтому они так или иначе интегрированы с сетями масштаба страны.
- Городские сети позволяют взаимодействовать на территориальных образованиях меньших размеров и работают на скоростях от средних до высоких. Они меньше замедляют передачу данных, чем глобальные, но не могут обеспечить взаимодействие на больших расстояниях. Протяженность городских сетей находится в пределах от десятков до сотен километров.
- Локальные вычислительные сети обеспечивают наивысшую скорость обмена информацией между компьютерами. Типичная локальная сеть занимает пространство в одно здание. Протяженность локальных сетей составляет около одного километра. Их основное назначение состоит в объединении пользователей для совместной работы. Такие сети организуются внутри здания, этажа или комнаты.

Механизмы и скорости передачи данных в локальных и глобальных сетях существенно отличаются.

Кроме разницы в скорости передачи данных, между этими категориями сетей существуют и другие отличия. В локальных сетях каждый компьютер имеет сетевой адаптер, который соединяет его со средой передачи. Городские сети содержат активные коммутирующие устройства, а глобальные сети обычно состоят из групп мощных маршрутизаторов пакетов, объединенных каналами связи. Кроме того, сети могут быть частными, корпоративными, а также сетями общего пользования.

Основная задача корпоративной сети заключается в обеспечении передачи информации между различными приложениями, используемыми в организации. Под *приложением*, понимается программное обеспечение, которое непосредственно и нужно пользователю, например, базы данных, электронная почта и т. д. Корпоративная сеть позволяет взаимодействовать приложениям, зачастую расположенным в географически различных областях, и обеспечивает доступ к ним удаленных пользователей. На рис. 2.1 показана обобщенная функциональная схема корпоративной сети.



Рис. 3.1. Обобщенная схема корпоративной сети

Успешная работа многих организаций и компаний сегодня напрямую зависит от средств коммуникаций. Большую роль в деловой жизни стали играть Internet и мультимедиа. А успешно применять современные информационные технологии позволяют только современные программные и технические средства. Очень важно сделать правильный стратегический выбор пути развития сети своего предприятия. Для этого необходимо иметь всю информацию о современных сетевых технологиях, знать их возможности и уметь оценивать стоимость.

АТМ-технология. С точки зрения стратегии развития технология АТМ представляется одной из наиболее перспективных. Она, безусловно,

сможет удовлетворить запросы большинства пользователей в обозримом будущем. С учетом того, что АТМ не стоит на месте, а постоянно развивается, границы этого будущего отодвигаются все дальше и дальше. Только богатые функциональные возможности этой технологии позволяют в полной мере использовать существующую сетевую инфраструктуру. В данном случае под словом инфраструктура понимаются магистральные каналы связи, огромное количество локальных сетей и сетевое оборудование.

Широкому внедрению технологии АТМ способствует широчайший спектр предлагаемого оборудования — для построения магистрали сети, поддержки рабочих групп, доступа к глобальным и локальным сетям, сетевые адаптеры. Широкая номенклатура позволяет разработчикам гибко реагировать на все пожелания заказчиков при создании сети и решать задачу любой сложности — от подключения рабочей группы до создания магистрали. Однако необходимо помнить, что установка АТМ достаточно дорогостоящее мероприятие.

Ethernet – Стандарт организации локальных сетей (ЛВС), описанный в спецификациях IEEE и других организаций. IEEE 802.3. Ethernet использует полосу 10Мбит/с и метод доступа к среде МДКН/ОС (CSMA/CD). Наиболее популярной реализацией Ethernet является 10Base-T. Развитием технологии Ethernet является Fast Ethernet (100 Мбит/с) и Gigabit Ethernet (1000 Мбит/с).

Token Ring – Спецификация локальной сети, стандартизованная в IEEE 802.5. Кадр управления (supervisory frame), называемый также маркером (token), последовательно передается от станции к соседней. Станция, которая хочет получить доступ к среде передачи, должна ждать получения кадра и только после этого может начать передачу данных.

Модель OSI

Эталонная модель OSI, иногда называемая стеком OSI представляет собой 7-уровневую сетевую иерархию разработанную Международной организацией по стандартам (International Standardization Organization - ISO). Эта модель содержит в себе по сути 2 различных модели:

1. горизонтальную модель на базе протоколов, обеспечивающую механизм взаимодействия программ и процессов на различных машинах
2. вертикальную модель на основе услуг, обеспечиваемых соседними уровнями друг другу на одной машине

В горизонтальной модели двум программам требуется общий протокол для обмена данными. В вертикальной - соседние уровни обмениваются данными с использованием интерфейсов API.

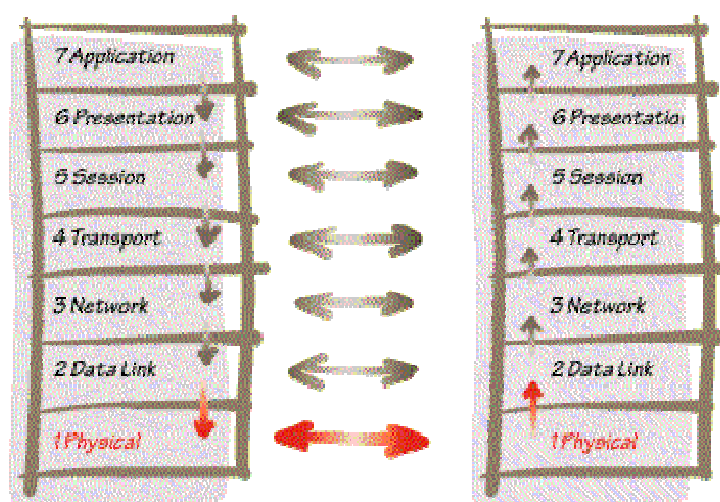


Рисунок 3.2 Модель OSI

Уровень 1, физический

Физический уровень получает пакеты данных от вышележащего канального уровня и преобразует их в оптические или электрические сигналы, соответствующие 0 и 1 бинарного потока. Эти сигналы посылаются через среду передачи на приемный узел. Механические и электрические/оптические свойства среды передачи определяются на

физическом уровне и включают:

- Тип кабелей и разъемов
- Разводку контактов в разъемах
- Схему кодирования сигналов для значений 0 и 1

К числу наиболее распространенных спецификаций физического уровня относятся:

- EIA-RS-232-C, CCITT V.24/V.28 - механические/электрические характеристики несбалансированного последовательного интерфейса.
- EIA-RS-422/449, CCITT V.10 - механические, электрические и оптические характеристики сбалансированного последовательного интерфейса.
- IEEE 802.3 -- Ethernet
- IEEE 802.5 -- Token ring

Уровень 2, каналный

Канальный уровень обеспечивает создание, передачу и прием кадров данных. Этот уровень обслуживает запросы сетевого уровня и использует сервис физического уровня для приема и передачи пакетов. Спецификации IEEE 802.x делят каналный уровень на два подуровня: управление логическим каналом (LLC) и управление доступом к среде (MAC). LLC обеспечивает обслуживание сетевого уровня, а подуровень MAC регулирует доступ к разделяемой физической среде.

Наиболее часто используемые на уровне 2 протоколы включают:

- HDLC для последовательных соединений
- IEEE 802.2 LLC (тип I и тип II) обеспечивают MAC для сред 802.x
- Ethernet
- Token ring
- FDDI
- X.25

- Frame relay

Уровень 3, сетевой

Сетевой уровень отвечает за деление пользователей на группы. На этом уровне происходит маршрутизация пакетов на основе преобразования MAC-адресов в сетевые адреса. Сетевой уровень обеспечивает также прозрачную передачу пакетов на транспортный уровень.

Наиболее часто на сетевом уровне используются протоколы:

- IP - протокол Internet
- IPX - протокол межсетевого обмена
- X.25 (частично этот протокол реализован на уровне 2)
- CLNP - сетевой протокол без организации соединений

Уровень 4, транспортный

Транспортный уровень делит потоки информации на достаточно малые фрагменты (пакеты) для передачи их на сетевой уровень.

Наиболее распространенные протоколы транспортного уровня включают:

- TCP - протокол управления передачей
- NCP - Netware Core Protocol
- SPX - упорядоченный обмен пакетами
- TP4 - протокол передачи класса 4

Уровень 5, сеансовый

Сеансовый уровень отвечает за организацию сеансов обмена данными между оконечными машинами. Протоколы сеансового уровня обычно являются составной частью функций трех верхних уровней модели.

Уровень 6, уровень представления

Уровень представления отвечает за возможность диалога между приложениями на разных машинах. Этот уровень обеспечивает преобразование данных (кодирование, компрессия и т.п.) прикладного уровня в поток информации для транспортного уровня. Протоколы уровня представления обычно являются составной частью функций трех верхних уровней модели.

Уровень 7, прикладной

Прикладной уровень отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и управление сетью.

К числу наиболее распространенных протоколов верхних уровней относятся:

- FTP - протокол переноса файлов
- TFTP - упрощенный протокол переноса файлов
- X.400 - электронная почта
- Telnet
- SMTP - простой протокол почтового обмена
- CMIP - общий протокол управления информацией
- SNMP - простой протокол управления сетью
- NFS - сетевая файловая система
- FTAM - метод доступа для переноса файлов

Стандарты локальных сетей IEEE 802.

IEEE (Institute of Electrical and Electronics Engineers) является профессиональной организацией (США), определяющей стандарты, связанные с сетями и другими аспектами электронных коммуникаций. Группа IEEE 802.X содержит описание сетевых спецификаций и содержит стандарты, рекомендации и информационные документы для сетей и

телекоммуникаций.

Публикации IEEE являются результатом работы различных технических, исследовательских и рабочих групп.

Рекомендации IEEE связаны главным образом с 2 нижними уровнями модели OSI - физическим и канальным. Эти рекомендации делят канальный уровень на 2 подуровня нижний - MAC (управление доступом к среде) и верхний - LLC (управление логическим каналом).

Часть стандартов IEEE (802.1 - 802.11) была адаптирована ISO (8801-1-8802-11, соответственно), получив статус международных стандартов. В литературе, однако, гораздо чаще упоминаются исходные стандарты, а не международные (IEEE 802.3, а не ISO/IEC 8802-3).

Ниже приведено краткое описание стандартов IEEE 802.X:

802.1 - задает стандарты управления сетью на MAC-уровне, включая алгоритм Spanning Tree. Этот алгоритм используется для обеспечения единственности пути (отсутствия петель) в многосвязных сетях на основе мостов и коммутаторов с возможностью его замены альтернативным путем в случае выхода из строя. Документы также содержат спецификации сетевого управления и межсетевого взаимодействия.

802.2 - определяет функционирование подуровня LLC на канальном уровне модели OSI. LLC обеспечивает интерфейс между методами доступа к среде и сетевым уровнем. Прозрачные для вышележащих уровней функции LLC включают кадрирование, адресацию, контроль ошибок. Этот подуровень используется в спецификации 802.3 Ethernet, но не включен в спецификацию Ethernet II.

802.3 - описывает физический уровень и подуровень MAC для сетей с немодулированной передачей (baseband networks), использующих шинную топологию и метод доступа CSMA/CD. Этот стандарт был разработан совместно с компаниями Digital, Intel, Xerox и весьма близок к стандарту Ethernet. Однако стандарты Ethernet II и IEEE 802.3 не полностью идентичны и для обеспечения совместимости разнотипных

узлов требуется применять специальные меры. 802.3 также включает технологии Fast Ethernet (100BaseTx, 100BaseFx, 100BaseFl).

802.5 - описывает физический уровень и подуровень MAC для сетей с кольцевой топологией и передачей маркеров. Этому стандарту соответствуют сети IBM Token Ring 4/16 Мбит/с.

802.8 - отчет TAG по оптическим сетям. Документ содержит обсуждение использования оптических кабелей в сетях 802.3 - 802.6, а также рекомендации по установке оптических кабельных систем.

802.9 - отчет рабочей группы по интеграции голоса и данных (IVD). Документ задает архитектуру и интерфейсы устройств для одновременной передачи данных и голоса по одной линии. Стандарт 802.9, принятый в 1993 году, совместим с ISDN и использует подуровень LLC, определенный в 802.2, а также поддерживает кабельные системы UTP (неэкранированные кабели из скрученных пар).

802.10 - в этом отчете рабочей группы по безопасности ЛВС рассмотрены вопросы обмена данными, шифрования, управления сетями и безопасности в сетевых архитектурах, совместимых с моделью OSI.

802.11 - имя рабочей группы, занимающейся спецификаций 100BaseVG Ethernet 100BaseVG. Комитет 802.3, в свою очередь, также предложил спецификации для Ethernet 100 Мбит/с

Отметим, что работа комитета 802.2 послужила базой для нескольких стандартов (802.3 - 802.6, 802.12). Отдельные комитеты (802.7 - 802.11) выполняют в основном информационные функции для комитетов, связанных с сетевыми архитектурами.

Отметим также, что разные комитеты 802.X задают разный порядок битов при передаче. Например, 802.3 (CSMA/CD) задает порядок LSB, при котором передается сначала наименее значимый бит (младший разряд), 802.5 (token ring) использует обратный порядок - MSB, как и ANSI X3T9.5 - комитет, отвечающий за архитектурные спецификации FDDI. Эти два варианта порядка передачи известны как "little-endian" (канонический) и

"big-endian" (неканонический), соответственно. Эта разница в порядке передачи имеет существенное значение для мостов и маршрутизаторов, связывающих различные сети.

Сетевые протоколы.

Сетевой протокол есть формат описания передаваемых сообщений и правила, по которым происходит обмен информацией между двумя или несколькими системами.

TCP/IP (Transmission Control Protocol/Internet Protocol - протокол управления передачей/протокол Internet) известен также, как стек протоколов Internet (Internet Protocol Suite). Данный стек протоколов используется в семействе сетей Internet и для объединения гетерогенных сетей.

IPX/SPX - Internet Packet eXchange/Sequenced Packet eXchange. IPX используется в качестве основного протокола в сетях Novell NetWare для обмена данными между узлами сети и приложениями, работающими на различных узлах. Протокол SPX содержит расширенный по сравнению с IPX набор команд, позволяющий обеспечить более широкие возможности на транспортном уровне. SPX обеспечивает гарантированную доставку пакетов.

NetBEUI - NetBIOS Extended User Interface. Транспортный протокол, используемый Microsoft LAN Manager, Windows for Workgroups, Windows NT и других сетевых ОС.

Сетевое оборудование

Сетевой адаптер (Network Interface Card, NIC) - это периферийное устройство компьютера, непосредственно взаимодействующее со средой передачи данных, которая прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надежного обмена двоичными данными,

представленными соответствующими электромагнитными сигналами, по внешним линиям связи. Как и любой контроллер компьютера, сетевой адаптер работает под управлением драйвера операционной системы и распределение функций между сетевым адаптером и драйвером может изменяться от реализации к реализации.

Сетевой концентратор (hub). В локальных сетях нашли применение топологии различных типов. Наряду с широко распространенной «шиной» применяются топологии «пассивная звезда» и «дерево». Все типы топологий могут использовать репитеры и пассивные концентраторы для объединения разных сегментов сети. Основное требование к данным топологиям — отсутствие петель (замкнутых контуров).

Если сети на базе спецификаций 10BASE-2 или 10BASE-5 имеют небольшие размеры, то вполне можно обойтись без концентраторов. Но концентраторы обязательно должны применяться для спецификации 10BASE-T, имеющей топологию «пассивная звезда».

Для подключения к сети удаленных групп могут быть использованы концентраторы с дополнительным волоконно-оптическим портом. Существуют три разновидности реализации такого порта:

- вставляемый в гнездо расширения slide-in-микротрансивер,
- вставляемый в гнездо разъема AUI навесной микротрансивер,
- постоянный оптический порт.

Оптические концентраторы применяются в качестве центрального устройства распределенной сети с большим количеством отдельных удаленных рабочих станций и небольших рабочих групп. Порты такого концентратора выполняют функции усилителей и осуществляют полную регенерацию пакетов. Существуют концентраторы с фиксированным количеством подключаемых сегментов, но некоторые типы концентраторов имеют модульную конструкцию, что позволяет гибко подстраиваться к существующим условиям. Чаще всего концентраторы и

репитеры представляют собой автономные блоки с отдельным питанием.

Для технологии Fast Ethernet определены два класса концентраторов:

1. Концентраторы первого класса преобразуют приходящие из сегментов сигналы в цифровую форму. И только после этого передают их во все другие сегменты. Это позволяет подключать к таким концентраторам сегменты, выполненные по разным спецификациям: 100BASE-TX, 100BASE-T4 или 100BASE-FX.
2. Концентраторы второго класса производят простое повторение сигналов без преобразования. К такому концентратору можно подключать сегменты только одного типа.

Мосты (bridge) имеют много отличий от повторителей. Повторители передают все пакеты, а мосты только те, которые необходимы. Если пакет не нужно передавать в другой сегмент, он фильтруется. Для мостов существуют многочисленные алгоритмы (правила) передачи и фильтрации пакетов на канальном уровне. Минимальным требованием является фильтрация пакетов по MAC-адресу получателя. Другим важным отличием мостов от повторителей является то, что сегменты, подключенные к повторителю, образуют одну разделяемую среду, а сегменты, подключенные к каждому порту моста, образуют свою среду. Следовательно, мост обеспечивает преимущества как с точки зрения расширения сети, так и обеспечения большей полосы для каждого пользователя. В первых сетях Ethernet использовалась шинная топология на основе коаксиального кабеля, а для расширения сетей применялись 2-портовые повторители или мосты. Технология 10Base-T привела к трансформации топологии сетей от шинной магистрали к организации соединений типа «звезда». Требования к повторителям и мостам для таких сетей существенно изменились по сравнению с простыми двухпортовыми устройствами для сетей с шинной топологией. Современные мосты и повторители представляют собой сложные

многопортовые устройства. Мосты позволяют сегментировать сети на меньшие части, в которых общую среду разделяет небольшое число пользователей.

Маршрутизаторы (Router), подобно мостам, также позволяют сегментировать сети на сетевом уровне, фильтруя и пересылая сетевой трафик на основе алгоритмов и правил, существенно отличающихся от тех, что используются мостами. Сегоднешние модульные концентраторы (повторители) обеспечивают организацию нескольких сегментов, каждый из которых предоставляет пользователям отдельную разделяемую среду. Некоторые концентраторы разрешают программным путем разделять порты устройства на независимые сегменты, реализуя тем самым функции моста. Такая возможность называется переключением портов. Переключение портов обеспечивает администратору сети высокую гибкость организации сегментов, позволяя переносить порты из одного сегмента в другой программными средствами. Эта возможность особенно полезна для распределения нагрузки между сегментами сети и снижения расходов, связанных с подобными операциями.

Коммутаторы (Switch) подобно мостам и маршрутизаторам позволяют сегментировать сети. Как и мосты, коммутаторы являются устройствами канального уровня и передают пакеты между портами на основе MAC-адреса получателя, включенного в каждый пакет. Реализация коммутаторов обычно отличается от мостов в части возможности организации одновременных соединений между любыми парами портов устройства, что значительно расширяет суммарную пропускную способность сети.

Трафик

Характеристики трафика

Можно выделить две характеристики трафика — единица данных и способ упаковки этих единиц. Единицей данных может быть: бит, байт,

октет, сообщение, блок. Они упаковываются в файлы, пакеты, кадры, ячейки. Они могут также передаваться без упаковки.

Скорость измеряется в единицах данных за единицу времени. Например, пакеты в секунду, байты в секунду, транзакции в минуту и т. д. Скорость также определяет время, требуемое для передачи единицы данных по сети.

Реальный размер передаваемых по сети данных складывается из непосредственно данных и необходимого информационного обрамления, составляющего накладные расходы на передачу. Многие технологии устанавливают ограничения на минимальный и максимальный размеры пакета. Так, например, для технологии X.25 максимальный размер пакета составляет 4096 байт, а в технологии Frame Relay максимальный размер кадра составляет 8096 байт.

Можно выделить четыре наиболее общие характеристики трафика:

- «взрывообразность»
- терпимость к задержкам
- время ответа
- емкость и пропускная способность

Эти характеристики с учетом маршрутизации, приоритетов, соединений и т. д. как раз и определяют характер работы приложений в сети.

«Взрывообразность» характеризует частоту посылки трафика пользователем. Чем чаще пользователь посылает свои данные в сеть, тем она больше. Пользователь, который посылает данные регулярно, в одном темпе, сводит показатель «взрывообразности» практически к нулю. Этот показатель можно определить отношением максимального (пикового) значения трафика к среднему. Например, если максимальный объем пересылаемых данных в часы пик составляет 100 Мбит/с, а средний объем — 10 кбит/с, показатель «взрывообразности» будет равен 10.

Терпимость к задержкам характеризует реакцию приложений на все виды задержек в сети. Например, приложения, обрабатывающие финансовые транзакции в реальном масштабе времени, не допускают задержек. Большие задержки могут привести к неправильной работе таких приложений.

Приложения сильно различаются по допустимому времени задержки. Есть приложения, работающие в реальном времени (видеоконференции) — там время задержки должно быть крайне малым. С другой стороны, встречаются приложения, терпимые к задержкам в несколько минут или даже часов (электронная почта и пересылка файлов). На рис. 2.3 показано, из чего составляется общее время реакции системы.

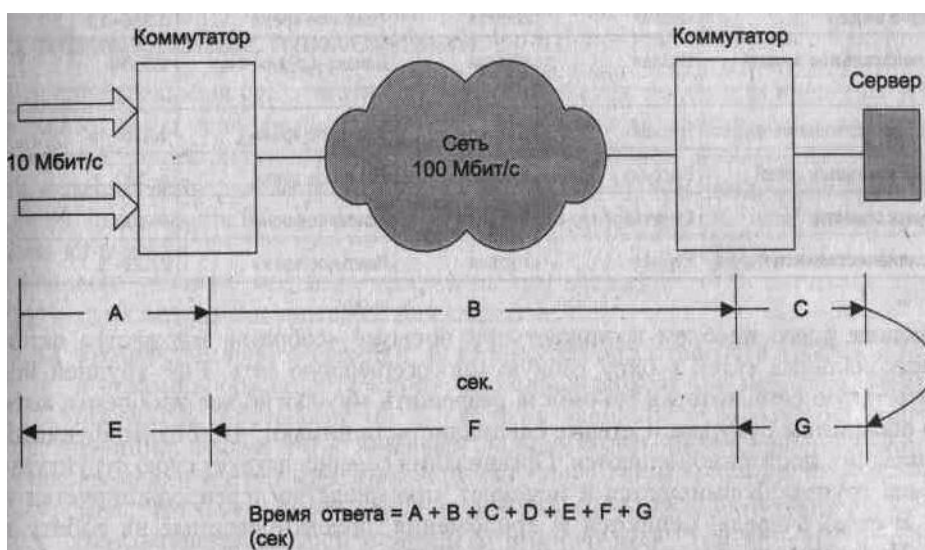


Рис. 3.3. Общее время ответа сети

Понятия емкости и пропускной способности сети связаны между собой, но, по сути, это не одно и то же. Емкость сети — это реальное количество ресурсов, доступных пользователю на определенном пути передачи данных. Пропускная способность сети определяется общим количеством данных, которые могут быть переданы в единицу времени. Емкость сети отличается от пропускной способности сети из-за наличия накладных расходов, которые зависят от способа использования сети.

Таблица 3.1 содержит характеристики трафика для различных приложений.

Нет ни пользователей, ни разработчиков, которые не были бы озабочены оптимальностью создаваемой сетевой инфраструктуры. При этом главный вопрос:

будет ли работа сети удовлетворительной по истечении некоторого времени после ее внедрения?

Таблица 3.1. Характеристики трафика разных приложений

Приложение/ Характеристика	Загруженность трафика	Терпимость к задержкам	Время ответа	Пропускная способность, Мбит/с
Электронная почта	Высокая	Высокая	Регламенти- руется	0.004-0.20
Передача файлов	Высокая	Высокая	Регламенти- руется	0.01-600
CAD/CAM- системы	Высокая	Средняя	Близко к РВ	1-100
Обработка транзакций	Высокая	Низкая	Близко к РВ	0.064-1.544
Связь локальных сетей	Высокая	Высокая	Реальное время	4-100
Доступ к серверу	Средняя	Высокая	Реальное время	4-100
Высококаче- ственное аудио	Низкая	Низкая	Реальное время	0.128-1

Больше всего проблем возникает при попытке «собрать» множество одно-функциональных сетей в одну гибкую многосервисную сеть. Еще трудней получить такую сеть, которая бы смогла разрешить абсолютно все проблемы, хотя бы в обозримом будущем. Сетевые специалисты понимают, что бизнес-функции организации постоянно меняются.

Организация совершенствует свою структуру, рабочие группы формируются и исчезают, производство перепрофилируется и т.д. В свою очередь, меняются и приложения, ориентированные на работу в сети. Пользовательские рабочие станции сейчас предоставляют услуги по обработке сообщений, видеоинформации, телефонии и т. д.

В этой связи, при создании сети с комбинированными функциями нужно гарантировать необходимый уровень сервиса для каждого приложения. В противном случае пользователи будут вынуждены отказаться от многосервисной сети в пользу старой специализированной сети. Как показывает текущее состояние сети Internet, обработка всего трафика на равных правах приводит к серьезным проблемам, особенно при ограниченной пропускной способности. Некоторые приложения требуют быстрой реакции сети. Поэтому возникла необходимость гарантировать время реакции, пропускную способность сети и подобные параметры. Такая технология была разработана и получила название качество обслуживания (Quality of Service, QoS). Качество обслуживания использует распределение по категориям и назначение приоритетов трафикам, что позволит гарантировать трафику с большим приоритетом лучшие условия передачи через сетевую магистраль, вне зависимости от требований к пропускной способности трафиков менее важных приложений. Технология качества обслуживания может применяться для определения стоимости услуг многосервисной сети. Качество обслуживания позволяет связать стоимость сетевых услуг с сетевой производительностью.

Однако возникает вопрос: какую именно технологию качества обслуживания должен выбрать сетевой специалист? Существует несколько вариантов: организация приоритетных очередей в маршрутизаторах, использование протокола RSVP, применение QoS ATM и т. д. Но следует отметить, что всегда можно отказаться от технологии качества обслуживания. Это можно сделать, например, введя «силовые»

методы распределения полосы пропускания и не используя эти методы там, где не нужно. Для выбора конкретной технологии качества обслуживания необходимо провести анализ требований пользователей к качеству обслуживания и рассмотреть возможные альтернативы.

Трафик разных приложений

В последнее время все отчетливее прослеживается тенденция введения в приложения услуг телефонии, групповой работы над документами, обработки сообщений, видео и т. д. Эта тенденция определяет требования к сетевой магистрали, которая, комбинируя ЛВС-, MAN- и WAN-магистрали, должна иметь многосервисную основу и передавать любые типы трафика с требуемым качеством.

Можно условно разделить трафик на три категории, отличающиеся друг от друга требованиями к задержке при передаче:

Трафик реального времени. К этой категории относятся трафик с аудио-и видеоинформацией, не допускающий задержки при передаче. Задержка обычно не превышает 0,1 с, включая время на обработку на конечной станции. Кроме того, задержка должна иметь небольшие колебания во времени (эффект «дрожания» должен быть сведен к нулю). Следует отметить, что при сжатии информации трафик данной категории становится очень чувствительным к ошибкам при передаче. При этом из-за требования малой задержки возникающие ошибки не могут быть исправлены с помощью повторной отправки;

Трафик транзакций. Эта категория требует задержки до 1 с. Увеличение этого предельного значения заставляет пользователей прерывать свою работу и ждать ответа, потому что только после получения ответа они могут продолжить отправлять свои данные. Поэтому большие задержки приводят к уменьшению производительности труда. Кроме того, разброс в значениях задержки приводит к дискомфорту в работе. В некоторых случаях превышение допустимого времени задержки приведет к сбою рабочей сессии и

пользовательским приложениям потребуется начать ее вновь;

Трафик данных. Эта категория трафика может работать практически с любой задержкой, вплоть до нескольких секунд. Особенностью такого трафика является повышенная чувствительность к доступной пропускной способности, но не к задержкам. Увеличение пропускной способности влечет за собой уменьшение времени передачи. Приложения, передающие большие объемы данных, разработаны, в основном, так, что захватывают всю доступную полосу пропускания сети. Редкими исключениями являются приложения потокового видео. Для них важны и пропускная способность и минимизация времени задержки.

Внутри каждой рассмотренной категории графики классифицируются по присвоенным им приоритетам. Трафик, имеющий более высокий приоритет, получает предпочтение при обработке. Примером приоритетного трафика может быть транзакция с заказом.

Введение приоритетов неизбежно при недостаточности ресурсов сети. Приоритеты могут использоваться для выделения групп, прикладных программ и отдельных пользователей в группах.

Передача аудио- и видеoinформации чувствительна к изменению задержки или, иными словами, к дрожанию. Например, превышение допустимого порога дрожания может привести к достаточно ощутимым искажениям изображений, необходимости дублирования видеок кадров и т. д. Передача звука также чувствительна к дрожанию, так как человеку трудно воспринимать неожиданные паузы в речи абонента.

Проведенные исследования показали, что в случае передачи низкокачественной аудиоинформации по сети, максимальная задержка сигнала должна находиться в пределах от 100 до 150 мс. В случае передачи изображений этот параметр не должен превышать 30 мс. Таблица 3.2 определяет диапазон приемлемых задержек при передаче аудиоинформации.

Таблица 3.2. Воздействие задержек на восприятие голосового сигнала

Задержка	Эффект для пользователя
>600 мс	Взаимодействие невозможно
600 мс	Взаимодействие затруднено
250 мс	Искажение речевого потока. Необходима адаптация к каналу связи
100 мс	Задержки практически незаметны
50 мс	Передача без искажений

Кроме того, так как потоки аудио- и видеоинформации следуют через различные устройства, которые обрабатывают трафик с учетом эффекта дрожания на основе разных алгоритмов, может быть быстро потеряна синхронизация между изображением и голосом (как это бывает в плохих фильмах). С эффектом дрожания можно бороться, применяя буферную память на принимающей стороне. Но следует помнить, что объем буфера может достигать значительных размеров, а это приводит как к удорожанию аппаратуры, так и к обратному эффекту — увеличению задержки за счет накладных расходов при обработке информации в большом буфере.

3.3. Описание программы Net Cracker.

Программа Net Cracker Professional предназначена для моделирования компьютерных сетей всех типов, а также имитации процессов в созданных сетях. При имитации процессов в созданных проектах сетей программа позволяет выдавать отчеты по результатам имитации. Методика построения проекта включает следующие шаги:

1. В окно проекта заносится сетевое оборудование, которое будет использоваться для построения сети. Если необходимо, то в рабочие станции и/или сервера добавляются сетевые адаптеры из списка. Возможно конфигурирование рабочих станций и серверов,

которое выполняется при нажатии на них правой кнопкой мыши.

2. В режиме “Link devices” соединяются сетевое оборудование и компьютеры.
3. Для того, чтобы можно было задать трафик на серверы обязательно устанавливается соответствующее общее программное обеспечение (ПО) (в списке оборудования выбирается опция Network and Enterprise Software).

Поддержка по умолчанию общим ПО типов трафика приведена в таблице 2.3.

Таблица 3.3. Поддержка трафика по умолчанию.

Общее ПО	Поддерживаемый трафик
E-mail server	SMTP; POP3
File-server	File client-server
SQL-server	SQL
FTP-server	FTP
Small office database server	Data base client-server; SQL
HTTP – server	HTTP

Если выбранное общее ПО не поддерживает конкретный тип трафика, то настройка осуществляется следующим образом:

- кликнуть правой клавишей по серверу в окне проекта;
- выбрать опцию *Configuration* в контекстном меню;
- выделить в окне конфигурации установленное на сервер общее ПО и нажать клавишу *Plug-in Setup*;
- выбрать вкладку *Traffic*;
- установить необходимые флаги типов трафика;
- нажать клавишу ОК;
- закрыть окно конфигурации.

В этом же окне конфигурации, на вкладке *Server* можно задать параметры ответа сервера на поступающие запросы.

Для задания трафика между компьютерами на панели инструментов надо нажать кнопку “Set Traffic, затем поочередно щелкнуть левой кнопкой мыши станцию-клиента и сервер, с которым клиент будет обмениваться данными. Трафик можно также задать и между клиентами. Направление трафика определяется от первого щелчка ко второму. Изменять свойства трафика можно с помощью пункта меню “Global”=>”Data Flow”, в том числе добавлять и удалять сетевой трафик.

1. При выборе компьютера или сегмента сети необходимо в соответствии с заданием указать типы отображаемой статистики. Для этого следует выбрать в выпадающем меню пункт “Statistics”, а в появившемся окне галочками отметить, в каком виде выводить статистику. Статистику можно выводить в виде диаграммы, числа, графика или голосом. Далее нажать ОК.
2. В случае многоуровневого проекта, когда при построении сети один фрагмент сети верхнего уровня детально показывается на нижнем уровне (например, когда требуется показать связи между зданиями и показать строение сети внутри здания), следует выделить раскрываемый фрагмент, нажать на правую кнопку мыши, и в выпадающем меню выбрать пункт => Expand. После этого можно продолжать рисовать сеть на новом листе.
3. Процесс имитации запускается с помощью кнопки “Start”.

После окончания процесса имитации отчеты выводятся следующим образом: в меню выбирается пункт “Tools” => “Reports” => “Wizard” => “Statistical” => в зависимости от задания. Отчет можно также получить, не используя услуги мастера, а просто выбрав соответствующий пункт в

подменю “Reports”. Полученный отчет можно распечатать или сохранить в виде файла.

Полученный рисунок сети можно вывести на печать, используя меню File=>Print.

Примечания:

- Длины кабелей берутся произвольно, но не должны превышать допустимые стандартом значения.
- Для сетей с топологией FDDI в базе данных нет устройств MSAU. Поэтому для этой топологии в базе следует выбрать “Generic LAN’s”=>FDDI (схематический рисунок FDDI).
- Устройства типа сервера удаленного доступа можно найти в базе устройств Routers and Bridges => Access Server => открыть любого производителя => найти там подходящее устройство. После этого к нему можно подключить либо модемы, либо устройства DSU/CSU.
- Построение многоуровневого (иерархического) проекта необходимо начинать с самого верхнего уровня (корня), раскрывая подуровни через контекстное меню (Expand) выделенного объекта текущего уровня.
- Фоновое изображение карты местности (Map) выбирается при настройке: меню Sites => Site Setup =>Background.
- Когда необходимо создать проект, используя однотипные устройства, можно выбирать их из списка недавно использовавшихся устройств, для чего щелкните на закладке недавно использовавшихся элементов (Recently Used) в панели изображений.

3.4. Пример выполнения задания.

Задание

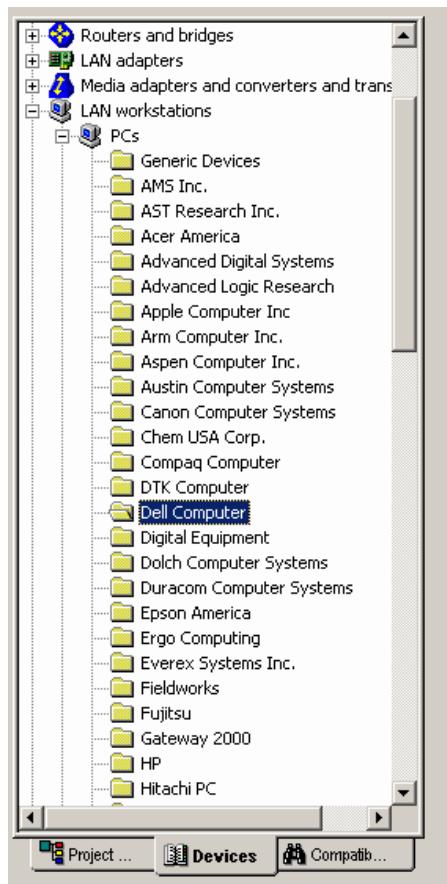
Построить ЛВС следующей топологии: 5 персональных компьютеров (PC) и сервер образуют сегмент 10BASE-T. Другие пять компьютеров объединены в сегмент по технологии 10BASE-5, оба сегмента соединены мостом. Сервер может обслуживать клиентов базы данных, CAD/CAM-приложений и предоставлять FTP доступ к файлам. Рабочие станции сегмента 10BASE-T являются клиентами CAD/CAM приложений, рабочие станции сегмента 10BASE-5 являются клиентами базы данных. Кроме этого, все рабочие станции обращаются на сервер за файлами по FTP, а внутри каждого сегмента взаимодействуют друг с другом по трафику Small office peer-to-peer.

Размер ответа сервера на запрос (Reply Size) рассчитывается по нормальному закону. Мат. ожидание – 1000, дисперсия - 800, размер в байтах. Задержка ответа на запрос (Replay Delay) рассчитывается по экспоненциальному закону, мат. ожидание – 5, время в секундах.

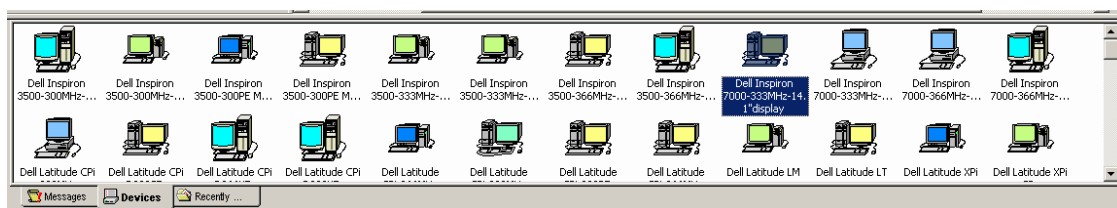
Вывести статистику: для сервера - текущую нагрузку (current workload) и количество полученных пакетов; для сегмента 10BASE-5 - процент использования (average utilization).

Порядок выполнения работы

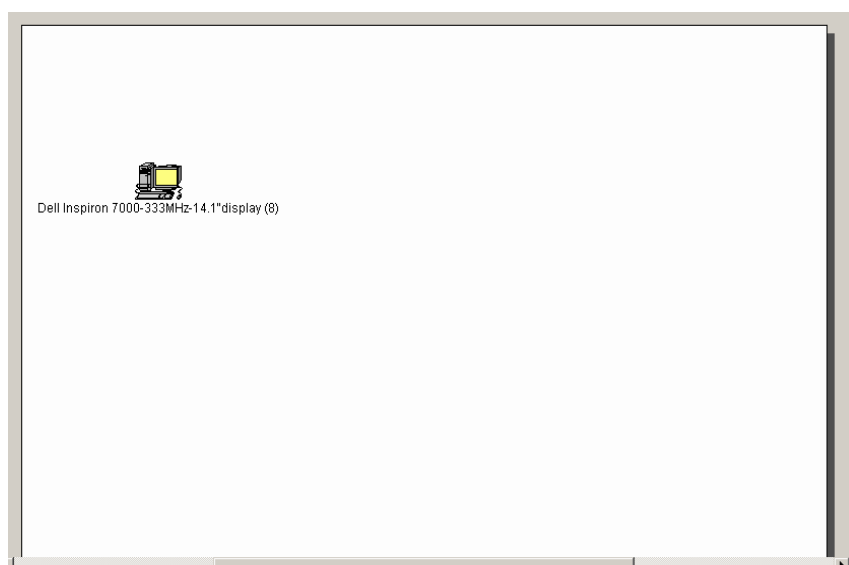
Запустите NetCracker Professional. При построении модели сети можно использовать сетевое оборудование любого, но конкретного производителя, и нельзя использовать общие (Generic Devices) сетевые устройства. Поэтому в качестве рабочих станций будем использовать компьютеры фирмы Dell Computer. В Device Browser'e найдите в разделе LAN workstations/PCs категорию рабочих станций фирмы Dell Computer.



Внизу на панели изображений устройств найдите модель Dell Inspiron 7000-333MHz-14.1"display



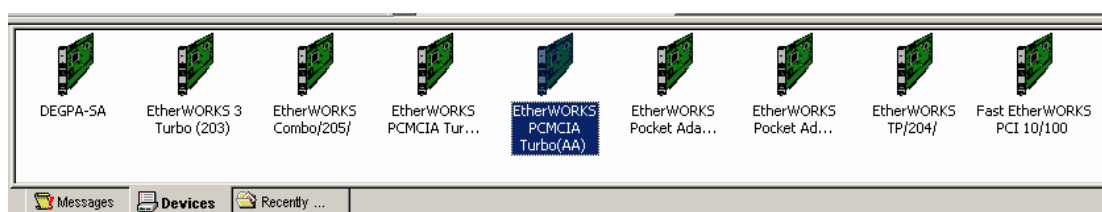
Переместите найденную модель на рабочую область в окно проекта



Найдите в Device Browser аналогичным образом сетевой адаптер EtherWORKS PCMCIA Turbo(AA) (Раздел LAN Adapters/Ethernet/Digital Equipment)



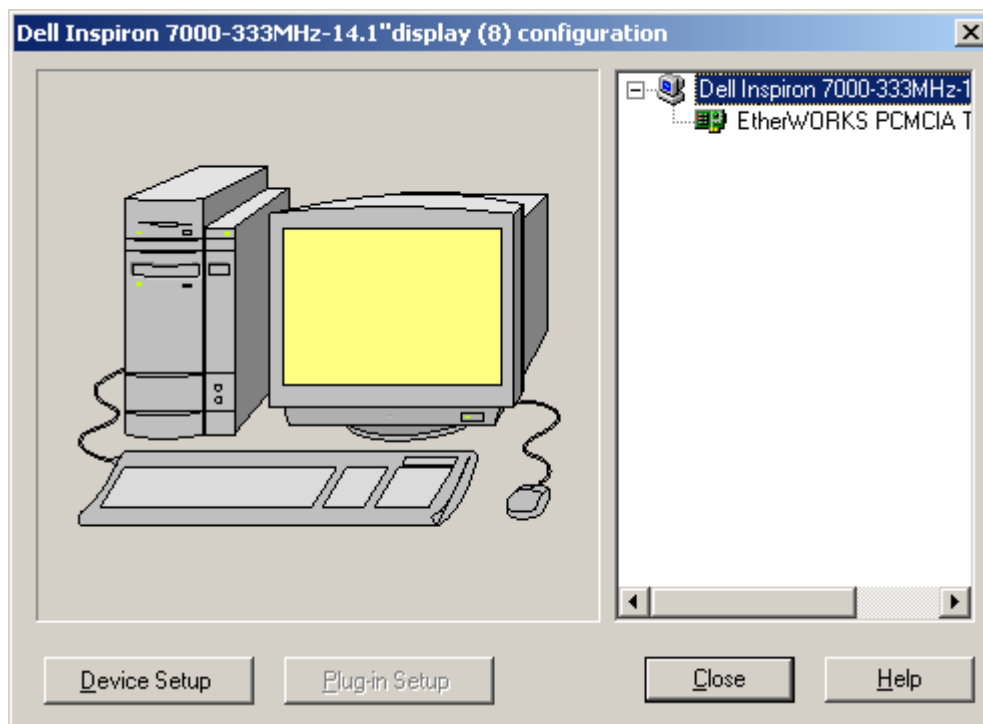
Искомая модель



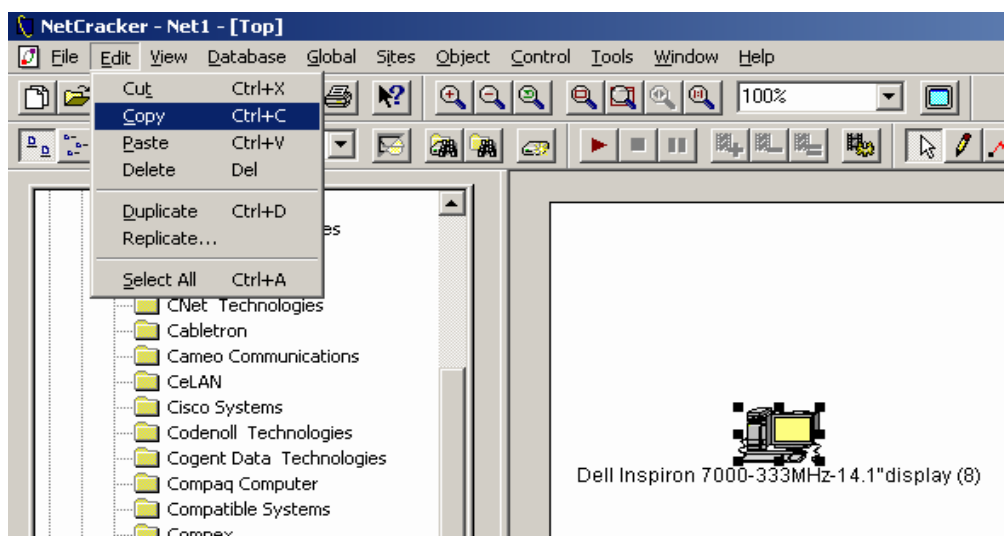
Перетащите найденный сетевой адаптер на имеющийся в рабочей области компьютер.

Теперь у вас рабочая станция оснащена сетевым адаптером.

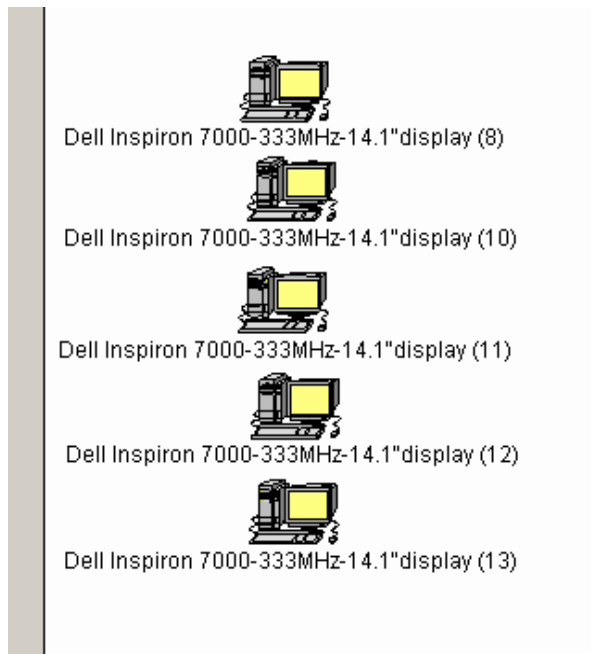
Проверить это вы сможете двойным щелчком левой клавиши мыши по изображению компьютера:



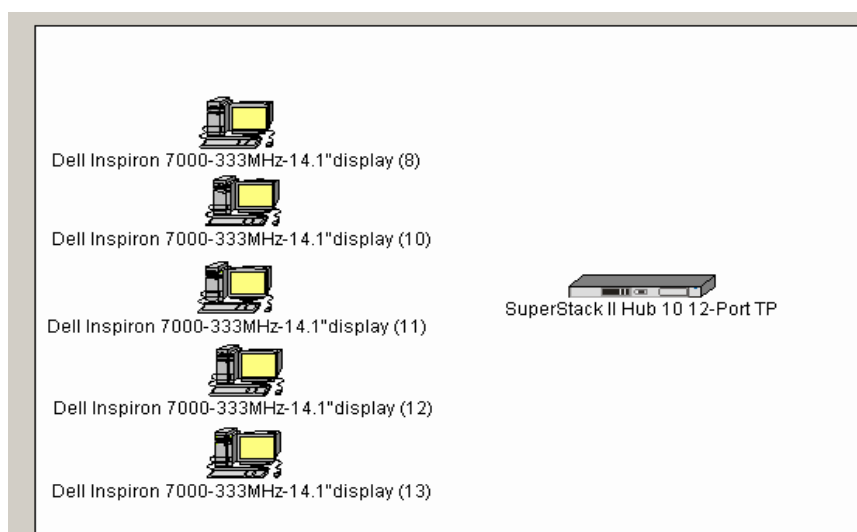
Для построения сегмента Ethernet 10Base-T вам по условию задачи необходимо 5 рабочих станций. Первую вы уже создали. Остальные 4 создадим с помощью копирования. Выделив имеющуюся рабочую станцию, выберите в меню Edit/Сору.



Затем с помощью Edit/Paste создаете на рабочей области 4 подобных компьютера:



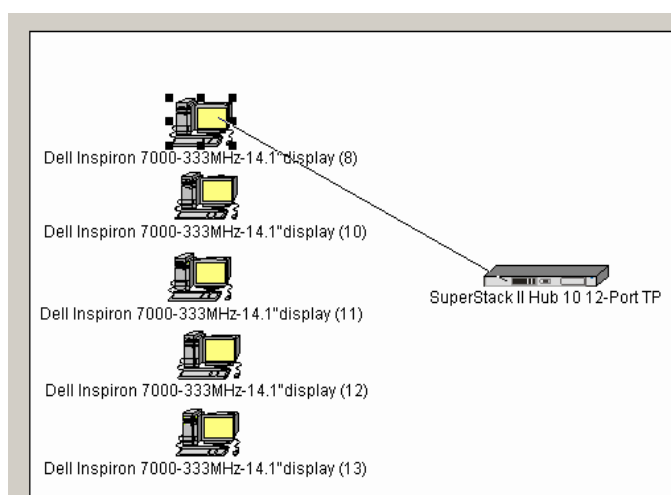
Для объединения рабочих станций в сегмент вам необходим концентратор. Найдите модель SuperStack II Hub 10 12-Port TP в разделе Hubs/Shared Media/Ethernet/3Com Corp. Переместите его на рабочую область:



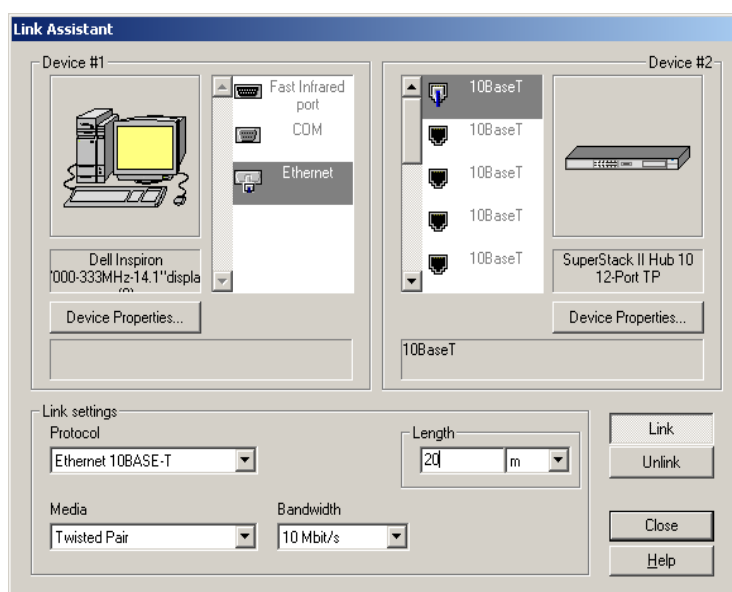
Перейдите в режим физического соединения устройств, щелкнув на панели инструментов по кнопке **Link devices**



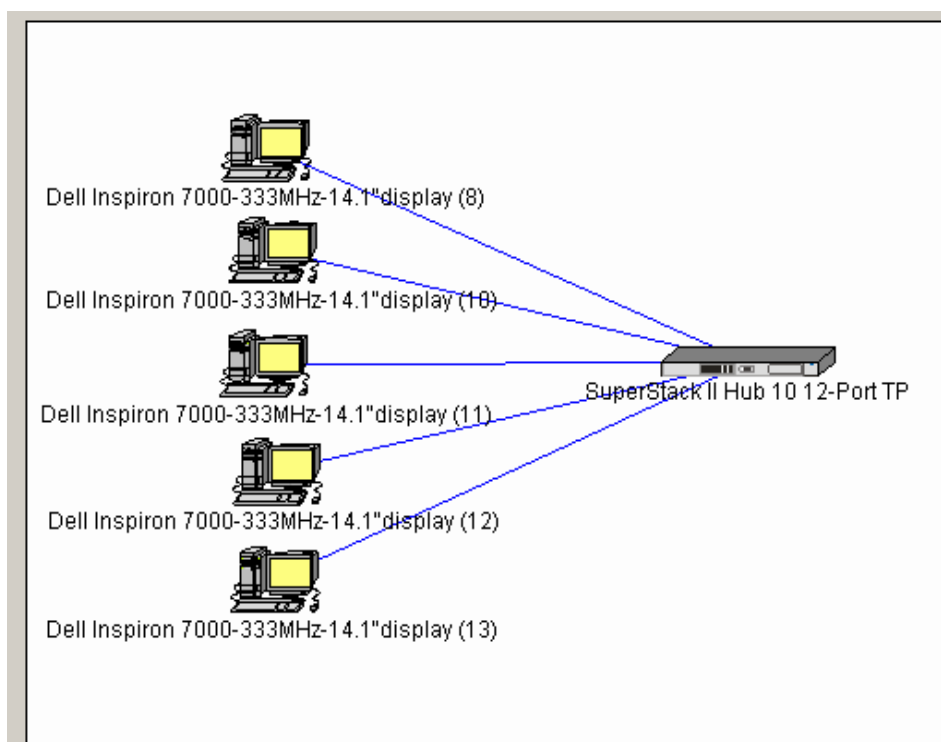
С помощью левой клавиши мыши соедините одну из рабочих станций с концентратором



В раскрывшемся окне Link Assistant нажмите кнопку **Link** и установите длину между устройствами. Остальные параметры оставьте без изменений. Закройте окно клавишей **Close**.



Подобным образом соедините оставшиеся 4 рабочие станции с концентратором:




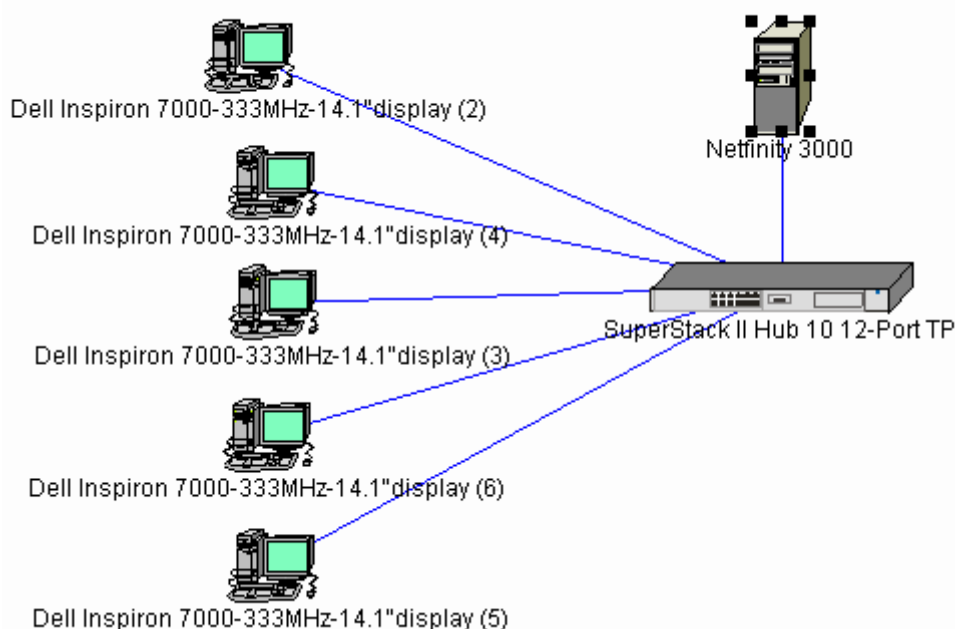
Итак, мы получили сегмент 10BASE-T. После необходимо установить сервер, который устанавливается следующим образом: в окне Device Brouser выбираем иконки в следующей последовательности



и выбираем модель:



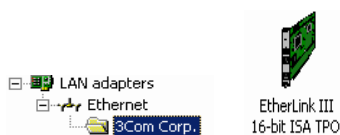
Подключим сервер к концентратору с помощью кнопки **Link Assistant**  в результате чего получаем следующую схему:



Следующим нашим шагом будет объединение других пяти компьютеров в сегмент 10BASE-2. По аналогии с пунктами 2-4 в Device Brouser'e находим в разделе LAN workstations/PCs категорию рабочих станций фирмы Hitachi :

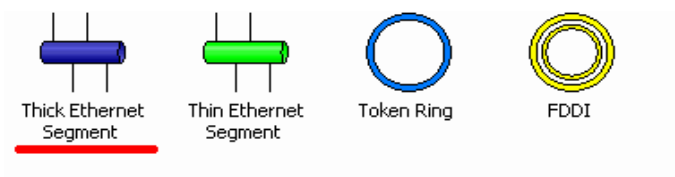


Переносим на рабочую область 5 компьютеров и оснащаем их соответствующими сетевыми адаптерами

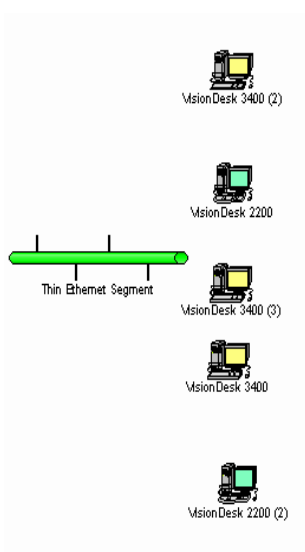


Для объединения компьютеров нам потребуется сегмент Thick Ethernet Segment, который мы находим следующим образом: в окне

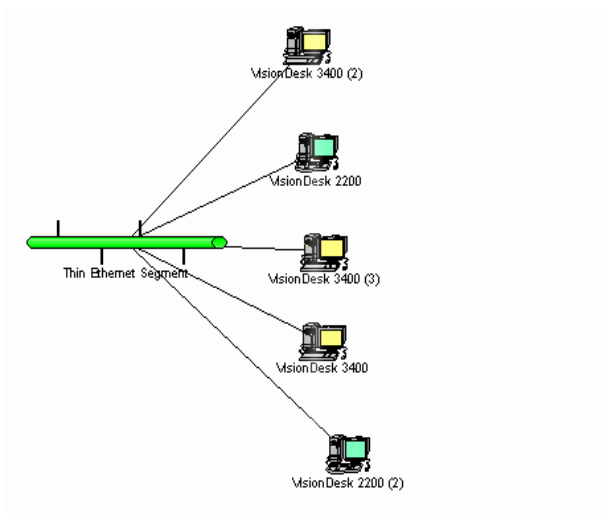
Device Brouser выбираем папку Generic LANs и в нижнем окне выбираем нужный сегмент:



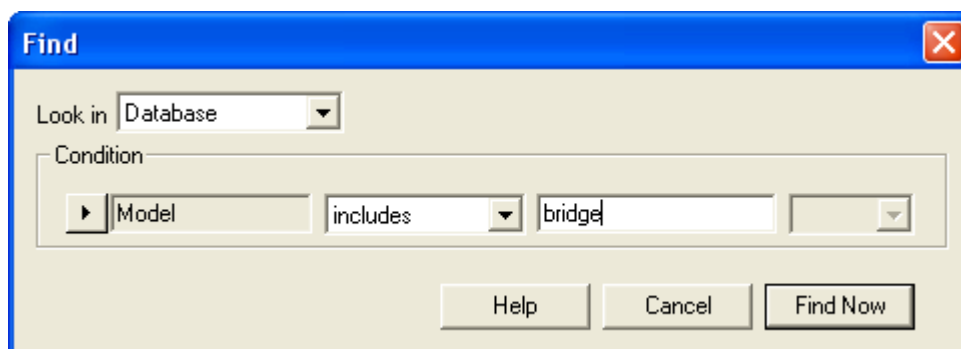
Разместив сегмент на рабочей области, получаем следующую схему:



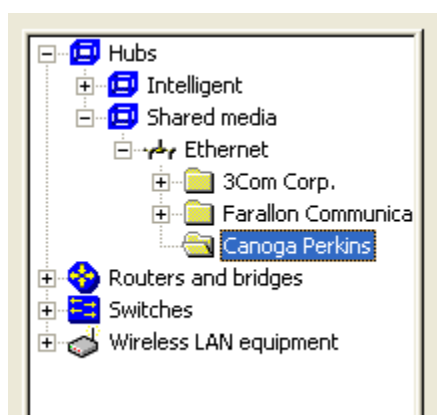
Следующим этапом будет объединение имеющихся рабочих станций в сегмент Thin Ethernet Segment с помощью режима физического соединения устройств (кнопка **Link devices**):



Согласно условию сегменты 10Base-T и 10Base-2 соединены мостом. Найдем мост следующим образом: в меню выберем Database/Find , в открывшемся окне поиска зададим условие поиска и нажмем клавишу Find Now.




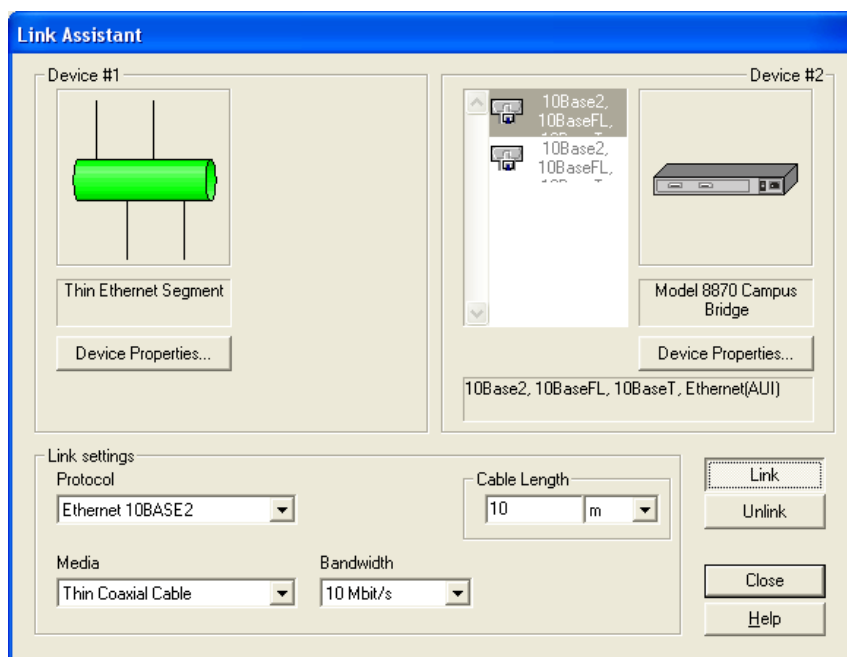
В окне результатов поиска Device Browser/Compatible Devices откроем папку Hubs/Shared media/Ethernet/Canoga Perkins:



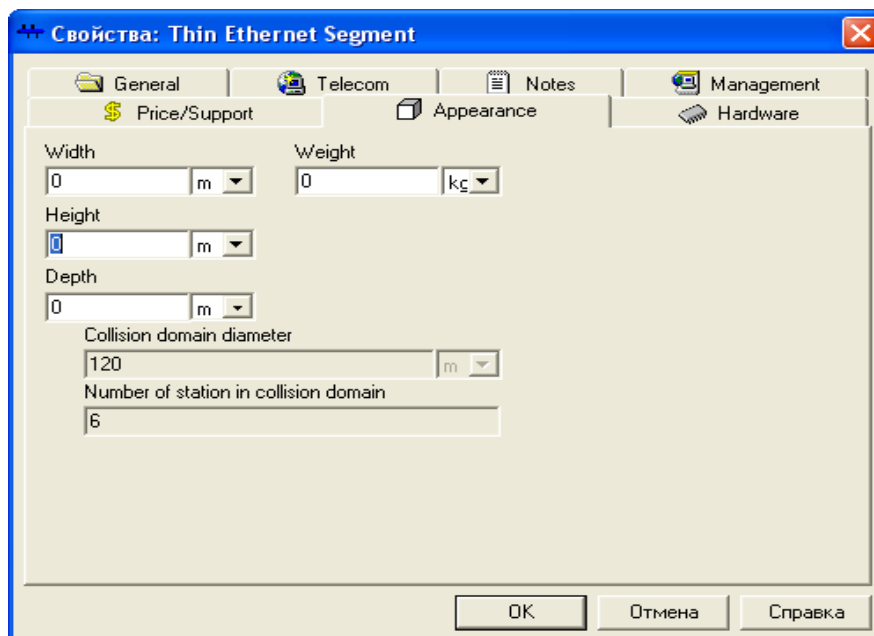
и в нижнем окне выберем мост Model 8870 Campus Bridge.



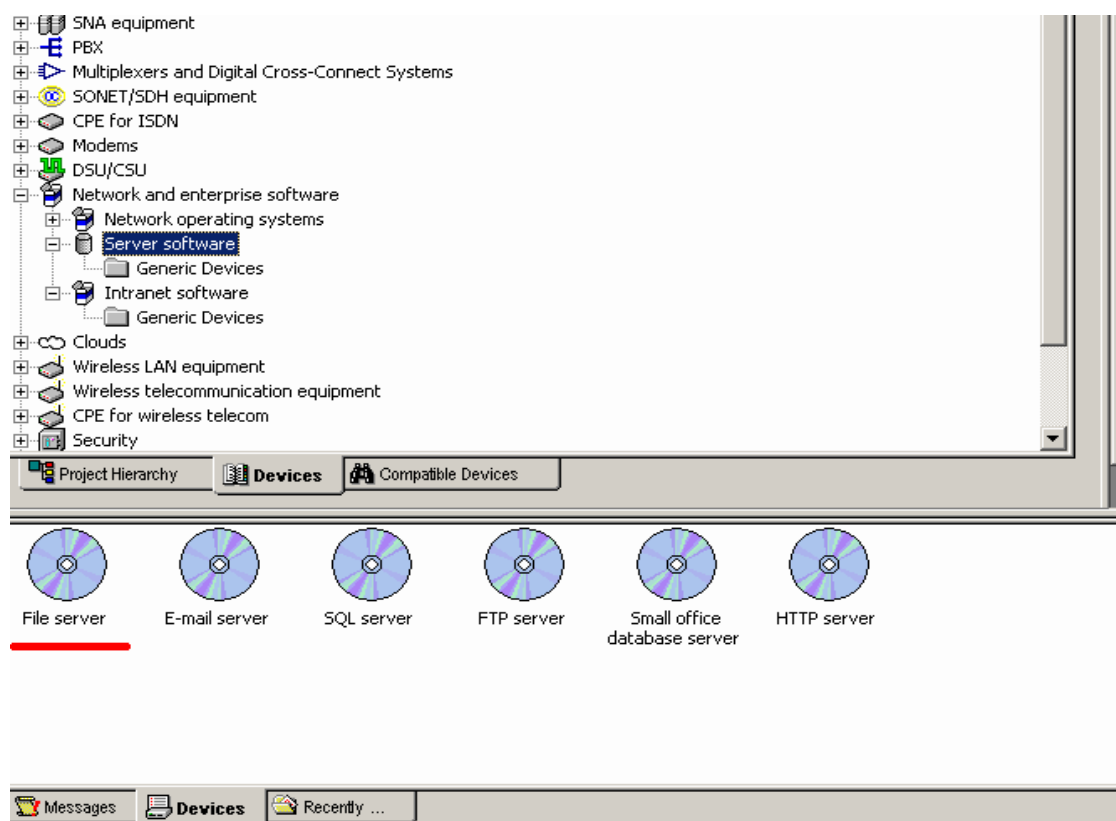
Подключим к мосту сегменты 10Base-T и 10Base-2 с помощью кнопки **Link Assistant** , настроим параметры соединения.



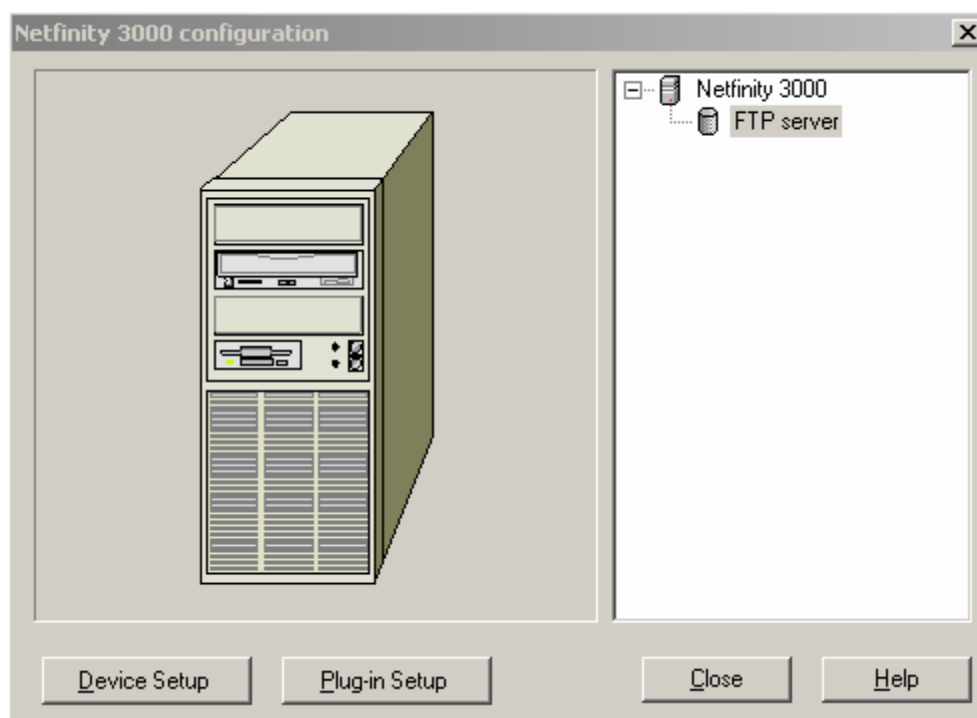
В окне свойств сетевого устройства, например, сегмента Thin Ethernet Segment, которое откроется при выборе в контекстном меню опции Properties, можно выбрать закладку Appearance и узнать диаметр домена коллизий и число устройств в этом домене.





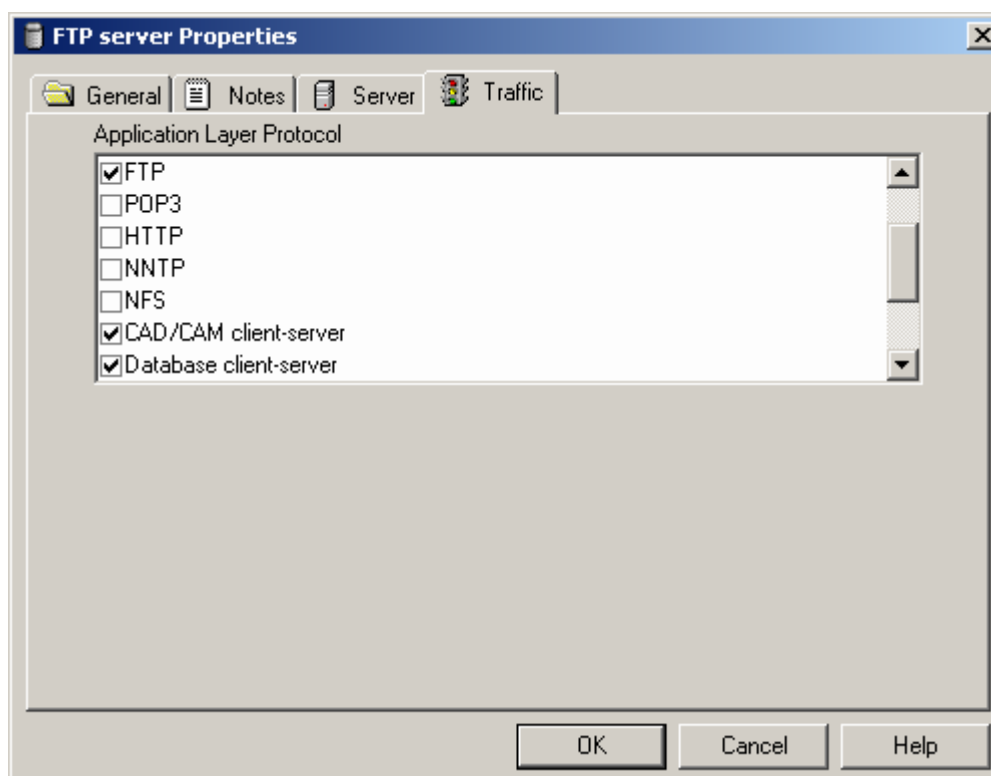
После того, как схема собрана, необходимо установить программное обеспечение на сервере: установим FTP server. Для этого в Device Brouser'e в разделе Network and Enterprise software/Server software выберем FTP server и, выделив объект левой клавишей мыши, переместим его на сервер в рабочей области.



После установки Server software надо настроить приложения и протоколы, которые поддерживает сервер. В данном примере сервер обслуживает клиентов базы данных, CAD/CAM-приложений и предоставляет FTP доступ к файлам. Настройка производится следующим образом: два раза щелкните на сервер левой кнопкой мыши, вызвав окно настройки конфигурации сервера:



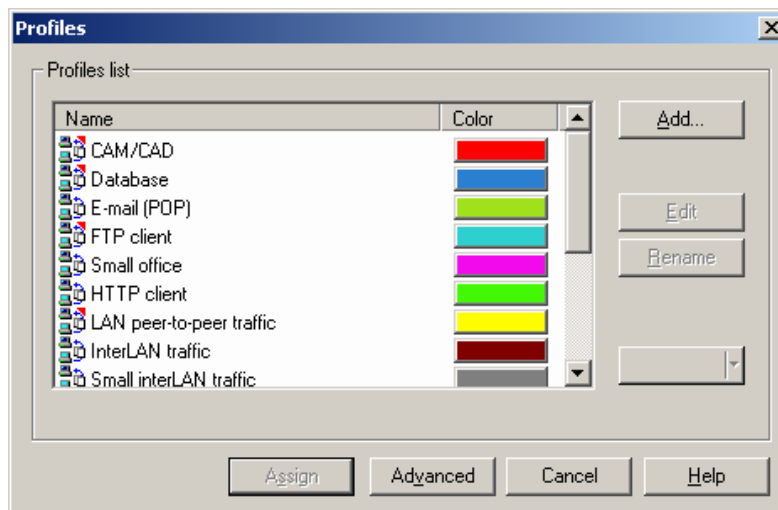
Двойным щелчком мыши по  FTP server вызываем окно настройки программного обеспечения. Перейдите на закладку  Traffic и настройте приложение следующим образом:



После установки программного обеспечения на сервере понадобится настроить трафик, по которому будет происходить обмен данными между компьютерами. Для этого на панели инструментов используйте режим **Set traffic**

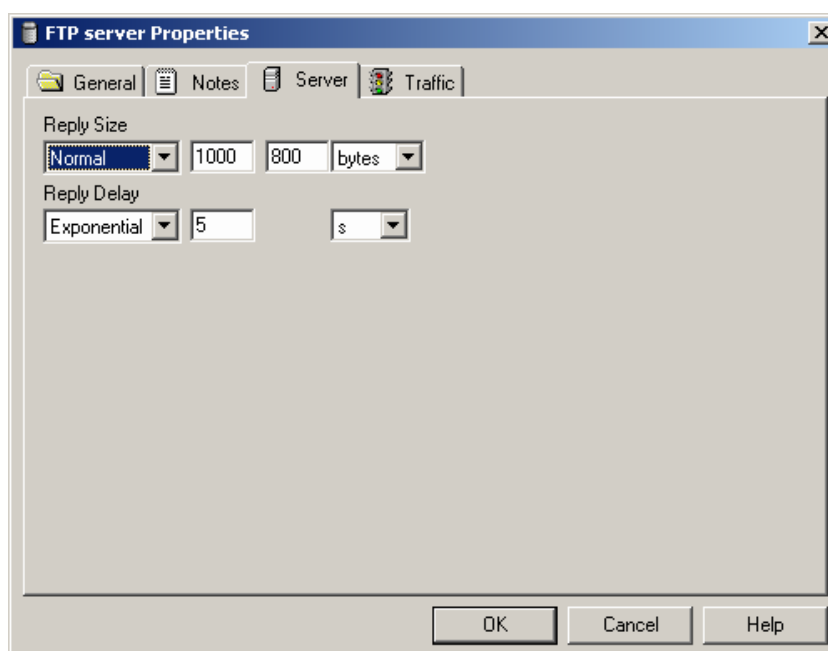


Сначала левой клавишей мыши щелкаете по рабочей станции (источнику трафика), затем – серверу (получателю), и выбираете нужный вид трафика

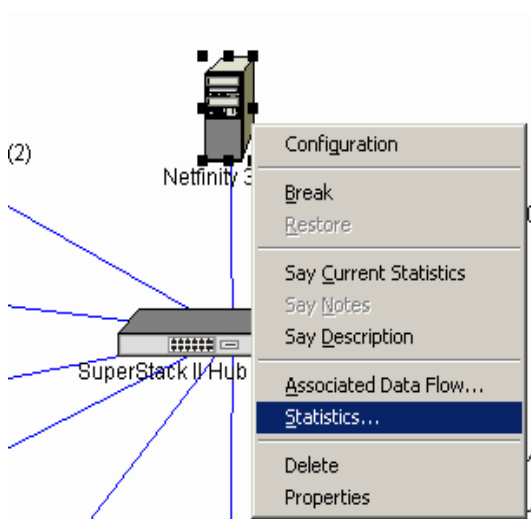


Подобным образом настройте трафик каждой рабочей станции согласно поставленной задаче.

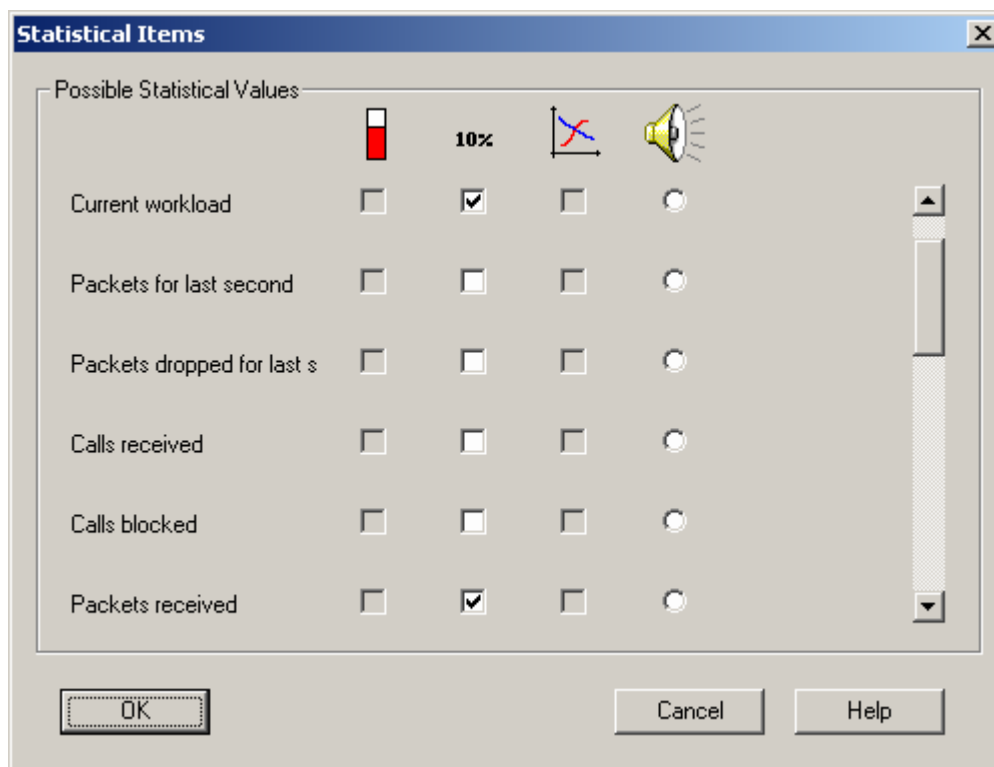
Проверить правильность соединения вы можете следующим образом: в меню программы выберите Global/Data Flow:



Далее выведем статистику: для сервера - текущую нагрузку (current workload) и количество полученных пакетов; для сегмента 10BASE-2 - процент использования (average utilization). Это можно следующим образом: щелкнув правой клавишей мыши по серверу и вызвав контекстное меню, выберите пункт Statistics:



Укажите нужную статистику для вывода:



Аналогично выведем статистику для остальных устройств в соответствии с заданием.

3.5. Задания.

Лабораторная работа №3. Построить ЛВС следующей топологии: Два компьютера PC1 и PC2 через внешние модемы и телефонную сеть общего пользования PSTN имеют FTP-доступ к серверу, расположенному в локальной сети отдельного здания. Эта ЛВС имеет следующую топологию: В рабочих станций, серверы (FS1) и (FS2), а также сервер удаленного доступа (Access Server) образуют сегмент сети 100Base-TX на базе коммутатора. К серверу (FS2) подключен принтер. К серверу удаленного доступа подключен внешний модем, имеющий доступ к PSTN, и через сегмент Thick Ethernet - сервер (FS3) и D рабочих станций. Сервер

(FS1) может обслуживать **F**-клиентов и **G**-клиентов. 2 рабочие станции сегмента Fast Ethernet являются **F**-клиентами. Остальные являются **G**-клиентами. Кроме этого, все рабочие станции обращаются на сервер (FS2) за по протоколу **K**, а локальные станции Thick Ethernet используют **H**-траффик. Принтер обслуживает все локальные рабочие станции. Помимо серверов локальные рабочие станции взаимодействуют друг с другом по трафику Small office peer-to-peer внутри каждого сегмента. Сервер (FS3) является **H** и **J** сервером. Все локальные станции являются **H** и **J** клиентами.

Размер ответа на запрос (Reply Size) для всех серверов рассчитывается по нормальному закону. Мат. ожидание – 1024, дисперсия - 768, размер в байтах. Задержка ответа на запрос (Replay Delay) распределена по экспоненциальному закону, мат. ожидание – 2, время в секундах.

Вывести статистику: для серверов - текущую нагрузку (current workload); для сегмента Ethernet - процент использования (average utilization).

Лабораторная работа №4. Построить ЛВС следующей топологии: Сегмент №1 на концентраторе 10BaseT из **A** рабочих станций и сегмент №2 10Base5 из **B** рабочих станций посредством Ethernet to FDDI switch-1 подключены к кольцу FDDI, к которому подключены через оптоволокно также серверы FS1 и FS2. **C** рабочих станций подключены к тому же кольцу FDDI через Ethernet to FDDI switch -2. Сервер FS1 обслуживает **F**-клиентов и **G**-клиентов, сервер FS2 – **H**, **I** и **J**-клиентов. Станции сегмента №1 являются **F**-клиентами, станции, подключенные через Ethernet to FDDI switch-2 - **G**-клиентами, компьютеры сегмента №2 – **H**-клиентами. Все рабочие станции обращаются за FS2 с **I** и **J** траффиком.

Размер ответа на запрос (Reply Size) для сервера FS1 рассчитывается по нормальному закону: Мат. ожидание – 800, дисперсия - 1000, размер в

байтах. Задержка ответа на запрос (Replay Delay) распределена по экспоненциальному закону, мат. ожидание – 1, время в секундах.

Размер ответа на запрос (Reply Size) для сервера FS2 рассчитывается по равномерному закону: Мат. ожидание – 1024, дисперсия - 2048, размер в байтах. Задержка ответа на запрос (Replay Delay) распределена по экспоненциальному закону, мат. ожидание – 2, время в секундах.

Вывести статистику: для серверов – среднюю нагрузку (average workload); для остального сетевого оборудования - процент использования (average utilization).

3.6. Контрольные вопросы

1. Что такое LAN? Назовите типы сетей и их особенности.
2. Что такое «домен коллизии» и как определить его диаметр?
3. В чем состоит основное различие между концентратором и коммутатором?
4. Сетевые протоколы и их назначение.
5. Модель OSI и её уровни.

3.7. Варианты параметров.

Вариант	A	B	C	D	E	F	G	H	I	J	K
1	2	3	4	9	10	HTTP	FTP	POP3	CAD/CAM	SQL	File Server
2	3	4	5	9	2	File Server	HTTP	FTP	POP3	CAD/CAM	SQL
3	4	5	6	7	8	SQL	File Server	HTTP	FTP	POP3	CAD/CAM
4	5	6	3	8	9	File Server	SQL	CAD/CAM	HTTP	FTP	POP3
5	6	7	8	2	3	HTTP	POP3	CAD/CAM	SQL	File Server	FTP
6	7	4	3	6	2	POP3	CAD/CAM	SQL	File Server	FTP	HTTP
7	8	9	5	2	3	CAD/CAM	SQL	File Server	FTP	HTTP	POP3
8	9	4	2	3	4	SQL	File Server	FTP	HTTP	POP3	CAD/CAM
9	7	3	5	4	6	File Server	FTP	HTTP	POP3	CAD/CAM	SQL
10	2	4	6	8	3	FTP	HTTP	POP3	CAD/CAM	SQL	File Server
11	4	6	8	3	1	POP3	File Server	SQL	CAD/CAM	HTTP	FTP
12	6	8	5	1	3	FTP	POP3	File Server	SQL	CAD/CAM	HTTP

13	8	4	1	3	5	HTTP	FTP	POP3	CAD/CAM	SQL	File Server
14	4	6	3	5	7	File Server	HTTP	FTP	POP3	CAD/CAM	SQL
15	1	3	5	7	2	SQL	File Server	HTTP	FTP	POP3	CAD/CAM
16	3	5	7	2	4	File Server	POP3	CAD/CAM	HTTP	FTP	SQL
17	5	7	2	4	6	HTTP	POP3	CAD/CAM	SQL	File Server	FTP
18	7	3	4	6	8	POP3	CAD/CAM	SQL	File Server	FTP	HTTP
19	2	4	6	8	4	CAD/CAM	SQL	File Server	FTP	HTTP	POP3
20	4	6	8	5	2	SQL	File Server	FTP	HTTP	POP3	CAD/CAM
21	2	5	8	3	6	File Server	FTP	HTTP	POP3	CAD/CAM	SQL
22	5	8	3	6	9	FTP	HTTP	POP3	CAD/CAM	SQL	File Server
23	8	3	6	9	4	POP3	File Server	SQL	CAD/CAM	HTTP	FTP
24	3	6	9	4	7	FTP	SQL	File Server	POP3	CAD/CAM	HTTP
25	6	9	4	7	3	SQL	File Server	POP3	CAD/CAM	HTTP	FTP

Лабораторные работы №5, №6 и №7.

«Проектирование и анализ локальных вычислительных сетей в пакете Cisco Packet Tracer. Адресация. Статическая и динамическая маршрутизация».

4.1. Цель лабораторных работ.

Закрепление теоретических знаний в области конструирования и исследования характеристик локальных вычислительных сетей. Изучение программы Cisco Packet Tracer 5.3, приобретение практических навыков проектирования и моделирования работы сети, а также оценки принятых проектных решений.

4.2. Теоретическая часть.

IP-адресация версии 4

На сетевом уровне (или IP) мы должны уникально идентифицировать каждое устройство в Интернете, чтобы обеспечить глобальную связь между всеми устройствами. Эта адресация напоминает нумерацию в телефонной сети, где каждый абонент имеет уникальный номер телефона, содержащий международный код (код страны), междугородний код города и т. д., который идентифицирует его местоположение.

Установление соединения между двумя и более узлами происходит на основе обработки адресной информации, которая по мере необходимости обрабатывается устройствами 3-го уровня в маршрутизаторах. К адресу предъявляются следующие требования:

- адрес должен быть универсальным;
- адрес должен иметь иерархическую структуру, удобную для обработки соответствующими узлами;
- адрес должен быть удобен для пользователя.

Идентификатор, используемый на уровне IP набора протокола TCP/IP, чтобы идентифицировать каждое устройство, подключенное к Интернету, назван адресом Интернета, или адресом IP. Адрес IP — двоичный адрес на 32 бита, который уникально и универсально определяет подключение хоста или маршрутизатора к Интернету.

Адреса IP уникальны. Они уникальны в том смысле, что каждый адрес определяет одно и только одно подключение к Интернету. Два устройства в Интернете никогда не могут иметь одного того же адреса. Если устройство имеет два подключения к Интернету, через две сети, оно имеет два адреса IP.

Адреса IP универсальны потому, что система адресации должна быть принята любым хостом, который хочет быть связанным с Интернетом.

Адресное пространство.

Протокол, подобный IP, то есть определяющий адреса, имеет адресное пространство. Адресное пространство — общее количество адресов, применяемых в соответствии с протоколом. Если протокол использует N бит, чтобы определить адрес, адресное пространство — 2^N , потому что каждый бит может иметь два различных значения (0 и 1), а N бит могут иметь 2^N значений.

IPv4 использует адреса на 32 бита, то есть адресное пространство — 2^{32} или 4,294,967,296 (больше чем четыре миллиарда). Это означало бы, что, теоретически, если не было бы никаких ограничений, к Интернету

могли бы быть подключены более чем 4 миллиарда устройств. Мы вскоре увидим, что фактически номеров намного меньше.

Маска

Уже давно наблюдается дефицит IP-адресов, который обусловлен не только ростом числа пользователей, но и необходимостью выделения IP-адресов на каждый порт маршрутизатора. Имеется несколько подходов смягчения этой проблемы, в том числе за счет использования масок.

Традиционно номер сети и узла определяется в зависимости от класса адреса. Однако наличие только четырех классов адресов часто бывает неудобно. Например, администратор получил от поставщика услуг номер сети 135.38.0.0 (адрес класса В, двоичный код сети – 10000111 00100110 00000000 00000000). В такой сети потенциально можно иметь 65 534 узла, но такое количество узлов администратору не нужно, ему достаточно иметь 32 000. Проблема решается с помощью масок. Количество "единиц" в маске показывает число старших разрядов, которые определяют номер сети. Для нашего случая следует выбрать маску со значением 255.255.192.0 (двоичный код 11111111 11111111 11000000 00000000). В результате наложения маски на сетевой адрес получается четыре подсети: 135.38.0.0; 135.38.64.0; 135.38.128.0; 135.38.192.0.

Протокол маршрутизации RIP

RIP был повсеместно принят производителями персональных компьютеров (PC) для использования в их изделиях передачи данных по сети. Например, протокол маршрутизации AppleTalk (Протокол поддержания таблицы маршрутизации - RTMP) является модернизированной версией RIP. RIP также явился базисом для

протоколов Novell, 3Com, Ungermann-Bass и Banyan. RIP компаний Novell и 3Com в основном представляет собой стандартный RIP компании Xerox. Ungermann-Bass и Banyan внесли незначительные изменения в RIP для удовлетворения своих нужд.

Формат таблицы маршрутизации

Каждая запись данных в таблице маршрутизации RIP обеспечивает разнообразную информацию, включая конечный пункт назначения, следующую пересылку на пути к этому пункту назначения и показатель (metric). Показатель обозначает расстояние до пункта назначения, выраженное числом пересылок до него. В таблице маршрутизации может находиться также и другая информация, в том числе различные таймеры, связанные с данным маршрутом. Типичная таблица маршрутизации RIP показана на таблице 4.1.

Таблица 4.1.

Destination	Next hop	Distance	Timers	Flags
Network A	Router 1	3	t1, t2, t3	x, y
Network B	Router 2	5	t1, t2, t3	x, y
Network C	Router 1	2	t1, t2, t3	x, y

RIP поддерживает только самые лучшие маршруты к пункту назначения. Если новая информация обеспечивает лучший маршрут, то эта информация заменяет старую маршрутную информацию. Изменения в топологии сети могут вызывать изменения в маршрутах, приводя к тому, например, что какой-нибудь новый маршрут становится лучшим маршрутом до конкретного пункта назначения. Когда имеют место изменения в топологии сети, то эти изменения отражаются в сообщениях о корректировке маршрутизации. Например, когда какой-нибудь роутер

обнаруживает отказ одного из каналов или другого роутера, он повторно вычисляет свои маршруты и отправляет сообщения о корректировке маршрутизации. Каждый роутер, принимающий сообщение об обновлении маршрутизации, в котором содержится изменение, корректирует свои таблицы и распространяет это изменение.

Как и другие протоколы маршрутизации, RIP использует определенные таймеры для регулирования своей работы. Таймер корректировки маршрутизации RIP (routing update timer) обычно устанавливается на 30 сек., что гарантирует отправку каждым роутером полной копии своей маршрутной таблицы всем своим соседям каждые 30 секунд. Таймер недействующих маршрутов (route invalid timer) определяет, сколько должно пройти времени без получения сообщений о каком-нибудь конкретном маршруте, прежде чем он будет признан недействительным. Если какой-нибудь маршрут признан недействительным, то соседи уведомляются об этом факте. Такое уведомление должно иметь место до истечения времени таймера отключения маршрута (route flush timer). Когда заданное время таймера отключения маршрута истекает, этот маршрут удаляется из таблицы маршрутизации. Типичные исходные значения для этих таймеров - 90 секунд для таймера недействующего маршрута и 270 секунд для таймера отключения маршрута.

Доменные имена

Чтобы понять, как действует DNS, полезно ознакомиться с окружением, в котором выполняет свою работу эта служба. Начнем с иерархии DNS. Корневой домен известен просто как ".". Верхний домен находится на один уровень ниже, и на этом уровне находится целый ряд доменов DNS. Вы знаете все эти суффиксы: .COM (коммерческие), .GOV (правительственные), .EDU (образование), .INT (международные), .ORG

(организация), .NET (Net-провайдеры, ISP [Провайдеры услуг Интернет] и т.д.) .MIL (военные) и другие.

На следующем уровне обычно находится домен, поддерживаемый частной фирмой; для примера мы будем рассматривать корпоративный домен .COM, но для этого можно было бы использовать любой из только что перечисленных доменов. Вы можете рассматривать домены DNS аналогично структуре папок на жестком диске в смысле разбиения пространства имен DNS. Домен "." аналогичен корневой папке; далее идет верхний уровень (.COM), и на следующем уровне находится "папка", представляющая корпоративный объект некоторого рода (частный или общественный). Имя верхнего уровня (.COM) и корпоративная "папка", например, Microsoft, совместно образуют доменное имя. Пространство имен Microsoft может разбиваться на меньшие домены ("подпапки") Active Directory/DNS, представляющие в компании Microsoft различные подразделения или службы, например, `accounting.microsoft.com` (бухгалтерский учет). Такой домен обычно недоступен из Internet в отличие от `microsoft.com`; однако имеются исключения, например, `http://support.microsoft.com/default.aspx?scid=fh` представляет службы поддержки Microsoft. Здесь находится база знаний вместе с различными рекомендациями от службы поддержки Microsoft. Каждая субзона отвечает за правильность собственной работы.

Записи DNS

В DNS имеется много типов записей, но мы рассмотрим только наиболее употребительные типы записей.

- А-запись. Указывает адрес хоста. Она отображает хост-имя на адрес и может выглядеть следующим образом:

`Myhost.mycompany.com IN A 192.168.0.1`

- AAAA-запись. Указывает адрес хоста в среде IPv6. Выглядит

следующим образом:

IN AAAA 1234:1:2:3:4:567:89cd

- NS-запись. Эта запись идентифицирует сервер имен для заданного домена DNS. В NS-записях указываются первичные и вторичные серверы для пространства имен плюс дочерние зоны, происходящие из него.
- MX-запись. Эта запись для почтового обмена. Вы можете иметь несколько записей, которые указывают несколько ближайших почтовых серверов, и можете располагать их в нужном вам порядке.

4.3. Введение в пакет Cisco Packet Tracer 5.3

Данный программный продукт разработан компанией Cisco и рекомендован использоваться при изучении телекоммуникационных сетей и сетевого оборудования.

Packet Tracer 5.3 включает следующие особенности:

- моделирование логической топологии: рабочее пространство для того, чтобы создать сети любого размера на CCNA-уровне сложности;
- моделирование в режиме реального времени;
- режим симуляции;
- моделирование физической топологии: более понятное взаимодействие с физическими устройствами, используя такие понятия как город, здание, стойка и т.д.;
- улучшенный GUI, необходимый для более качественного понимания организации сети, принципов работы устройства;
- многоязыковая поддержка: возможность перевода данного программного продукта практически на любой язык, необходимый

пользователю;

- усовершенствованное изображение сетевого оборудования со способностью добавлять / удалять различные компоненты;
- наличие Activity Wizard позволяет студентам и преподавателям создавать шаблоны сетей и использовать их в дальнейшем.

С помощью данного программного продукта преподаватели и студенты могут придумывать, строить, конфигурировать сети и производить в них поиск неисправностей. Packet Tracer дает возможность более подробно представлять новейшие технологии, тем самым делая учебный процесс чрезвычайно полезным с точки зрения усвоения полученного материала.

Данный симулятор позволяет студентам проектировать свои собственные сети, создавая и отправляя различные пакеты данных, сохранять и комментировать свою работу. Студенты могут изучать и использовать такие сетевые устройства, как коммутаторы второго и третьего уровней, рабочие станции, определять типы связей между ними и соединять их (см. рис. 4.1).

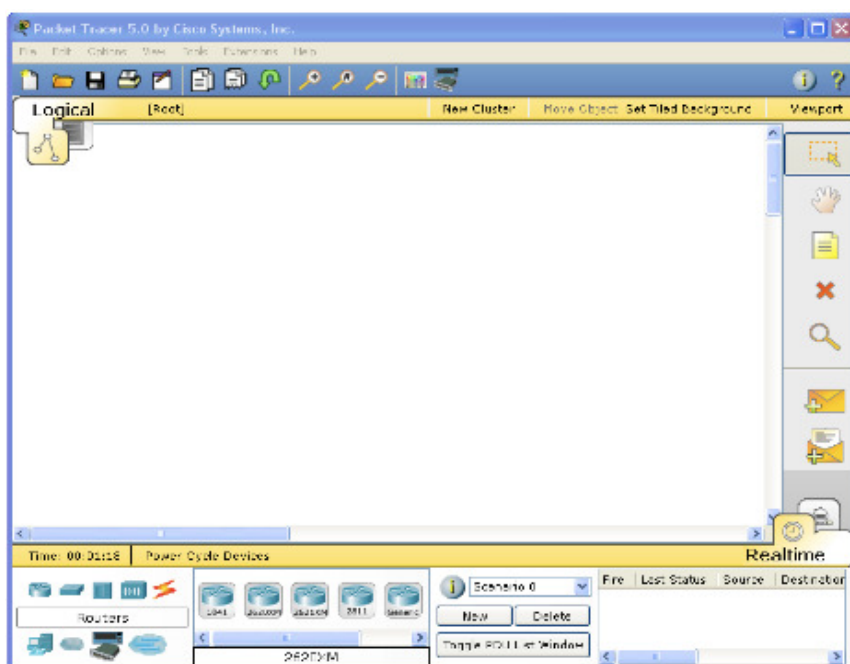


Рис. 4.1 Cisco Packet Tracer

Отличительной особенностью данного симулятора является наличие в нем «Режима симуляции» (рис. 4.2). В данном режиме все пакеты, пересылаемые внутри сети, отображаются графически. Эта возможность позволяет студентам наглядно продемонстрировать, по какому интерфейсу в данный момент перемещается пакет, какой протокол используется и т. д.

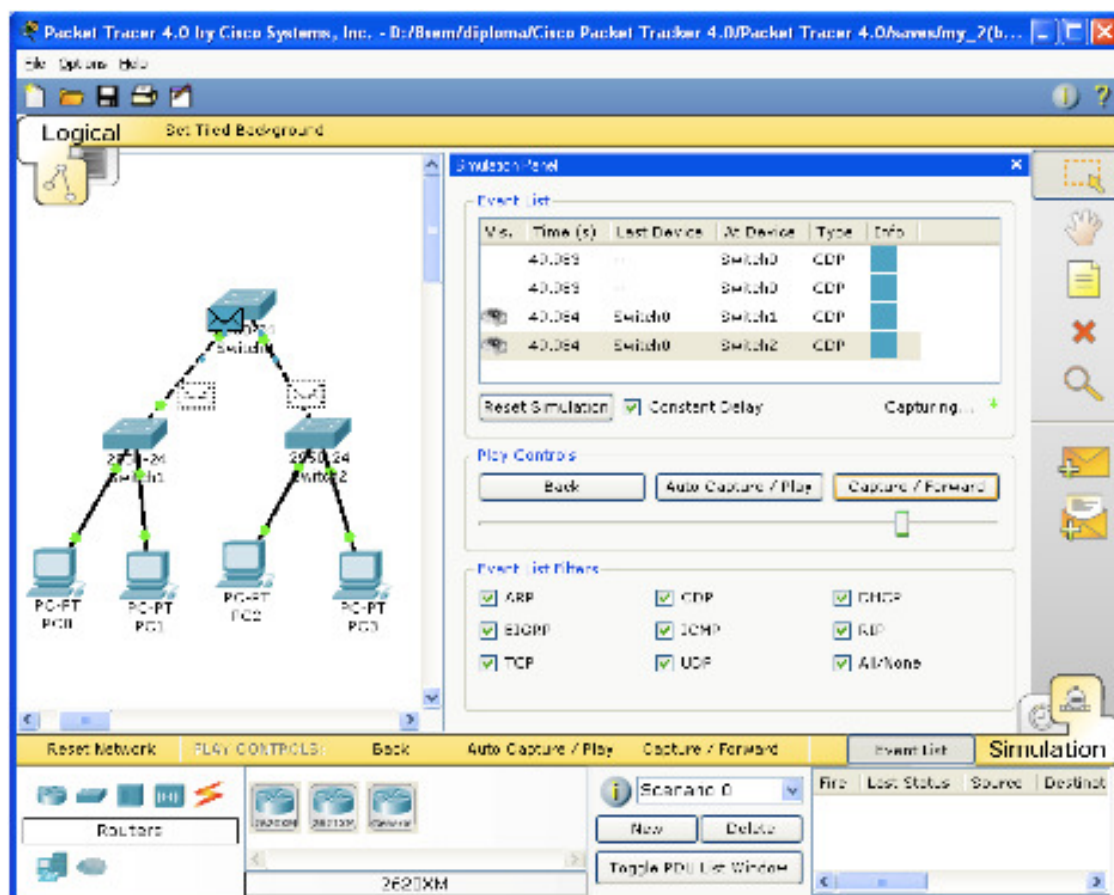


Рис. 4.2. Режим «Симуляции» в Cisco Packet Tracer

Однако, это не все преимущества Packet Tracer: в «Режиме симуляции» студент может не только отслеживать используемые протоколы, но и видеть, на каком из семи уровней модели OSI данный протокол задействован (см. рис. 4.3).

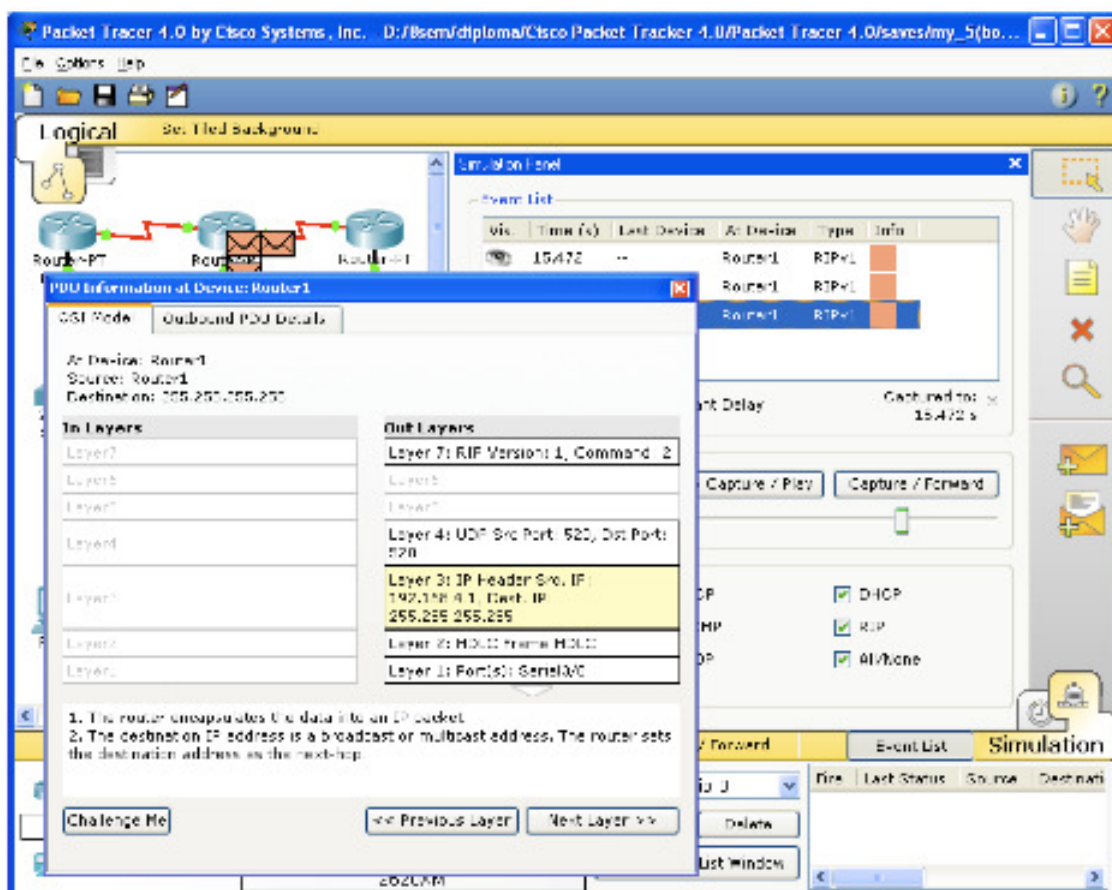


Рис. 4.3. Анализ семиуровневой модели OSI в Cisco Packet Tracer

Такая, кажущаяся на первый взгляд, простота и наглядность, делает практические занятия чрезвычайно полезными, совмещая в них как получение, так и закрепление полученного материала.

Packet Tracer способен моделировать большое количество устройств различного назначения, а так же немало различных типов связей, что позволяет проектировать сети любого размера на высоком уровне сложности:

Моделируемые устройства:

- коммутаторы третьего уровня
- коммутаторы второго уровня
- сетевые концентраторы
- конечные устройства (рабочие станции, ноутбуки, сервера, принтеры)

- беспроводные устройства
- глобальная сеть WAN.

Типы связей:

1. консоль;
2. медный кабель без перекрещивания (прямой кабель);
3. медный кабель с перекрещиванием (кросс-кабель);
4. волоконно-оптический кабель;
5. телефонная линия;
6. Serial DCE;
7. Serial DTE.

Каждое устройство в программном продукте Cisco Packet Tracer может быть сконфигурировано через окно свойств, которое вызывается по двойному клику на устройстве. Первая вкладка отвечает за физические параметры устройства. В маршрутизаторы и коммутаторы можно добавлять новые модули, в рабочие станции и серверы — вставлять сетевые адаптеры. Для того чтобы переконфигурировать устройство, необходимо предварительно его выключить, нажав на кнопку отключения питания на физическом изображении устройства (рис. 4.4).

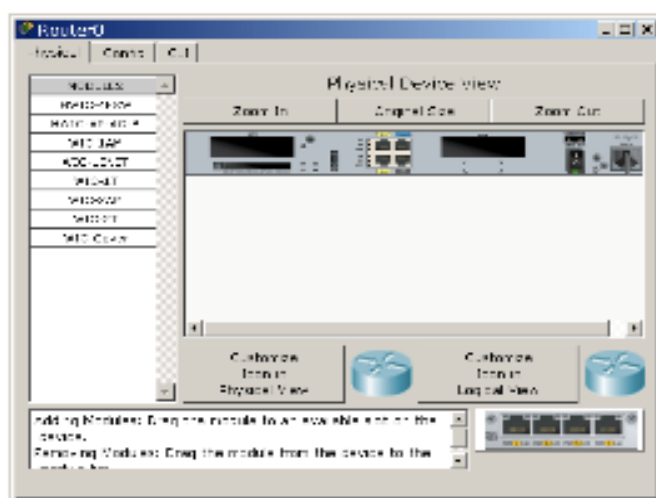


Рисунок 4.4. Физический вид устройства.

На вкладке Config можно задавать основные параметры сетевых интерфейсов (IP-адреса, маску подсети, параметры беспроводной сети и пр.) В сетевых устройствах также можно конфигурировать маршрутизацию – статическую и по протоколу RIP, у серверов – конфигурировать службы. (рис. 4.5.)

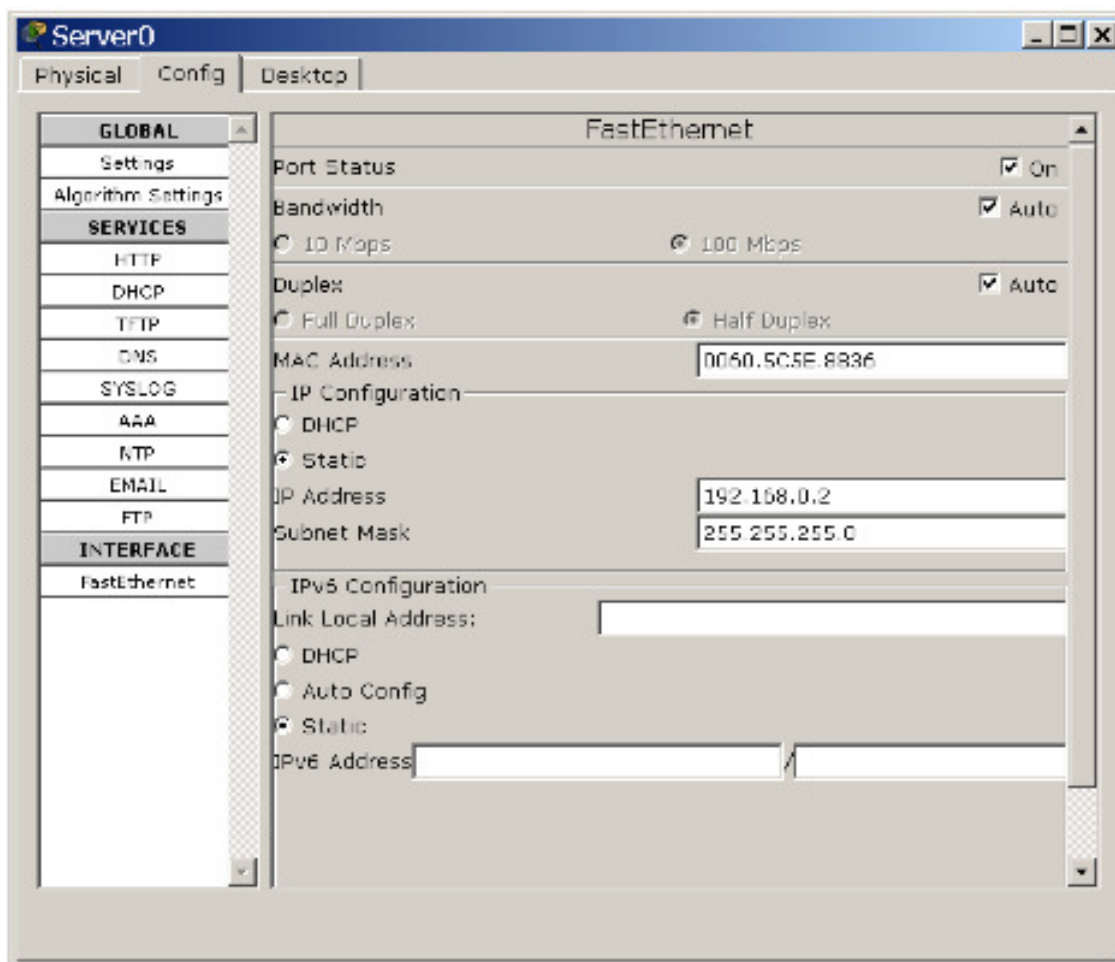


Рисунок 4.5. Конфигурация сервера.

Третья вкладка сетевых устройств обеспечивает доступ к командной строке операционной системы IOS, с которой мы познакомимся подробнее на следующих лабораторных работах. Третья вкладка рабочих станций и серверов содержит интерфейсы доступа к различным сетевым параметрам, а также несколько клиентских приложений. (рис. 4.6.)

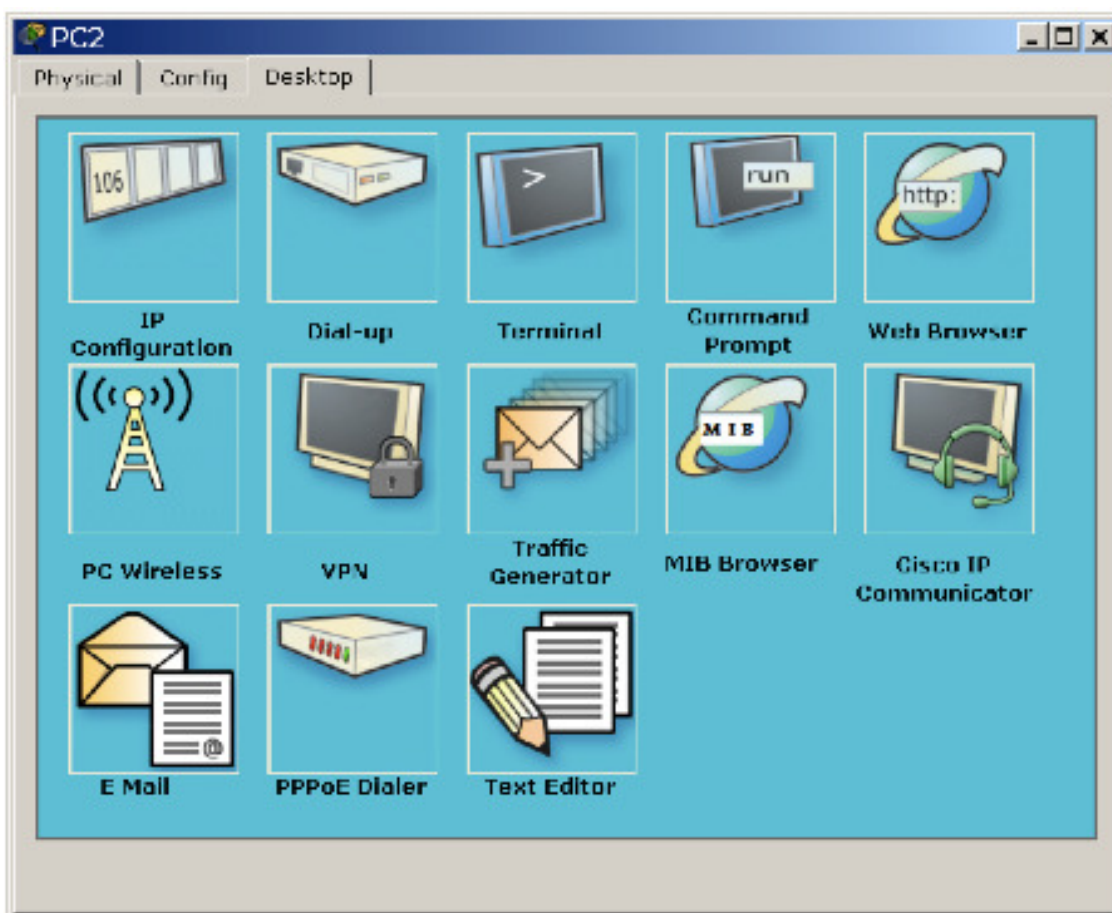


Рисунок 4.6. Вкладка Desktop рабочей станции.

4.4. Командный режим операционной системы IOS

Маршрутизатор конфигурируется в командной строке операционной системы Cisco IOS. Подсоединение к маршрутизатору осуществляется через Telnet на IP-адрес любого из его интерфейсов или с помощью любой терминальной программы через последовательный порт компьютера, связанный с консольным портом маршрутизатора. Последний способ предпочтительнее, потому что процесс конфигурирования маршрутизатора может изменять параметры IP-интерфейсов, что приведет к потере соединения, установленного через Telnet. Кроме того, по соображениям безопасности доступ к маршрутизатору через Telnet следует запретить.

При работе в командной строке Cisco IOS существует несколько контекстов (режимов ввода команд).

Контекст пользователя открывается при подключении к маршрутизатору; обычно при подключении через сеть требуется пароль, а при подключении через консольный порт пароль не нужен. В этот же контекст командная строка автоматически переходит при продолжительном отсутствии ввода в контексте администратора. В контексте пользователя доступны только простые команды (некоторые базовые операции для мониторинга), не влияющие на конфигурацию маршрутизатора. Вид приглашения командной строки:

```
router>
```

Вместо слова `router` выводится имя маршрутизатора, если оно установлено.

Контекст администратора (контекст "exec") открывается командой **enable**, поданной в контексте пользователя; при этом обычно требуется пароль администратора. В контексте администратора доступны команды, позволяющие получить полную информацию о конфигурации маршрутизатора и его состоянии, команды перехода в режим конфигурирования, команды сохранения и загрузки конфигурации. Вид приглашения командной строки:

```
router#
```

Обратный переход в контекст пользователя производится по команде **disable** или по истечении установленного времени неактивности. Завершение сеанса работы - команда **exit**.

Глобальный контекст конфигурирования открывается командой **config terminal** ("конфигурировать через терминал"), поданной в контексте администратора. Глобальный контекст конфигурирования содержит как непосредственно команды конфигурирования маршрутизатора, так и команды перехода в контексты конфигурирования подсистем маршрутизатора.

Контекст конфигурирования интерфейса открывается командой

interface *имя_интерфейса* (например, **interface serial0**), поданной в глобальном контексте конфигурирования;

Контекст конфигурирования процесса динамической маршрутизации открывается командой **router** *протокол номер_процесса* (например, **router ospf 1**, поданной в глобальном контексте конфигурирования).

Существует множество других контекстов конфигурирования. Некоторые контексты конфигурирования находятся внутри других контекстов конфигурирования.

Вид приглашения командной строки в контекстах конфигурирования, которые будут встречаться наиболее часто:

<code>router(config)#</code>	/глобальный/
<code>router(config-if)#</code>	/интерфейса/
<code>router(config-router)#</code>	/динамической маршрутизации/
<code>router(config-line)#</code>	/терминальной линии/

Запомните вид приглашений командой строки во всех вышеуказанных контекстах и правила перехода из контекста в контекст. В дальнейшем примеры команд всегда будут даваться вместе с приглашениями, из которых студенты должны определять контекст, в котором подается команда. Примеры не будут содержать указаний, как попасть в необходимый контекст.

Выход из глобального контекста конфигурирования в контекст администратора, а также выход из любого подконтекста конфигурирования в контекст верхнего уровня производится командой **exit** или **Ctrl-Z**. Кроме того, команда **end**, поданная в любом из контекстов конфигурирования немедленно завершает процесс конфигурирования и возвращает оператора в контекст администратора.

Любая команда конфигурации вступает в действие немедленно после ввода, а не после возврата в контекст администратора.

Упрощенная схема контекстов представлена на рис. 4.7.

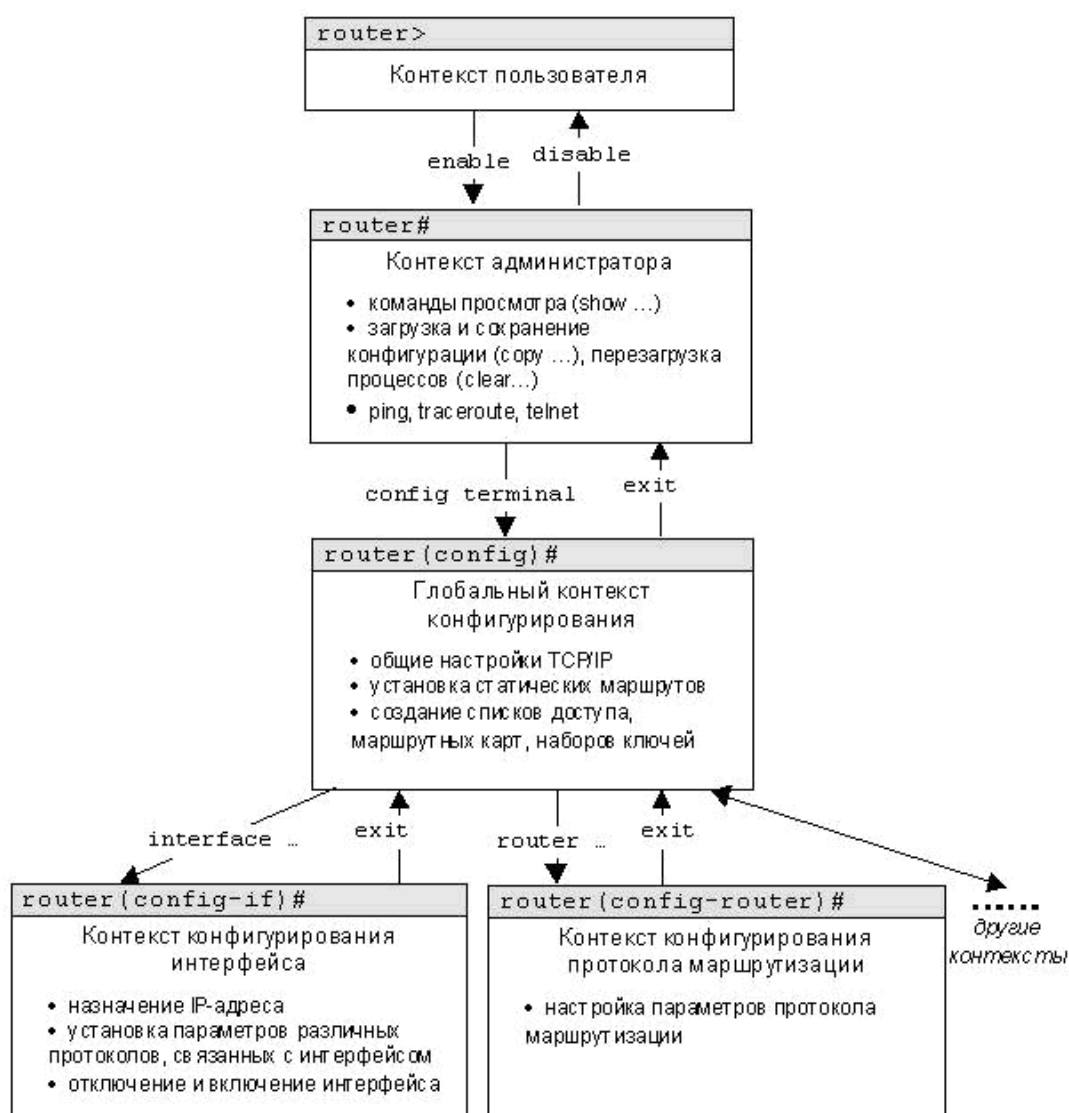


Рис. 4.7. Схема контекстов Cisco IOS

Все команды и параметры могут быть сокращены (например, **"enable"** - **"en"**, **"configure terminal"** - **"conf t"**); если сокращение окажется неоднозначным, маршрутизатор сообщит об этом, а по нажатию табуляции выдаст варианты, соответствующие введенному фрагменту.

В любом месте командной строки для получения помощи может быть использован вопросительный знак:

```

router#?      /список всех команд данного контекста с комментариями/
router#co?    /список всех слов в этом контексте ввода, начинающихся на
"co" - нет пробела перед "?"/
  
```

router#conf ? /список всех параметров, которые могут следовать за командой config – перед "?" есть пробел/

Список команд сгруппирован в соответствии с контекстами, в котором они применяются. В данном списке собраны те команды конфигурирования, которые необходимы для выполнения всех лабораторных работ.

4.4.1. Глобальный контекст конфигурирования

Команда «Access-list»

Критерии фильтрации задаются в списке операторов разрешения и запрета, называемом списком доступа. Строки списка доступа сравниваются с IP-адресами и другой информацией пакета данных последовательно в том порядке, в котором были заданы, пока не будет найдено совпадение. При совпадении осуществляется выход из списка. При этом работа списка доступа напрямую зависит от порядка следования строк.

Списки доступа имеют 2 *правила*: permit – разрешить, и deny – запретить. Именно они определяют, пропустить пакет дальше или запретить ему доступ.

Списки доступа бывают 2-ух типов: standard – стандартные (номера с 1 до 99) и extended – расширенные (номера с 100 до 199). Различия заключаются в возможности фильтровать пакеты не только по ip-адресу, но и по другим параметрам.

Формат команды (стандартные списки доступа):

access-list номер_списка/имя правило A.B.C.D a.b.c.d , где A.B.C.D a.b.c.d – ip-адрес и подстановочная маска соответственно.

Пример выполнения команды:

```
Router(config)#access-list 10 deny 192.168.3.0 0.0.0.3
Router(config)#
```

Данная команда означает, что данный список доступа блокирует любые пакеты с ip-адресами 192.168.3.1 - 192.168.3.3.

Команда «Enable secret»

Обычно при входе в привилегированный режим требуется ввести пароль. Данная функция позволяет предотвратить несанкционированный доступ в данный режим, ведь именно из него можно изменять конфигурацию устройства. Данная команда позволяет установить такой пароль.

Формат команды:

```
enable secret пароль
```

Пример выполнения команды:

```
Switch(config)#enable secret 123
Switch(config)#
%SYS-5-CONFIG_I: Configured from console by console
Switch#exit
Switch con0 is now available
Press RETURN to get started.
Switch>enable
Password:
```

После того, как был установлен пароль, при попытке входа в привилегированный режим, коммутатор будет требовать от пользователя его ввести – в противном случае вход будет невозможен.

Команда «Interface»

Команда для входа в режим конфигурирования интерфейсов конфигурируемого устройства. Данный режим представляет собой одно из подмножеств режима глобального конфигурирования и позволяет

настраивать один из доступных сетевых интерфейсов (fa 0/0, s 2/0 и т.д.). Все изменения, вносимые в конфигурацию коммутатора в данном режиме относятся только к выбранному интерфейсу.

Формат команды (возможны 3 варианта):

```
interface тип порт  
interface тип слот/порт  
interface тип слот/подслот/порт
```

Примеры выполнения команды:

```
Router(config-if) #  
Switch(config)#interface vlan 1  
Switch(config-if) #  
Router(config)#interface s 3/0
```

После введения данной команды с указанным интерфейсом пользователь имеет возможность приступить к его конфигурированию. Необходимо заметить, что, находясь в режиме конфигурирования интерфейса, вид приглашения командной строки не отображает имя данного интерфейса.

Команда «Ip route»

Статическая маршрутизация предполагает фиксированную структуру сети: каждый маршрутизатор в сети точно знает, куда нужно отправлять пакет, чтобы он был доставлен по назначению. Для этого можно прописать статические маршруты, используя данную команду. Команда может быть записана в двух форматах:

Первый формат команды:

```
ip route A.B.C.D a.b.c.d A1.B1.C1.D1 ,
```

где A.B.C.D и a.b.c.d – сетевой адрес и маска подсети, куда необходимо доставить пакеты, A1.B1.C1.D1 – ip-адрес следующего маршрутизатора в пути или адрес сети другого маршрутизатора из

таблицы маршрутизации, куда должны переадресовываться пакеты;

Второй формат команды:

```
ip route A.B.C.D a.b.c.d выходной_интерфейс_маршрутизатора
```

Примеры выполнения команды:

```
Router(config)#ip route 76.115.253.0 255.0.0.0 76.115.252.0
```

```
Router(config)#
```

Данной командой указывается маршрут, по которому пакеты из одной подсети будут доставляться в другую. Маршрут по умолчанию (Router(config)#ip route 0.0.0.0 0.0.0.0 serial 2/0) указывает, что пакеты, предназначенные узлам в другой подсети должны отправляться через данный шлюз.

Команда «Hostname»

Данная команда используется для изменения имени конфигурируемого устройства.

Формат команды:

```
hostname новое_имя
```

Пример выполнения команды:

```
R1(config)#
```

```
Router(config)#hostname R1
```

Как видно, маршрутизатор поменял своё имя с Router на R1.

Команда «Router rip»

RIP – Routing Information Protocol – протокол динамической маршрутизации. При его использовании отпадает необходимость вручную прописывать все маршруты – необходимо лишь указать адреса сетей, с которыми нужно обмениваться данными. Данная команда позволяет включить rip-протокол.

Пример выполнения команды:

```
Router(config-router)#  
Router(config)#router rip
```

Данная команда включает rip-протокол на данном маршрутизаторе. Дальнейшая настройка производится из соответствующего контекста маршрутизации, описанного отдельно.

4.4.2. Контекст конфигурирования интерфейса

Команда «Ip access-group»

Данная команда используется для наложения списков доступа. Список накладывается на конкретный интерфейс, и указывается один из 2-ух параметров: in (на входящие пакеты) или out (на исходящие). Необходимо знать, что на каждом интерфейсе может быть включен только один список доступа.

Формат команды:

```
ip access-group номер_списка/имя_параметр
```

Пример выполнения команды:

```
Router(config-if)# ip access group 10 in  
Router(config-if)#
```

В данном примере на выбранный интерфейс накладывается список доступа под номером 10: он будет проверять все входящие в интерфейс пакеты, так как выбран параметр in.

Команда «Bandwidth»

Данная команда используется только в последовательных интерфейсах и служит для установки ширины полосы пропускания. Значение устанавливается в килобитах.

Формат команды:

```
bandwidth ширина_полосы_пропускания
```

Пример выполнения команды:

```
Router(config)#interface serial 2/0
```

```
Router(config-if)#bandwidth 560
```

```
Router(config-if)#
```

После выполнения данной команды ширина полосы пропускания для serial 2/0 будет равна 560 kbits.

Команда «Clock rate»

Для корректной работы участка сети, где используется последовательный сетевой интерфейс, один из коммутаторов 3-его уровня должен предоставлять тактовую частоту. Это может быть окончное кабельное устройство DCE. Так как маршрутизаторы CISCO являются по умолчанию устройствами DTE, то необходимо явно указать интерфейсу на предоставление тактовой частоты, если этот интерфейс работает в режиме DCE. Для этого используют данную команду (значение устанавливается в битах в секунду).

Формат команды:

```
clock rate тактовая_частота
```

Пример выполнения команды:

```
Router(config)#interface serial 2/0
```

```
Router(config-if)#clock rate 56000
```

```
Router(config-if)#
```

После выполнения данной команды тактовая частота для serial 2/0 будет равна 56000 bits per second.

Команда «Ip address»

Каждый интерфейс должен обладать своим уникальным ip-адресом – иначе взаимодействие устройств по данному интерфейсу не сможет быть осуществлено. Данная команда используется для задания ip-адреса выбранному интерфейсу.

Формат команды:

```
ip address A.B.C.D a.b.c.d ,
```

где A.B.C.D a.b.c.d – ip-адрес и маска подсети соответственно.

Пример выполнения команды:

```
Switch(config)#interface vlan 1  
Switch(config-if)#ip address 172.16.10.5 255.255.0.0  
Switch(config-if)#
```

Результат можно проверить командой

```
Switch#show ip interface vlan 1
```

Данной командой интерфейсу vlan 1 назначен ip-адрес 172.16.10.5 с маской подсети 255.255.0.0.

Команда «No»

Данная команда применяется в случае необходимости отменить действие какой-либо команды конфигурирования.

Формат команды:

```
no команда_которую_следует_отменить
```

Пример выполнения команды:

```
Switch(config-if)# no shutdown  
%LINK-5-CHANGED: Interface Vlan1, changed state to up  
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state  
to up  
Switch(config-if)#
```

В данном примере использовалась команда `shutdown`, которая отключает выбранный интерфейс. В итоге после выполнения `no shutdown` интерфейс включается.

Команда «Encapsulation»

Данная команда обеспечивает инкапсуляцию - метод, используемый многоуровневыми протоколами, в которых уровни добавляют заголовки в модуль данных протокола (protocol data unit - PDU) из вышележащего.

Формат команды:

```
encapsulation тип_протокола
```

Пример выполнения команды:

```
Router(config-if)#encapsulation frame-relay
```

```
Router(config-if)#
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed  
state to up
```

В данном примере маршрутизатор Router сможет пересылать пакеты по протоколу Frame Relay.

4.4.3. Контекст администратора

Команда «Configure terminal»

Для конфигурирования устройства, работающего под управлением IOS, следует использовать привилегированную команду `configure`. Эта команда переводит контекст пользователя в так называемый «режим глобальной конфигурации» и имеет три варианта:

1. конфигурирование с терминала;
2. конфигурирование из памяти;
3. конфигурирование через сеть.

Из режима глобальной конфигурации можно делать изменения,

который касаются устройства в целом. Также данный режим позволяет входить в режим конфигурирования определенного интерфейса.

Пример выполнения команды:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Переход в режим глобальной конфигурации, о чем свидетельствует изменившийся вид приглашения командной строки.

Команда «Сору»

После настройки коммутатора рекомендуется сохранять его текущую конфигурацию. Информация помещается в энергонезависимую память и хранится там столько, сколько нужно. При необходимости все настройки могут быть восстановлены или сброшены.

Формат команды:

```
copy running-config startup-config - команда для сохранения
конфигурации

copy startup-config running-config - команда для загрузки
конфигурации
```

Пример выполнения команды:

```
Switch#copy running-config startup-config
Building configuration...
[OK]
Switch#
```

В данном примере текущая конфигурация коммутатора была сохранена в энергонезависимую память.

Команда «Show»

Show (англ. - показывать) – одна из наиболее важных команд, используемых при настройке коммутаторов. Она применяется для просмотра информации любого рода и применяется практически во всех контекстах. Эта команда имеет больше всех параметров.

Здесь будут рассмотрены только те параметры, которые требуются в рамках данного курса. Другие параметры студент может изучить самостоятельно.

Параметр «running-config» команды «Show»

Для просмотра текущей работающей конфигурации коммутатора используется данная команда.

Пример выполнения команды:

```
Switch#show running-config
!
version 12.1
!
hostname Switch
...
```

На экран выводится текущие настройки коммутатора.

Параметр «startup-config» команды «Show»

Для просмотра сохраненной конфигурации используется данная команда.

Пример выполнения команды:

```
Switch#show startup-config
Using 1540 bytes
!
version 12.1
!
```

...

Если энергонезависимая память не содержит информации, тогда коммутатор выдаст сообщение о том, что конфигурация не была сохранена.

Пример выполнения команды:

```
Switch #show startup-config
startup-config is not present
Switch #
```

Вывод сообщения о том, что в памяти отсутствует какая-либо информация.

Параметр «ip route» команды «Show»

Данная команда применяется для просмотра таблицы маршрутов.

Пример выполнения команды:

```
Router#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial2/0
S    192.168.3.0/24 is directly connected, Serial2/0
S    192.168.4.0/24 is directly connected, Serial2/0
S    192.168.5.0/24 is directly connected, Serial2/0
S*   0.0.0.0/0 is directly connected, Serial2/0
```

Router#

Параметр «ip protocols» команды «Show»

Данная команда используется для просмотра протоколов маршрутизации, включенных на данном устройстве.

Пример выполнения команды:

```
Router#show ip protocols

Routing Protocol is "rip"

Sending updates every 30 seconds, next due in 18 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip

Default version control: send version 1, receive any version

  Interface                Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0           1      2 1
  Serial2/0                 1      2 1

Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:

  192.168.1.0
  192.168.2.0

Passive Interface(s):

Routing Information Sources:

  Gateway                Distance    Last Update
  192.168.2.2             120
Distance: (default is 120)

Router#
```

Выводится информация о включенных протоколах маршрутизации.

Команда «Ping»

Для проверки связи между устройствами сети можно использовать данную команду. Она отправляет эхо-запросы указанному узлу сети и фиксирует поступающие ответы.

Формат команды:

```
ping A.B.C.D
```

Пример выполнения команды:

```
Router#ping 77.134.25.133
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 77.134.25.133, timeout is 2
seconds:
..!!!!
Success rate is 60 percent (3/5)
```

Каждый ICMP-пакет, на который был получен ответ, обозначается восклицательным знаком, каждый потерянный пакет – точкой.

4.4.4. Контекст пользователя

Команда «Enable»

Выполнение конфигурационных или управляющих команд требует вхождения в привилегированный режим, используя данную команду.

Пример выполнения команды:

```
Router>enable
Router#
```

При вводе команды маршрутизатор перешел в привилегированный режим. Для выхода из данного режима используется команда `disable` или `exit`.

Также следует отметить, что в данном контексте можно пользоваться командой `show` для просмотра некоторой служебной информации.

4.4.5. Контекст конфигурирования маршрутизации

Команда «Network»

Данной командой указывают адреса сетей, которые будут доступны данному маршрутизатору.

Формат команды:

```
network A.B.C.D , где A.B.C.D – адрес сети
```

Пример выполнения команды:

```
Router(config-router)#network 192.168.3.0
```

Данная команда означает, что пакеты, направленные в подсеть 192.168.3.0 будут отправляться через данный шлюз.

4.4.6 Контекст конфигурирования динамического распределения ip адресов.

Вход в данный контекст осуществляется из контекста администратора с помощью команды **ip dhcp pool**.

Команда «ip dhcp pool»

Данная команда организует набор динамического распределения ip-адресов.

Формат команды:

```
ip dhcp pool название_набора
```

Пример выполнения команды:

```
Router(config)#ip dhcp pool POOL1
```

```
Router(dhcp-config)#
```

В данном примере организуется набор динамического распределения ip-адресов под именем POOL1.

Команда «network»

Данная команда задает диапазон IP адресов выбранного набора.

Формат команды:

```
network название_набора [маска]
```

Пример выполнения команды:

```
Router(dhcp-config)#network 192.168.0.0 255.255.255.0
```

В данном примере создается диапазон адресов 192.168.0.x с маской подсети 255.255.255.0.

Команда «default-router»

При включении динамического распределения ip-адресов клиенты начинают посылать пакеты маршрутизатору, назначенному по умолчанию. Его IP адрес должен быть из той же подсети. Данная команда задает этот маршрутизатор.

Формат команды:

```
default-router адрес_1 [адрес_2, ...]
```

Пример выполнения команды:

```
Router(dhcp-config)#default-router 192.168.0.1
```

Команда «dns-server»

Данная команда задает адрес DNS сервера.

Формат команды:

```
dns-server адрес_1 [адрес_2, ...]
```

Пример выполнения команды:

```
Router(dhcp-config)#dns-server 192.168.0.1
```

4.5. Пример выполнения задания.

Задание: Построить локальную сеть, состоящую из сегмента на основе коммутатора из 5 компьютеров и сервера. Коммутатор соединен с маршрутизатором, к которому также подключен сервер. Необходимо задать статические IP адреса сетевым интерфейсам маршрутизаторов, локальных компьютеров и серверов. Установить на маршрутизаторах пароли для доступа к привилегированному режиму. Настроить маршрутизацию по протоколу RIP. Добиться возможности пересылки данных по протоколу ICMP между всеми объектами сети.

Расставляем на рабочем поле необходимые узлы, используя браузер в нижней части окна (рис. 4.8). Соединяем узлы в соответствии с заданием с помощью витой пары. Сервер с маршрутизатором соединяется кросс-овером. Соединения, обозначенные зеленым цветом указывают, что они активны, оранжевым – что они на стадии подключения, красным – не рабочие.

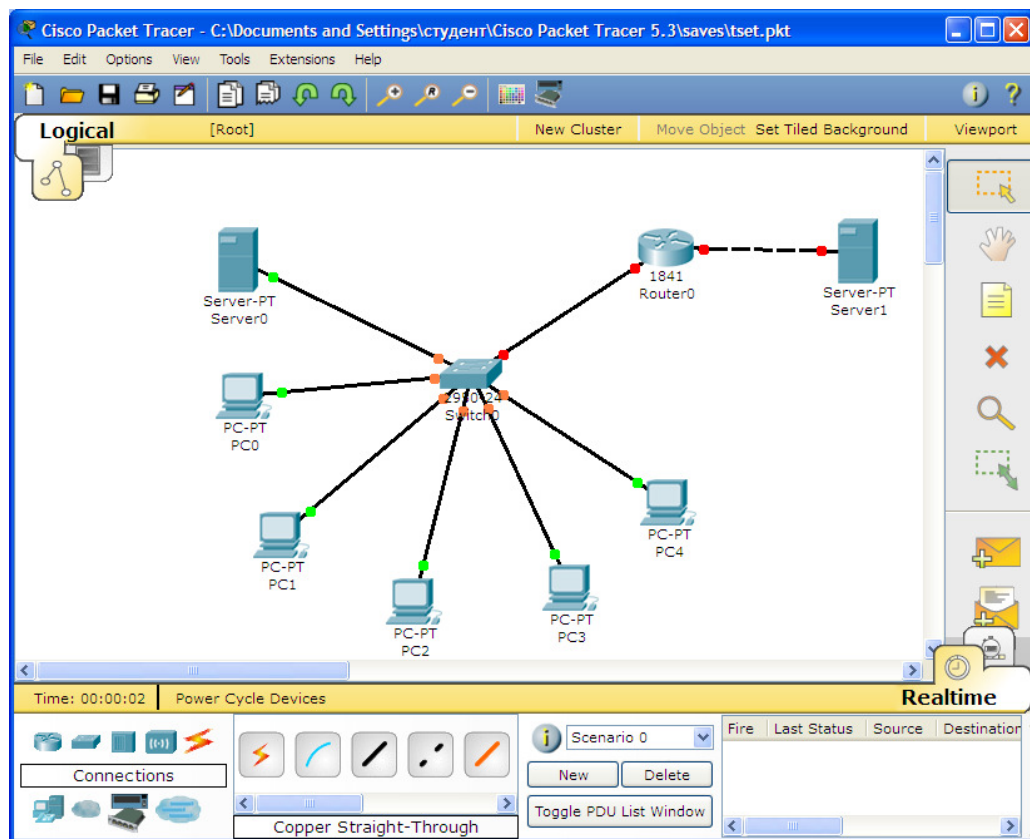


Рисунок 4.8. Рабочее поле

Зададим ip-адреса узлам сегмента в диапазоне 192.168.0.x, а серверу, подключенному к маршрутизатору – 192.168.1.1. Маска подсети – 255.255.255.0. (Рис. 4.9).

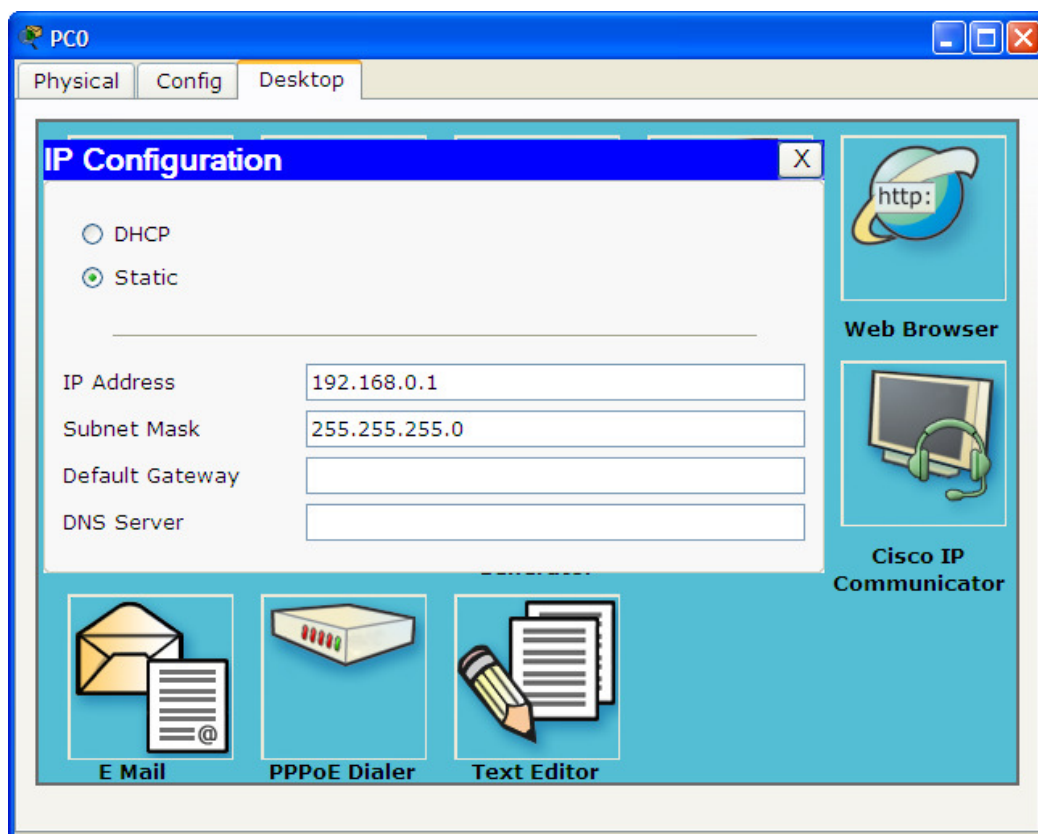


Рисунок 4.9. IP конфигурация рабочей станции.

Зададим соответствующие ip адреса на интерфейсах маршрутизатора и включим эти порты. (Рис. 4.10).

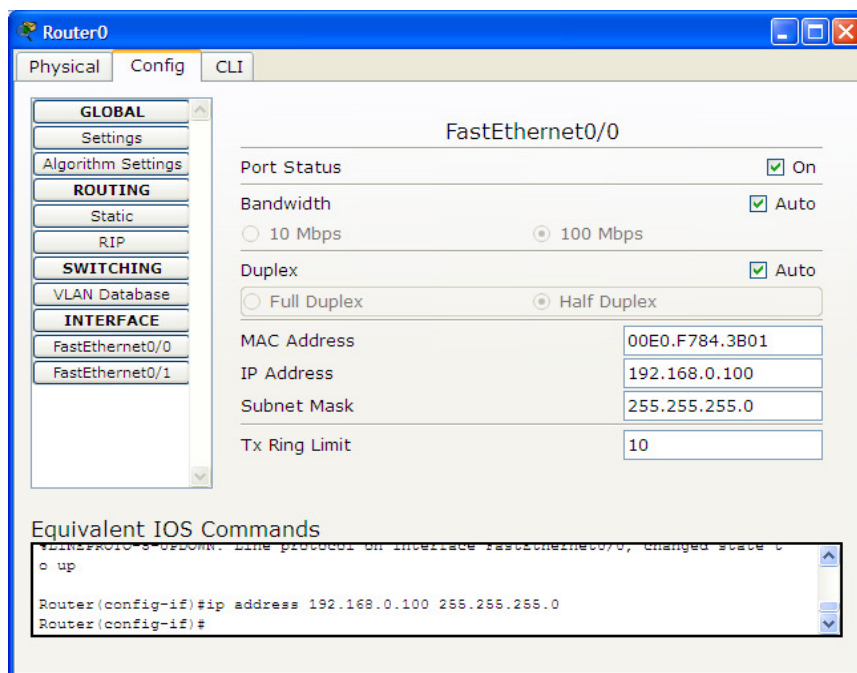


Рисунок 4.10. IP конфигурация маршрутизатора.

Зайдем в Command Line Interface маршрутизатора и с помощью команды `enable secret` зададим пароль для доступа в привилегированный режим и сохраним конфигурацию. (Рис. 4.11).

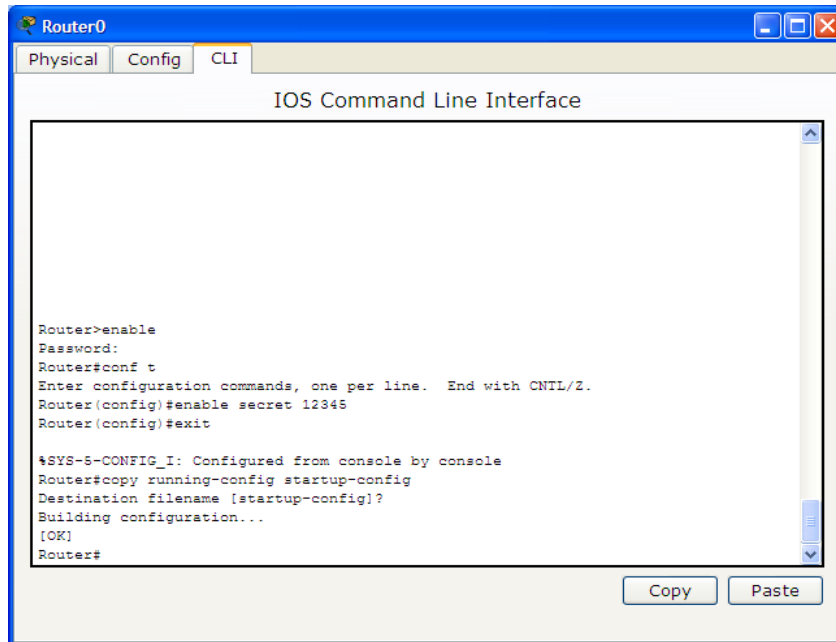


Рисунок 4.11. Работа в Command Line Interface.

Для настройки маршрутизации по протоколу RIP откроем вкладку Config в окне свойств маршрутизатора и выберем пункт RIP. Зададим там адреса всех подсетей, которым разрешено общение. (Рис. 4.12).

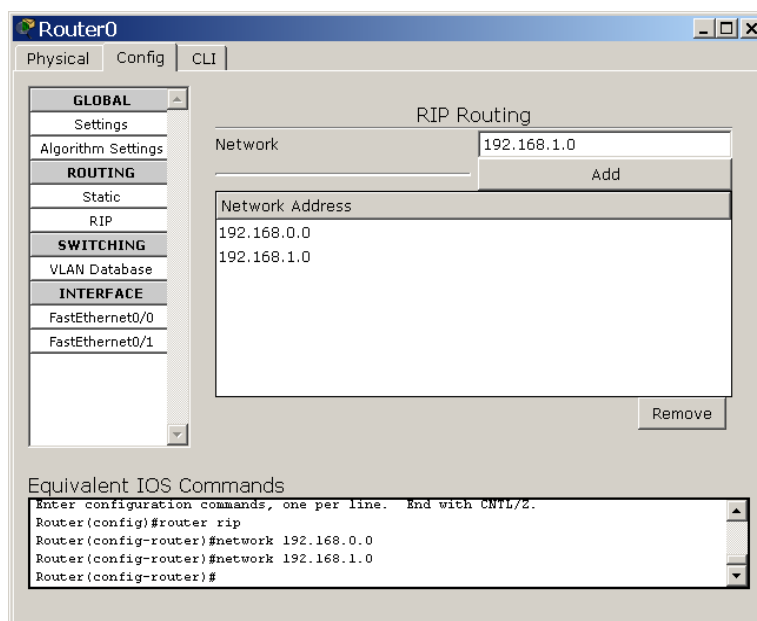


Рисунок 4.12. Настройка маршрутизации по протоколу RIP.

Проверяем доступность рабочих станций друг для друга. Для этого в правом столбце выбираем инструмент Add simple PDU и выбираем станцию-отправитель и станцию-получатель. Убеждаемся, что передача завершена успешно. (Рис. 4.13).

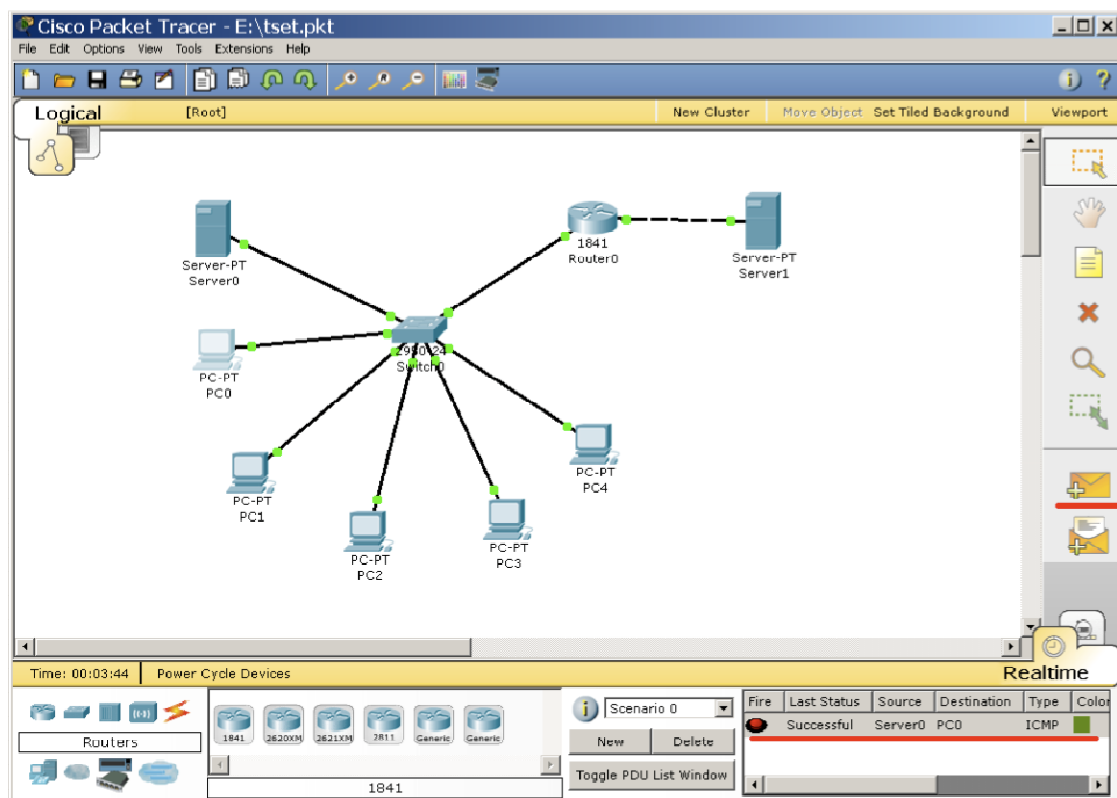


Рисунок 4.13. Проверка доступности узлов в сети.

4.6. Задания.

Лабораторная работа №5. Командная строка операционной системы IOS.

Изучить интерфейс программы Cisco Packet Tracer. Изучить команды и контексты, описанные в пункте 4.4. Опробовать их в эмуляторе терминала. Ответить на контрольные вопросы.

Лабораторная работа №6. Статическая маршрутизация.

Построить сеть из трех сегментов, каждый из которых состоит из С,

Д и **Е** рабочих станций соответственно. Каждый сегмент построен на базе коммутатора, и каждый коммутатор подключен к отдельному маршрутизатору. Шлюзом для каждого сегмента служит соответствующий маршрутизатор. Маршрутизаторы соединены между собой с помощью интерфейса DTE. Необходимо задать IP адреса сетевым интерфейсам маршрутизаторов и локальных компьютеров. Задать параметр Clock Rate на маршрутизаторах. Установить на маршрутизаторах пароли для доступа к привилегированному режиму. Настроить статическую маршрутизацию и добиться возможности пересылки данных по протоколу ICMP между всеми объектами сети.

Лабораторная работа №7. Динамическое распределение IP-адресов и DNS.

Построить сеть, состоящую из двух сегментов на основе коммутаторов. Сегмент №1 содержит **В** рабочих станций, сегмент №2 – **С** рабочих станций и сервер. Маршрутизатор является шлюзом. Сегменты соединены маршрутизатором. В первом сегменте IP адреса раздаются маршрутизатором динамически, во втором IP адреса заданы статически. Настроить маршрутизацию по протоколу RIP. Сервер является DNS и веб-сервером. Настроить на сервере веб-страницу произвольного формата. Добиться возможности пересылки данных по протоколу ICMP между всеми объектами сети. Добиться просмотра веб-страницы с сервера во встроенных браузерах рабочих станций.

4.7. Контрольные вопросы.

1. Сетевое оборудование и его функции.
2. Стек протоколов TCP/IP.
3. Отличие между различными стандартами сетей Ethernet.

4. Формат кадров в сети Ethernet.
5. Сети wi-fi – основные стандарты и принципы работы.
6. Назначение шлюза.
7. Маршрутизация.
8. Эталонная модель OSI/ISO.

4.8. Варианты заданий.

Вариант	A	B	C	D	E
1	2	3	4	9	10
2	3	4	5	9	2
3	4	5	6	7	8
4	5	6	3	8	9
5	6	7	8	2	3
6	7	4	3	6	2
7	8	9	5	2	3
8	9	4	2	3	4
9	7	3	5	4	6
10	2	4	6	8	3
11	4	6	8	3	1
12	6	8	5	1	3
13	8	4	1	3	5
14	4	6	3	5	7
15	1	3	5	7	2
16	3	5	7	2	4
17	5	7	2	4	6
18	7	3	4	6	8
19	2	4	6	8	4
20	4	6	8	5	2
21	2	5	8	3	6
22	5	8	3	6	9
23	8	3	6	9	4
24	3	6	9	4	7
25	6	9	4	7	3

Лабораторная работа №8 и №9.

«Исследование протоколов сетевого и транспортного уровней IP-сетей с помощью анализатора протоколов».

5.1. Цель лабораторных работ.

Развитие практических навыков работы с протоколами стека TCP/IP и исследование возможностей протоколов ICMP, UDP, TCP.

5.2. Необходимое оборудование.

Аппаратные требования

- IBM - совместимый ПК в составе сети Интернет.
- 128 Мбайт оперативной памяти
- 5 Мбайт свободного места на HDD
- Используемый шлюз в Интернет должен пропускать ICMP, TCP и UDP трафик.

Программные требования

Лабораторная работа выполняется с помощью любых средств анализа сетевого трафика, в том числе с помощью встроенных средств и онлайн-сервисов. Рекомендованными средствами являются:

- Пакетный анализатор Wireshark <http://www.wireshark.org/>
- Сетевой анализатор NetInfo <http://netinfo.tsarfin.com/>

5.3. Теоретическая часть.

Протоколы - это правила работы программного обеспечения.

Стек протоколов - набор взаимодополняющих и тесно связанных друг с другом протоколов.

Термин "стек протоколов" происходит из концепции представления сети в виде вертикально расположенных уровней и сложенных в стек

протоколов и относится к любой комбинации сетевых уровней и соответствующих протоколов.

В настоящей лабораторной работе предметом исследований является стек протоколов TCP/IP – наиболее распространенный и являющийся основным в сети Интернет.

IP (Internet Protocol) - протокол межсетевого взаимодействия, является протоколом сетевого уровня модели OSI и отвечает за перемещение данных между сетевыми компьютерами в Интернет.

TCP(Transmission Control Protocol) - протокол управления передачей, который перемещает данные между прикладными программами.

UDP (User Datagram Protocol) - протокол пользовательских дейтаграмм, который также перемещает данные между приложениями. Он - более простой и менее надежный, чем TCP.

ICMP (Internet Control Message Protocol) - протокол управляющих сообщений Интернет, который управляет сетевыми сообщениями об ошибках и другими ситуациями, требующими вмешательства сетевых программ.

Схема движения данных.

Данные по сети передаются в три этапа:

- Информация должна пройти между приложениями и сетью. Это путь сквозь стек протоколов вниз к транспортному уровню.
- Определение сетью адреса получателя данных.
- Маршрутизация данных и прохождение данных сквозь стек протоколов вверх к сетевому приложению.

Схема движения данных пользователя представлена на рис. 5.1.

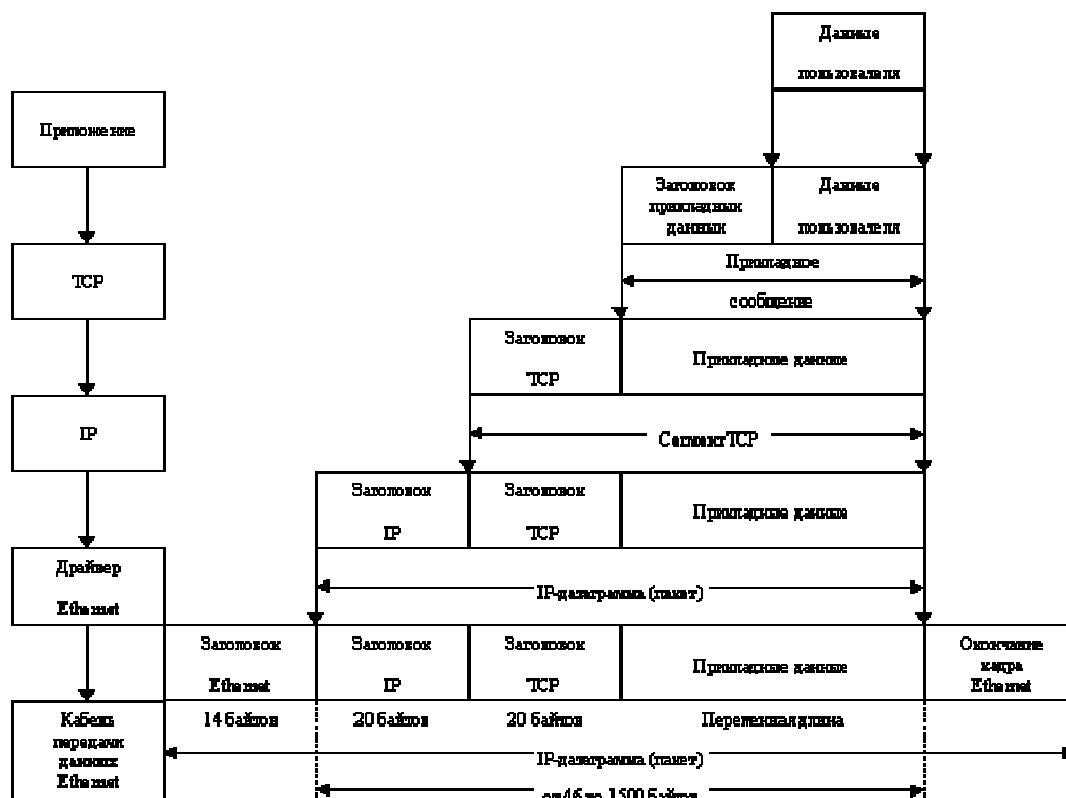


Рис. 5.1. Схема движения данных пользователя.

Протокол IP

Формат IP-дейтаграммы и поля заголовка представлены на рис.5.2.

0		15	16	31
Версия (TOS) 4 бита	Длина заголовка (HLEN) 4 бита	Тип службы (TOS) 8 битов	Общая длина пакета в байтах 16 битов	
Идентификатор фрагмента 16 битов			Флаги 3 бита	Смещение фрагмента 13 битов
Время существования (TOS) 8 битов		Протокол 8 битов	Контрольная сумма заголовка 16 битов	
IP-адрес источника 32 бита				
IP-адрес получателя 32 бита				
Опции (если есть)			Заполнение (если нужно)	
Данные				

Рис. 5.2. Формат IP-дейтаграммы и поля заголовка.

Поля IP - протокола.

Номер версии VERS. Протокол IP постоянно развивается, необходимо знать, номер версии, чтобы правильно интерпретировать дейтаграмму.

Длина заголовка (HLEN) в 32 разрядных словах. Чаще всего длина IP-заголовка равна 20 байтам, поэтому данное поле обычно содержит число 5 (0101).

Тип сервиса (TOS). Поле "тип сервиса" разделено на 5 подразделов (рис.5.3).



Рис. 5.3. Формат поля TOS.

Первое трехразрядное субполе *приоритет (precedence)* редко применяется на практике. Последнее безымянное одноразрядное субполе всегда содержит 0. Между ними находятся четыре одноразрядных субполя, которые и называют собственно битами TOS. Каждому из четырех битов TOS сопоставлен определенный критерий доставки дейтаграмм: *минимальная задержка, максимум пропускной способности, максимум надежности и минимум стоимости*. Только один бит TOS может быть установлен в 1. По умолчанию все четыре бита равны 0, что означает отсутствие особых требований, то есть обычный сервис.

Длина пакета. Поле "длина пакета" задает длину IP-пакета, включая сам заголовок. Если локальная сеть построена по технологии Ethernet, уровень соединения инкапсулирует IP-дейтаграммы в кадры Ethernet перед передачей их в Интернет. Спецификация Ethernet ограничивает длину пакета до 1500 байтов.

Идентификатор. Наличие этого поля обусловлено фрагментацией пакетов в Интернет. Сетевые компьютеры используют поле с целью однозначной идентификации каждого посланного фрагмента для

дейтаграммы, к которой он относится.

Флаги и смещение. Информация, содержащаяся в полях идентификации флагов и смещения фрагмента позволяет правильно собрать фрагментированный пакет.

Время существования (TTL). Время существования определяет «время жизни» пакета в сети и не дает пакету возможность быть вечным скитальцем.

Протокол. Поле «протокол» в IP-заголовке указывает на протокол-источник данных, инкапсулированных в IP-дейтаграмму.

Контрольная сумма заголовка. Поле контрольной суммы в IP-заголовке содержит 16-ти битное число, являющееся контрольной суммой только для IP-заголовка.

IP-адрес источника и получателя. 32-битное поле «адрес источника» содержит IP-адрес компьютера - отправителя данных (вернее адрес его сетевого интерфейса).

Адрес получателя. Адрес получателя является 32-битным адресом пункта назначения пакета. В случае широковещательной передачи он состоит из единиц.

Опции IP. Это поле позволяет тестировать разнообразные сетевые приложения.

Протокол пользовательских дейтаграмм (UDP)

UDP-протокол умеет распознавать то приложение среди многих, работающих внутри компьютера, которому предназначены данные. Как правило, сеть назначает таким приложениям определенный номер порта. UDP пользуется дейтаграммами для доставки данных. Точно так же, как IP прицепляет к данным IP-заголовки, UDP прицепляет к ним UDP-заголовок, структура которого представлена на рис.4.

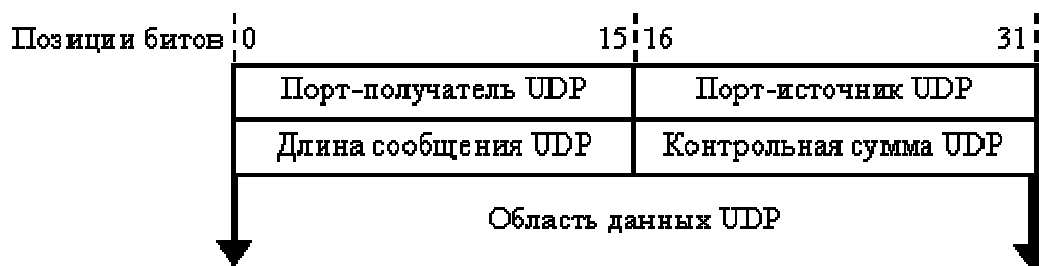


Рис. 5.4. Структура UDP – заголовка.

Длина UDP-заголовка - восемь байтов. Поля портов состоят из 16-битных целых чисел, представляющих номера портов протоколов. Поле "порт-источник" содержит номер порта, которым пользуется приложение-источник данных. Поле "порт-получатель" соответственно указывает на номер порта приложения-получателя данных. Поле "длина сообщения" определяет длину (в байтах) UDP-дейтаграммы, включая UDP-заголовок. Наконец, поле "контрольная сумма", в отличие от контрольной суммы IP-заголовка, содержит результат суммирования всей UDP-дейтаграммы, включая ее данные, область которых начинается сразу после заголовка.

Модуль UDP отслеживает появление вновь прибывших дейтаграмм, сортирует их и распределяет в соответствии с портами назначения.

Протокол TCP

Протокол Транспортного уровня модели OSI служит для передачи данных между сетевым и прикладным уровнями сетевой модели. TCP призван обеспечивать надежную, потоковую, ориентированную на соединение службу доставки данных.

TCP пытается оптимизировать пропускную способность сети, то есть увеличивает производительность доставки пакетов в Интернет. Для этого он динамически управляет потоком данных в соединении. Если буфер приемника данных переполняется, TCP просит передающую сторону снизить скорость передачи.

Данные TCP всегда переносит IP, то есть данные TCP всегда

упаковываются в IP-дейтаграммы.

Для обеспечения надежной доставки и правильной последовательности данных в потоке ТСР пользуется принцип скользящего окна и тайм-аута. Принцип скользящего окна позволяет послать несколько сообщений и только потом ожидать подтверждения. ТСР накладывает окно на поток данных, ожидающих передачи, и передает все данные, попавшие в окно. Приняв подтверждение о доставке всех данных, ТСР перемещает окно дальше по потоку и передает следующие попавшие в него сообщения. Работая сразу с несколькими сообщениями, ТСР может одновременно "выставить" их на сетевой канал и только потом ожидать прихода подтверждения. Метод скользящего окна значительно увеличивает производительность соединения, а также эффективность циклов обмена сообщениями и подтверждениями об их доставке.

0 15		16 31	
Порт источника 16 бит		Порт назначения 16 бит	
Позиционный номер 32 бита			
Квитанция 32 бита			
Длина заголовка 4 бита	Резерв 16 бит	Флаги	Размер окна приема 16 бит
Контрольная сумма 16 бит		Указатель границы срочных данных 16 бит	
Опции (если таковые имеются)			
Данные (если таковые имеются)			

Рис.5.5. Формат заголовка сегмента ТСР.

ТСР регулирует полосу пропускания сети, договариваясь с другой стороной о некоторых параметрах данных. Причем процесс регулировки

происходит на протяжении всего соединения TCP. В частности, регулировка заключается в изменении размеров скользящего окна. Если сеть загружена не сильно и вероятность столкновения данных минимальна, TCP может увеличить размер скользящего окна. При этом скорость выдачи данных на канал увеличивается и соединение становится более эффективным.

Если, наоборот, вероятность столкновения данных велика, TCP уменьшает размер скользящего окна.

Как правило, модуль TCP передает несколько сегментов, прежде чем скользящее окно заполнится целиком. Большинство систем в Интернет устанавливают окно равным по умолчанию 4096 байтам. Иногда размер окна равен 8192 или 16384 байтам

Заголовок сегмента TCP представлен на рис.5. Обычно (при отсутствии опций) заголовок имеет размер 20 байтов. Напомним, что передаваемый TCP – сегмент с данными инкапсулируется в IP – дейтаграмму.

Номера портов источника и назначения (*source port number* и *destination port number*) идентифицируют взаимодействующие приложения.

Позиционный номер (*sequence number*) сегмента указывает то место в потоке данных от источника до конечного получателя, которое занимает первый байт содержащихся в этом сегменте данных. В начальном сегменте, посылаемом при установлении соединения, присутствует флаг SYN, а в поле позиционный номер содержит так называемый *начальный позиционный номер ISN(initial sequence number)*, выбранный данным хостом для этого нового соединения. Первому байту данных, переданному хостом по новому соединению, будет присвоен позиционный номер, равный ISN+1. Такой сдвиг в нумерации кратко формулируется правилом: флаг SYN поглощает одну позицию.

В поле **квитанция** (*acknowledgement - ACK*) передающей стороне

сообщается позиционный номер следующего в потоке данных сегмента, ожидаемого принимающей стороной. Это число всегда на единицу больше номера последнего успешно принятого байта.

Поле **размер заголовка** (*header length*) необходимо, поскольку в заголовке далее могут следовать поля опций переменной длины. Записанная в этом поле константа означает число отводимых под заголовок 32-разрядных слов, и, следовательно, длина заголовка не превышает 60 байтов. При отсутствии опций размер заголовка всегда равен 20 байтам.

В TCP-заголовке предусмотрены 6 двоичных **флагов** (*flags*) , причем некоторые из них могут быть установлены одновременно.

URG – флаг срочных данных. Поле **указатель границы срочных данных** заголовка имеет смысл только при URG =1

ACK – флаг квитирования. Поле **квитанция** имеет смысл только при ACK = 1.

PSH – флаг «проталкивания» (push). TCP-модуль хоста назначения должен незамедлительно отдать данные из сегмента своему приложению.

RST – флаг сброса соединения.

SYN – флаг синхронизации позиционных номеров сегментов при установлении соединения.

FIN – флаг окончания передачи. Он означает, что источник сегмента закончил передачу данных и закрывает свой канал вывода в текущем соединении.

Темп передачи потока данных в каждом направлении по TCP-соединению регулируется обеими участвующими в обмене сторонами благодаря тому, что каждая сторона объявляет свой размер окна приема (window size), то есть количество байтов, которое она в данный момент готова принять вслед за байтом, номер которого указан в поле **квитанция**.

Поле **контрольная сумма** (*TCP – checksum*) содержит значение, подсчитанное для всего сегмента, включая его заголовок и данные.

Указатель границы срочных данных (*urgent pointer*) действует лишь при условии, что в сегменте установлен флаг URG. Это положительная константа, равная смещению номера последнего байта срочных данных относительно позиционного номера в заголовке сегмента. Срочный режим (*urgent mode*) предусмотрен в TCP, чтобы в поток передаваемых обычных данных приложение могло внедрять цепочки каких-либо особым образом интерпретируемых байтов (например, команд) так, чтобы они были обнаружены и выделены из потока на принимающей стороне.

Открытие TCP соединения

Открытие TCP-соединения состоит из трех фаз.

1. Запрашивающая сторона (обычно это клиент) посылает сегмент с флагом SYN, указывая номер порта получателя (сервера) с которым хочет соединиться, а также начальный позиционный номер ISN(initial sequence number).
2. Сервер отвечает сегментом SYN, где сообщает свой начальный позиционный номер и одновременно подтверждает получение сегмента SYN от клиента – он устанавливает флаг ACK, а в качестве подтверждаемого позиционного номера указывает номер, на единицу больше принятого, то есть ISN клиента плюс один.
3. Клиент подтверждает получение сегмента SYN от сервера, выслав сегмент с флагом ACK и номером квитанции, равным принятому от сервера начальному позиционному номеру ISN плюс один.

Обмен этими тремя сегментами и составляет процедуру установления соединения. Часто такой механизм называют троекратным рукопожатием (*three-way handshake*).

Закрытие TCP соединения

Если для установления соединения необходим обмен тремя сегментами, то для его закрытия таковых требуется четыре. Поскольку соединение TCP является полнодуплексным (то есть данные могут передаваться в обоих направлениях независимо), каждое направление необходимо закрывать по отдельности. Закрытие одного направления называется полузакрытием (half-close). Согласно протоколу любая из сторон, закончив передачу данных, может послать сегмент FIN. Когда TCP-модуль получает сегмент FIN, он обязан уведомить обслуживаемое приложение, что другая сторона закрыла свое направление передачи данных.

Приход FIN означает лишь то, что поступление данных от партнера по этому соединению прекращается. Но TCP-модуль может посылать данные и после получения им FIN. Предоставляемая приложению возможность продолжать передачу по полузакрытому соединению на практике используется редко.

Говорят, что сторона, первой закрывающая соединение (то есть посылающая первый FIN), производит активное закрытие соединения (active close). Другая сторона (которая получает этот FIN и отвечает на него своим FIN) выполняет пассивное закрытие соединения (passive close). Итак:

1. TCP-модуль одной из сторон посылает сегмент FIN и тем самым закрывает поток данных со своей стороны.
2. В ответ на пришедший FIN TCP-модуль второй стороны посылает подтверждение полученного позиционного номера плюс один.
3. Приложение на второй стороне закрывает свой поток данных, и его TCP-модуль посылает FIN.
4. Первый хост отвечает сегментом ACK с квитанцией, равной позиционному номеру полученного им сегмента FIN плюс один.

Протокол управляющих сообщений ICMP

Протокол ICMP (Internet Control Message Protocol) служит для обмена сообщениями об ошибках и различных особых случаях, требующих обработки. ICMP-сообщения содержат управляющие данные, используемые либо на IP-уровне, либо на более высоком уровне (TCP или UDP). Некоторые ICMP-сообщения трансформируются в коды ошибок, возвращаемых пользовательским процессам. В иерархии протоколов ICMP часто относят к сетевому уровню, наряду с IP, но ICMP-сообщения инкапсулируются в IP-диаграммы. Структура ICMP-сообщения представлена на рис.5.6.

0 7	8 15	16 31
Тип (8 бит)	Код (8 бит)	Контрольная сумма (16 бит)
Содержание сообщения (зависит от типа и кода)		

Рис.5.6. Структура ICMP-сообщения

Первое слово (4 байта) содержит три поля, общие по смыслу и формату для любых разновидностей сообщений. Следующая затем содержательная часть сообщения форматируется по-разному в зависимости от типа сообщения.

Предусмотрено 15 различных значений для поля **тип** (*type*), которое идентифицирует разновидность ICMP-сообщения. Кроме того, некоторые типы ICMP-сообщений дополнительно используют значения поля **код**(*code*) для конкретизации тех или иных условий.

Поле контрольная сумма (*checksum*) относится ко всему ICMP-сообщению и является обязательным

Разновидности ICMP – сообщений

В таблице 5.1 приведены всевозможные разновидности ICMP-

сообщений, определяемые полями тип(type) и код (code). Последние два столбца таблицы позволяют отличить запросы и отклики на них от сообщений об ошибках. Необходимо различать эти две разновидности, потому что обработка ICMP-сообщения об ошибке имеет свою специфику.

Таблица 5.1 Разновидности ICMP-сообщений

Тип	Код	Описание	Запрос/Ответ	Ошибка
0	0	Эхо-ответ (echo reply)	+	
3		Адресат недоступен(destination unreachable)		
	0	сеть недоступна		+
	1	хост недоступен		+
	2	протокол недоступен		+
	3	порт недоступен		+
	4	необходима фрагментация, но есть флаг DF		+
	5	маршрутизация от источника невыполнима		+
	6	сеть назначения неизвестна		+
	7	хост назначения неизвестен		+
	8	хост источника изолирован(устарело)		+
	9	сеть назначения административно закрыта		+
	10	хост назначения административно закрыт		+
	11	сеть недоступна для данного типа сервиса TOS		+
	12	хост недоступен для данного типа сервиса TOS		+
	13	связь административно закрыта фильтром		+
	14	нарушение старшинства хостов		+
	15	действует отключение по старшинству		+
4	0	Прикрыть источник(source quench)		+
5		Перенаправление(redirect)		
	0	перенаправить путь на сеть		+
	1	перенаправить путь на хост		+
	2	перенаправить путь на сеть для типа сервиса TOS		+

	3	перенаправить путь на хост для типа сервиса TOS		+
8	0	Эхо-запрос(echo request)	+	
9	0	Объявление маршрутизатора(router advertisement)	+	
10	0	Запрос маршрутизатора(router solicitation)	+	
11		Срок истек(time exceeded)		
	0	срок истек на переходе(TTL = 0)		+
	1	срок истек при сборке		+
12		Нарушены параметры дейтаграммы		
	0	испорчен IP-заголовок		+
	1	отсутствует необходимая опция		+
13	0	Запрос отсчета времени(timestamp request)	+	
14	0	Отклик отсчета времени(timestamp reply)	+	
15	0	Запрос информации(устарело)	+	
16	0	Информационный отклик(устарело)	+	
17	0	Запрос адресной маски(address mask request)	+	
18	0	Ответ адресной маски(address mask reply)	+	

В ICMP-сообщении об ошибке всегда возвращается IP-заголовок и первые 8 байтов IP-дейтаграммы, признанной ошибочной. Это позволяет ICMP-модулю сопоставить полученное им сообщение об ошибке с конкретным протоколом TCP или UDP (по значению поля протокол в возвращенном IP-заголовке) и с конкретным пользовательским процессом (по номеру порта, который находится в TCP или UDP-заголовке в возвращенных первых 8 байтах IP-дейтаграммы)

ICMP-сообщение об ошибке никогда не генерируется в ответ на следующие пакеты:

1. На ICMP-сообщение об ошибке.
2. На дейтаграмму, посланную по широковещательному IP-адресу или групповому IP-адресу.
3. На какой-либо фрагмент дейтаграммы, кроме первого ее фрагмента.

4. На дейтаграмму, в которой адрес источника не определяет конкретный хост. Это означает, что адрес источника не может быть нулевым адресом, адресом внутренней петли хоста (loopback) и не может быть широковещательным или групповым адресом.

Эти правила предназначены для предотвращения так называемых широковещательных штормов(broadcast storms).

Как работает Ping

Программа проверяет доступность зондируемого ею объекта в сети подобно локатору: посылает хосту ICMP-сообщение эхо-запрос (echo request) и ждет от него эхо-отклик (echo reply).

Эта утилита внешне действует подобно клиенту, а адресуемый хост, от которого приходит отклик, выступает в роли сервера. Однако на самом деле обработка эхо-запросов и генерация откликов осуществляется не каким-либо пользовательским процессом, а непосредственно ядром.

Формат ICMP сообщений, содержащих эхо-запрос или эхо-отклик, представлен на рис.5.7

0 7	8 15	16 31
Тип (0 или 8)	Код (0)	Контрольная сумма
Идентификатор		Порядковый номер
Необязательные данные		

Рис.5.7. Формат ICMP сообщений

Как и при обработке любых других ICMP-запросов, в ответном сообщении возвращаются поля идентификатор (identifier) и порядковый номер (sequence number) запроса. Также должны быть возвращены и содержащиеся в запросе необязательные данные(optional data), так как они используются источником запроса при интерпретации пришедшего на запрос ответа.

Unix-реализации программы Ping фиксируют в поле идентификатор

(identifier) ICMP-сообщения свой системный идентификатор процесса (process ID), пославшего сообщение. Это в последствии позволяет правильно распределять возвращенные ответы, когда на одном хосте параллельно работают несколько копий Ping.

В первом пакете, посылаемом Ping, значение поля порядковый номер (sequence number) устанавливается равным 0. Каждый раз при отправлении нового запроса это значение увеличивается на единицу.

Как работает Traceroute

Эта утилита даёт возможность отследить текущий маршрут движения IP – дейтаграмм от одного хоста к другому. Кроме того, программа позволяет использовать IP – опцию маршрутизации от источника (source route).

В основе Traceroute лежит идея отправки источником UDP-пакета адресату и постепенного изменения времени жизни (time-to-live, TTL). Первоначально TTL пакета равен 1, и когда пакет достигает первого маршрутизатора, его TTL сбрасывается (текущее значение поля TTL уменьшается на единицу), а маршрутизатор генерирует и отправляет в адрес источника ICMP-пакет со сведениями о превышении лимита времени. Тогда источник увеличивает на 1 начальное значение TTL, так что на сей раз UDP-пакет достигает следующего маршрутизатора, а тот тоже отсылает ICMP-пакет по превышению лимита времени. Совокупность этих ICMP-сообщений даёт список IP-адресов, пройденных на пути к конечной точке. Когда TTL увеличится настолько, что UDP-пакет достигнет искомой конечной точки, возвращается ICMP сообщение о недостижимости порта, поскольку на получателе ни один процесс не ждёт вашего сообщения.

5.4. Изучение программы NetInfo.

Для изучения назначения и возможностей утилит NetInfo загрузите эту программу по адресу.

Утилита Local Info

Утилита для предоставления сетевой информации про локальный хост и текущую версию сокетов Windows.

Вы можете использовать **Local Info** для:

- идентификации своего компьютера в сети
- определения того, какая локальная информация доступна

Для отображения информации:

1. Выберите вкладку **Local Info**. Результат будет выведен в область **Response**.
2. Нажатие кнопки **Refresh** обновит информацию.

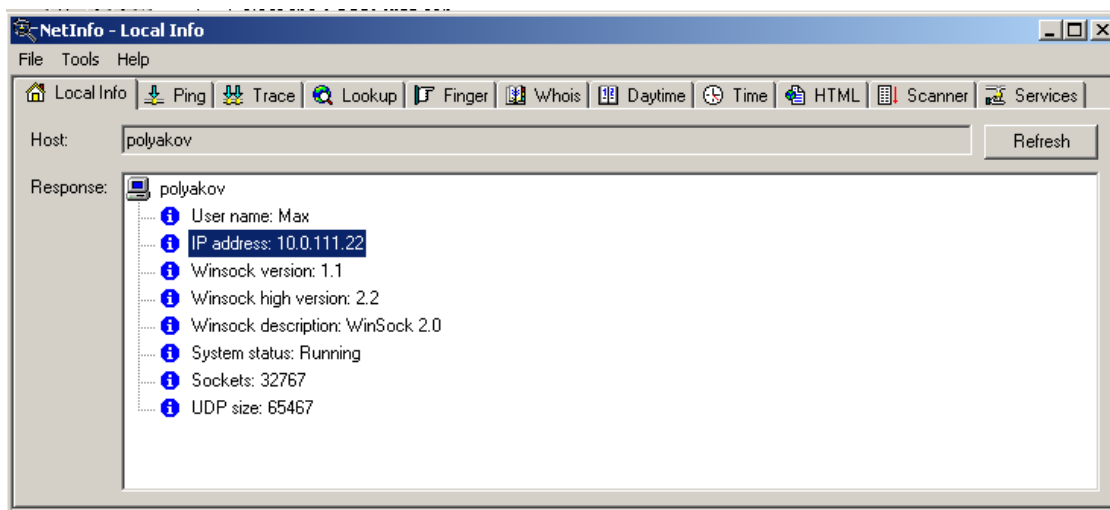


Рис.5.8. Пример использования Local Info

Утилита Ping (Packet Internet Groper)

Диагностическая утилита для проверки доступности удалённого хоста. Посылается ICMP эхо-запрос и ожидается ответ.

Вы можете использовать **Ping** для:

- «пинга» удалённого хоста для проверки сетевого соединения
- «пинга» удалённого хоста для определения скорости передачи данных в физической среде
- «пинга» хостов «по спирали» начиная с локального (127.0.0.1), затем всё более дальние компьютеры и шлюзы, для определения различных неисправностей.

Для «пинга» хоста:

1. Выберите вкладку **Ping**
2. В строке Host введите имя хоста или его IP адрес.
3. В опциях (вызываются нажатием правой клавиши мыши и выбором соответствующего пункта меню **Options**) устанавливаются следующие настройки:

- **Packets to send** - Количество посылаемых пакетов.
- **Timeout** - Сколько секунд будет ожидаться ответ от хоста.
- **Packet size** - Длина в байтах каждого пакета.

Нажмите кнопку Start. Утилита посылает эхо запрос и ждёт ответа. Если «пинг» успешен, то в области Response выводятся результаты.

Если Ping не получает ответа за время определённое в Timeout, выдаётся сообщение о неудаче. Существует несколько причин неудачи: удалённый хост может не функционировать, может не работать сеть или какой-либо шлюз или маршрутизатор на пути к удалённому хосту или же сервис Ping просто не поддерживается удалённым хостом.

Во время выполнения запроса кнопка Start изменяется на Stop, и Вы можете прекратить «пинг» в любой момент.

Примечание: Для использования Ping необходим статический IP адрес для Вашего компьютера. Также подойдёт IP адрес назначаемый DHCP сервером. Ping не будет работать в системах с эмуляцией IP адресов, таких, как, например, UNIX с запущенными TIA или Slirp. Ping не будет работать через брандмауэры, если только брандмауэр не

сконфигурирован для пропуска ICMP пакетов.

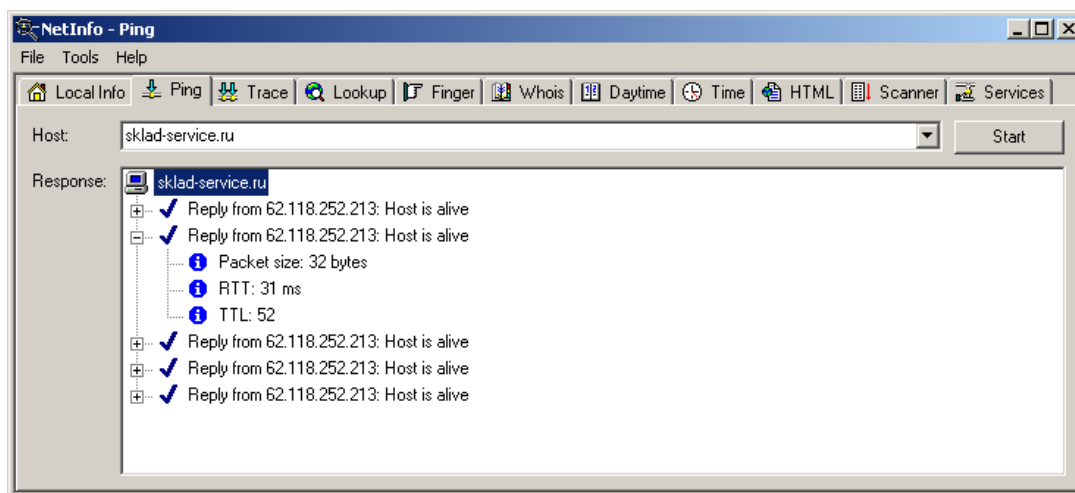


Рис. 5.9 Пример использования утилиты Ping

Утилита Trace (tracert)

Утилита, которая сообщает обо всех маршрутизаторах между компьютером, пославшим запрос, и удалённым хостом.

Tracert также показывает время ответа (в миллисекундах), которое показывает сколько нужно пакету времени для прохождения отрезка пути до определённого маршрутизатора и обратно. Это время зависит от загруженности сети.

Для инициации TraceRoute:

1. Выберите вкладку **Trace**.
2. В строке **Host** введите имя хоста или его IP адрес.
3. В опциях (вызываются нажатием правой клавиши мыши и выбором соответствующего пункта меню **Options**) устанавливаются следующие настройки:
 - **Timeout** - Сколько секунд **TraceRoute** будет пытаться найти путь к удалённому хосту.
 - **Packet Size** - Величина пакета в байтах.
 - **Number of Hops** - Число узлов до удалённого хоста (как правило за 30 «прыжков» можно достигнуть любого хоста)

Нажмите кнопку Start. Traceroute начинает поиск и выводит результаты в область Response.

Во время выполнения запроса кнопка Start изменяется на Stop, и вы можете прекратить поиск в любой момент.

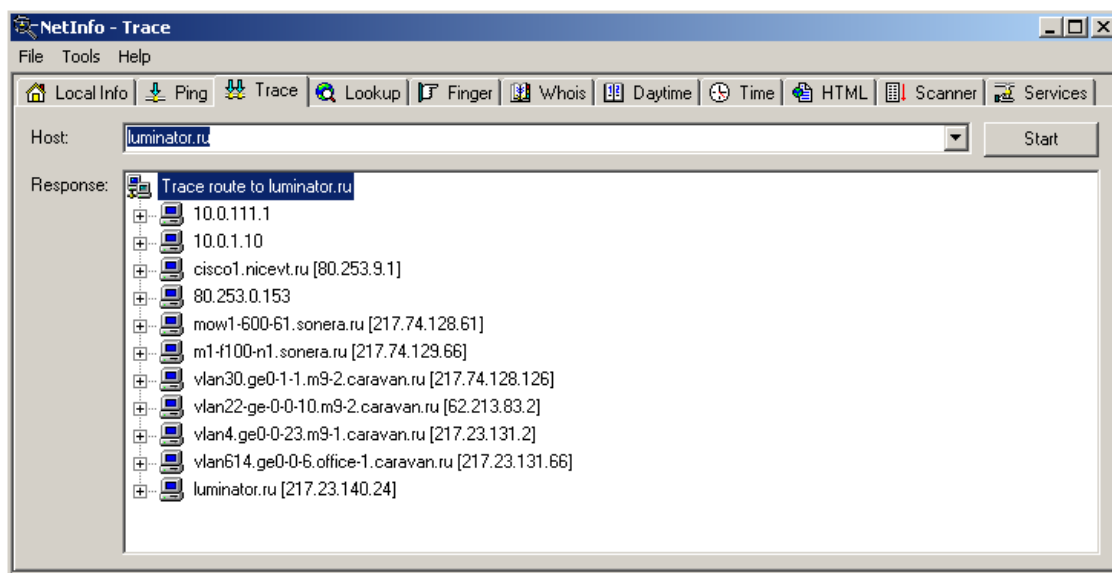


Рис. 5.10. Пример использования утилиты Trace

Утилита Lookup

Утилита возвращает официальное имя хоста, IP адрес и псевдоним (если он существует) из DNS.

Вы можете использовать утилиту Lookup для:

- Получения имени хоста из его IP адреса
- Получения IP адреса хоста из его имени.

Для инициализации запроса Lookup:

1. Выберите вкладку **Lookup**.
2. В строке **Host** введите имя хоста или его IP адрес.

Нажмите кнопку Start.

Lookup начинает поиск и выводит результаты в область Response.

Во время выполнения запроса кнопка Start изменяется на Stop, и Вы можете прекратить поиск в любой момент.

Примечание: Для Lookup необходима связь с сетью имеющей DNS

или WINS сервер или какой-нибудь другой сервер имён. Ваш компьютер должен быть сконфигурирован для доступа к DNS.

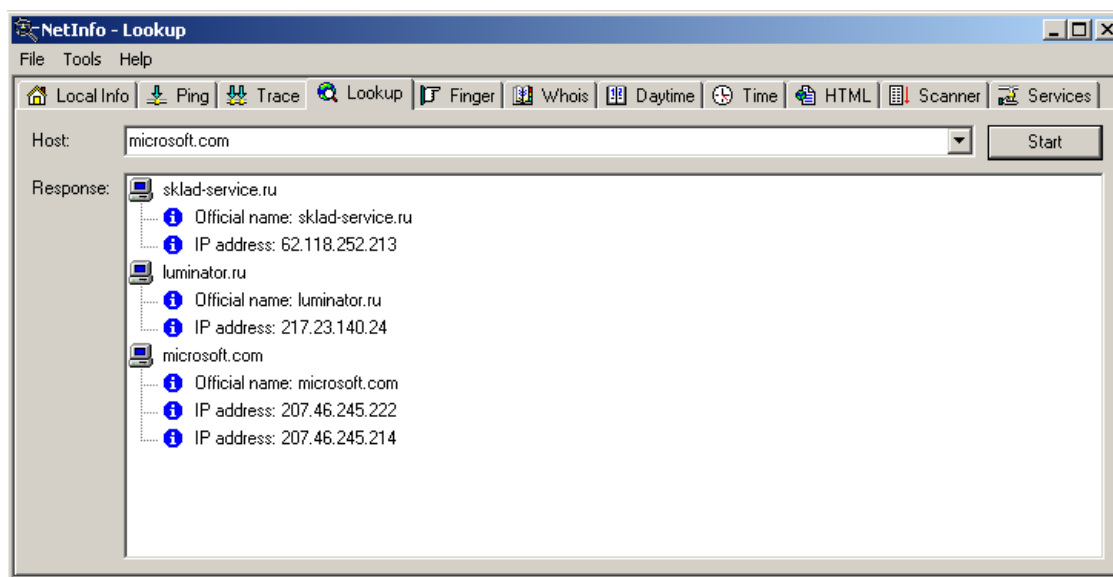


Рис.5.11. Пример использования утилиты Lookup

Утилита Finger

Утилита, позволяющая найти и отобразить информацию обо всех пользователях сетевого узла. Эта информация включает в себя список пользователей подключенных к хосту в данный момент (их идентификаторы и имена). Также для каждого пользователя указывается его корневая директория, время подключения, место нахождения офиса, когда они последний раз получали почту и когда они последний раз читали почту.

Запрос Finger также отображает всю информацию, содержащуюся в файлах .plan или .project в корневой директории. Эти файлы часто используются как простой путь для хранения информации. Например, сервер Finger на quake@geophys.washington.edu выдаёт систематизированную по датам информацию о землетрясениях, которые происходили в северо-западном регионе Соединённых Штатов.

Для инициализации запроса Finger:

1. Выберите вкладку Finger .

2. В строке Host введите имя хоста или его IP адрес.
3. Нажмите кнопку Start.

Клиент Finger связывается с сервером Finger. Результаты запроса отображаются в области Response. Если на удалённом хосте нет сервера Finger, то клиент выводит соответствующее сообщение — No server found there.

Примечание: Обычно только UNIX или NT хосты поддерживают сервер Finger. Многие системные администраторы выключают серверы Finger, так как они представляют потенциальный риск.

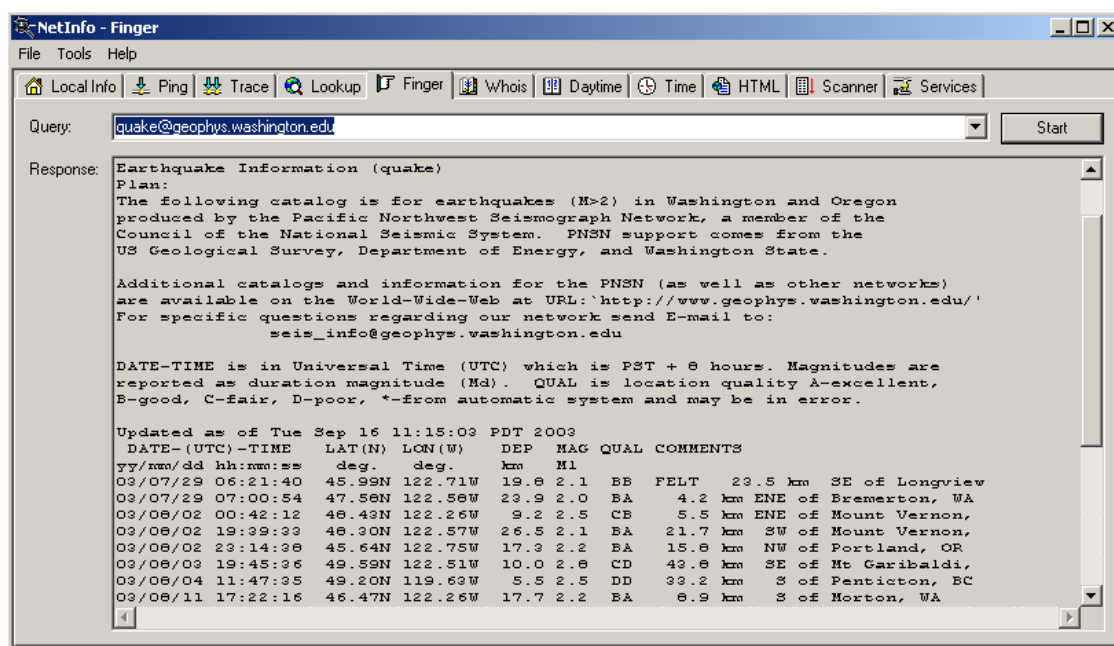


Рис.5.12. Пример использования утилиты Finger

Утилита Whois

Сетевая информационная утилита, которая предоставляет информацию о том, кто владеет Интернет хостом или доменом, и с кем вы можете проконтактировать относительно этого хоста или домена. Whois запрос показывает фамилию контактного лица, адрес его электронной почты, номер телефона, и сетевой почтовый ящик для всех пользователей и организаций, которые зарегистрированы на одном из

официальных Whois серверов, таких как база данных Internet Network Information Center (interNIC).

Для инициализации запроса Whois:

1. Выберите вкладку **Whois**.
2. В текстовом поле **Query**, введите текст запроса, например microsoft.com.

Введите имя или “указатель” (уникальный идентификатор, который соответствует Whois записи) человека или организации. Выпадающий список показывает запросы, которые были введены ранее. Запрос ищет все записи в базе данных Whois до точного совпадения имени или “указателя”.

Если вы не знаете имени или указателя, вы можете ввести строку запроса частично, поставив в конце одну или несколько точек. Например, введя “Mack.” вы найдете “Mack,” “Mackall,” “Maskey.”

Если получено более, чем одно местонахождение имени, Whois возвращает краткое описание каждого. Вы можете затем взять указатель имени (показанный в скобках), о котором вы хотите получить больше информации и ввести его как строку запроса, с восклицательным знаком впереди, например “!ABC.”

3. В диалоговом окне **Options** находится набор опций, которые вы можете использовать.

NetInfo позволяет автоматически определять Whois сервер, основываясь на стране, взятой из строки запроса. Вы можете убрать флажок **Autodetect** для того, чтобы запретить вашим запросам обращаться к серверам, указанным в установках Whois :

whois.internic.net - Предоставляет информацию о пользователях и организациях, зарегистрированных в Internet Network Information Center (interNIC).

whois.arin.net - Предоставляет информацию о пользователях и

организациях, зарегистрированных на сервере American Registry for Internet Numbers.

whois.nic.gov - Предоставляет информацию о пользователях и организациях, зарегистрированных в сети U.S. government.

whois.nic.mil - Предоставляет информацию о пользователях и организациях, зарегистрированных в сети U.S. military.

whois.ripe.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере European IP Address Allocations.

whois.nic.uk - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере United Kingdom IP Address Allocations.

whois.ripn.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере Russian IP Address Allocations.

whois.apnic.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере Asia Pacific IP Address Allocations.

whois.aunic.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере Australia IP Address Allocations.

Примечание: Вы можете выбирать из текущего списка известных Whois серверов и добавлять новые серверы к списку.

4. Нажмите кнопку **Start**.

Whois клиент соединится с указанным Whois сервером. Результат запроса появится в области **Response** . Если запрос найдет единственное нахождение строки поиска (человека или организации), он показывает детальную информацию для этого человека или организации. Если он находит больше, чем одно вхождение строки поиска, он показывает краткую информацию о каждой записи, которая совпадает с условием поиска.

Примечание: Чтобы показать экран помощи по использованию сервисов, которые предоставляются NIC через Whois, пошлите Whois запрос с текстом “help”.

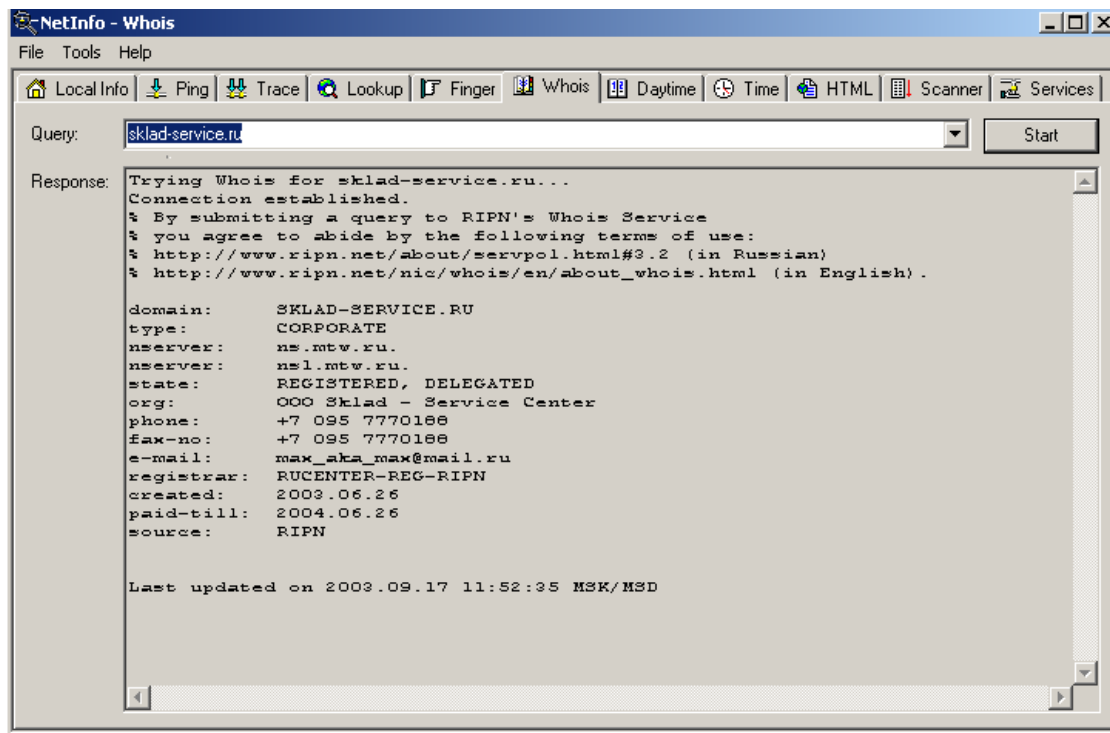


Рис.5.13. Пример использования утилиты Whois

Утилита Daytime

Сетевая информационная утилита, которая принимает локальное время с другого компьютера. Множество веб-серверов и DNS-серверов отвечают на такой запрос.

Чтобы получить локальное время, сделайте следующее:

1. Выберите вкладку **Daytime**.
2. В поле **Server**, введите имя хоста или ip адрес удаленного daytime сервера (например, www.mit.edu).

Выпадающий список показывает ранее введенные имена или ip адреса.

3. Нажмите на кнопку **Start**.

Daytime клиент соединяется с daytime сервером. Результат запроса показывается в области **Response**.

При попытке соединиться с сервером, который не поддерживает этот сервис, результатом будет сообщение об отказе соединения.

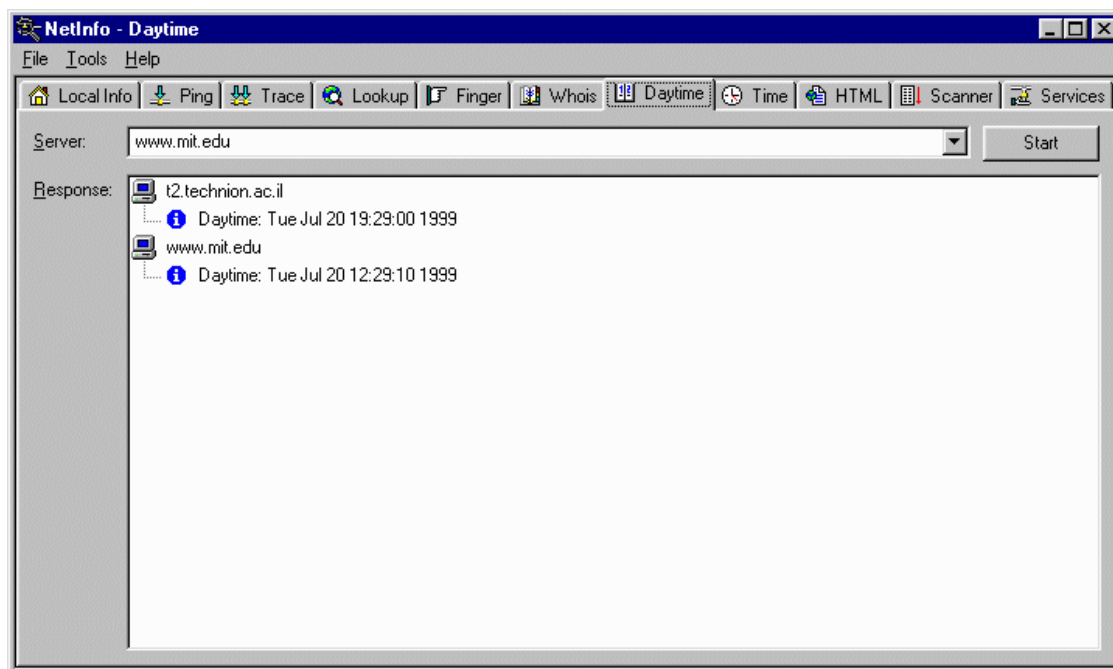


Рис.5.14. Пример использования утилиты Daytime

Утилита Time

Сетевая информационная утилита, которая получает точное значение времени с сервера времени. Эта утилита позволяет вам синхронизировать ваши локальные системные часы с часами на удаленном сервере.

Для синхронизации сделайте следующее:

1. Выберите вкладку **Time**.
2. В текстовом поле **Server**, введите имя или IP адрес сервера времени (например, www.mit.edu).

Выпадающий список показывает ранее введенные имена или ip адреса.

3. Нажмите кнопку **Start**.

Утилита Time устанавливает соединение с удаленным сервером и показывает имя сервера, текущее время, полученное с сервера, а также разницу во времени между часами сервера и вашими.

4. Нажмите правой кнопкой на сервере времени в области **Response**, чтобы показать выпадающее меню, а затем выберите **Synchronize**.

При успешном изменении, утилита Time показывает сообщение, которое говорит о том, что ваши часы были обновлены.

Примечание: Временной протокол имеет разрешение в одну секунду и не дает представления о времени прохождения пакета.

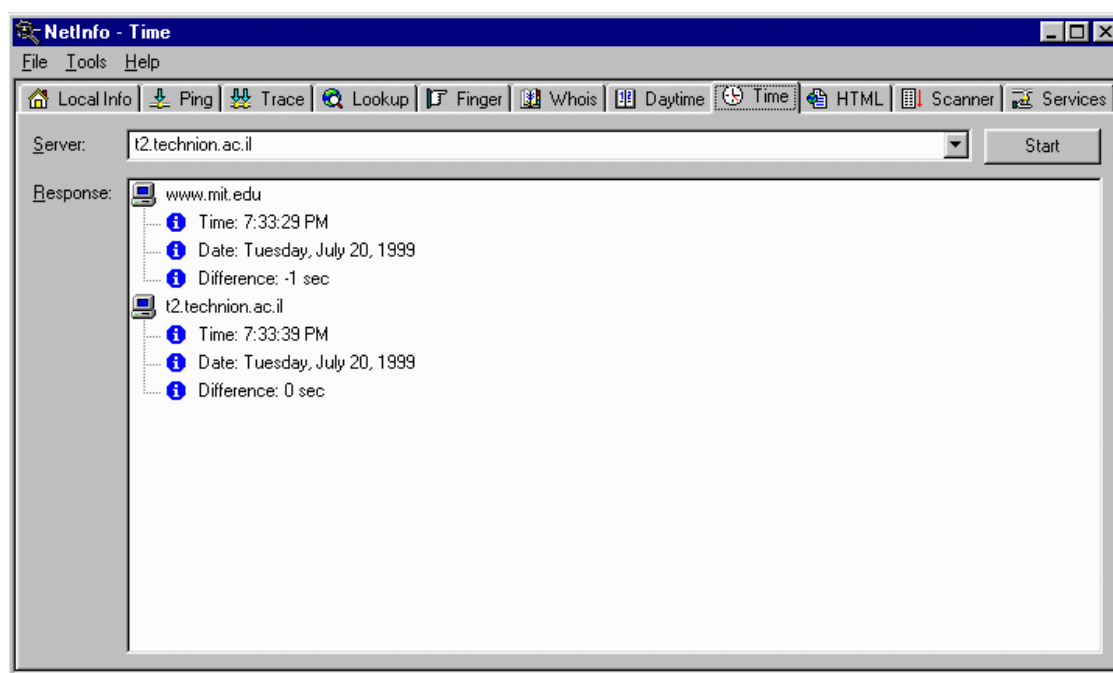


Рис.5.15. Пример использования утилиты Time:

Утилита HTML (HyperText Markup Language) – Гипертекстовый язык разметки

Сетевая информационная утилита, которая посылает запрос на указанный веб адрес ([URL](#)) и возвращает полную заголовочную информацию (включая [cookies](#)), а также возвращает информацию со страницы в форматированном HTML коде.

Вы можете использовать утилиту HTML для отладки вашего сайта.

Для запуска утилиты сделайте следующее:

1. Выберите вкладку **HTML**.
2. В поле **URL** введите адрес веб страницы, которую вы хотите запросить. Он должен указывать файл с веб сайта (например: <http://hostname/page> или hostname/page). Выпадающий список показывает адреса, введенные ранее.

3. В диалоговом окне **Options** находится набор опций:

Get from the wire - Получить данные из провода, даже если они были локально кэшированы.

Do not cache - Не кэшировать данные.

Full header information - Показывать полную заголовочную информацию.

Page data - Показывать данные со страницы.

Нажмите кнопку Start.

Результаты запроса появятся в области Response. Если указанный хост не имеет веб сервера, утилита HTML покажет сообщение: No server found there.

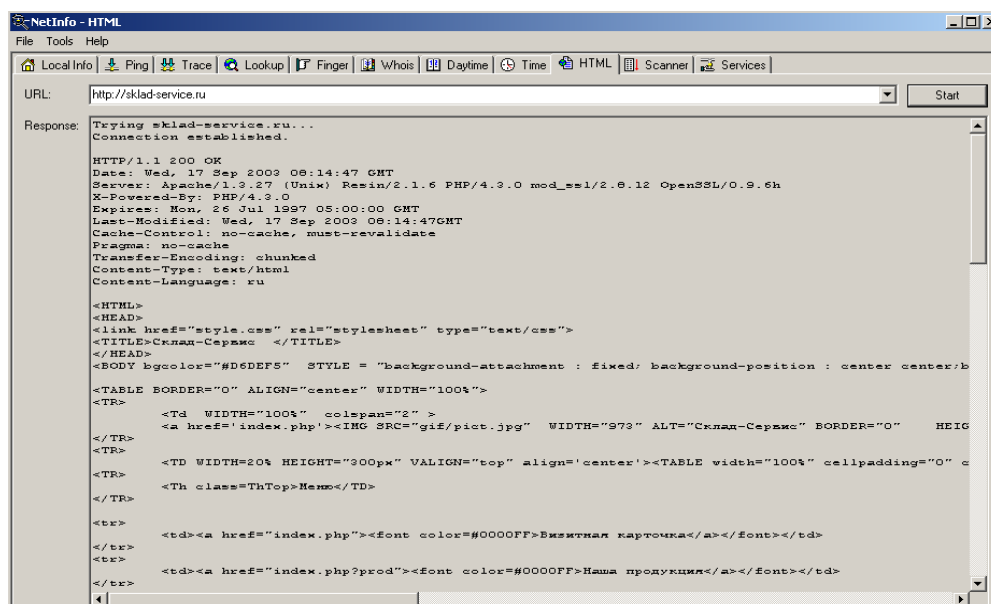


Рис.5.16. Пример использования утилиты HTML

Утилита **Scanner**

Сетевая информационная утилита, которая сканирует все хосты в указанном диапазоне IP адресов и проверяет статус хостов.

Чтобы запустить Scanner, сделайте следующее:

1. Выберите вкладку **Scanner**.
2. В поле **Address**, введите IP адрес для сканирования (например, 192.41.61.50).
3. В диалоговом окне **Options** находится набор опций:
Ascending - Когда эта опция включена, Scanner сканирует все имена хостов в в порядке возрастания IP адреса.
Enabled - Когда эта опция включена, Scanner проверяет статус хоста для каждого IP адреса.
Timeout - Количество секунд, когда Scanner проверяет хост, который не отвечает.
Retries - Количество попыток проверки не отвечающего хоста.
4. Нажмите кнопку **Start**.

Утилита Scanner сканирует диапазон IP адресов. Результат сканирования появляется в области **Hosts**.

В течение сканирования кнопка **Start** превращается в **Stop**. Вы можете нажать **Stop** в любое время, чтобы остановить процесс сканирования.

Замечание:

Проверка статуса хостов может значительно увеличить время, требующееся для завершения сканирования.

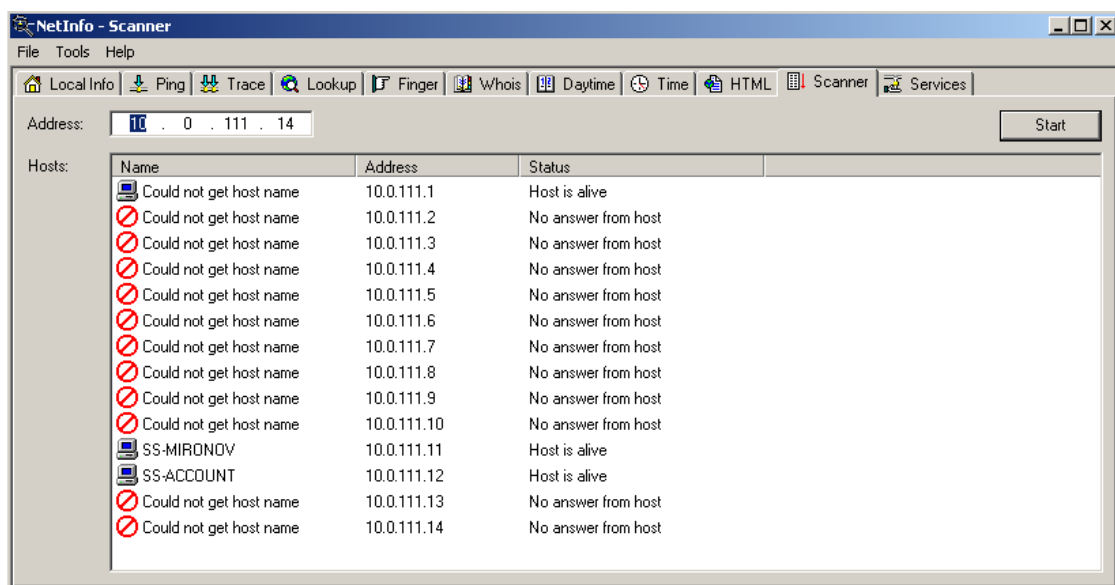


Рис.5.17. Пример использования утилиты Scanner.

Утилита Services

Сетевая диагностическая утилита, которая проверяет статус сервисов хоста.

Чтобы запустить Services, сделайте следующее:

1. Выберите вкладку **Services**.
2. В текстовом поле **Host** введите имя или IP адрес удаленного (например, www.mit.edu).
3. В диалоговом окне **Options** находится набор сервисов, которые вы хотите проверить..
4. Нажмите кнопку **Check**.

Утилита Services проверяет статус сервисов хоста. Результат сканирования появляется в области **Response**.

В течение проверки, кнопка **Check** превращается в **Stop**. Вы можете нажать **Stop** в любое время для завершения проверки.

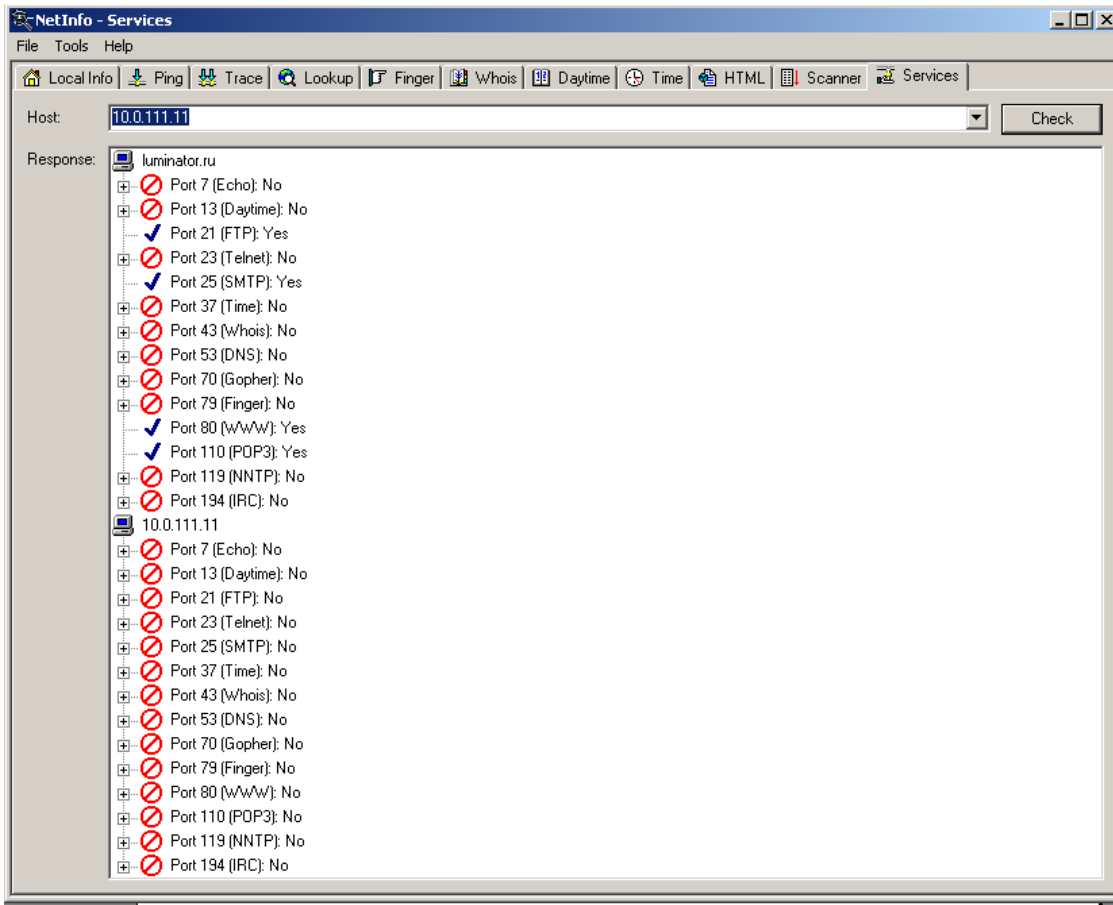


Рис.5.18. Пример использования утилиты Services:

Утилита IPMonitor

Утилита системного трее включена как часть NetInfo для наблюдения за включением / выключением компьютеров. IPMonitor проверяет сетевую доступность списка хостов, определенных пользователем и предупреждает вас об ошибках, используя аудио сигналы и оповещение с помощью иконок.

Чтобы создать список хостов, сделайте следующее:

1. Запустите **IPMonitor**.
2. Нажмите правой кнопкой на иконку IPMonitor в системном трее и в выпадающем меню выберите **Details**.
3. Нажмите кнопку **Add**. В текстовом поле **Host**, введите имя хоста или его IP адрес (например, www.netscape.com). В поле **Description** введите описание для этого хоста (например, Netscape

Netcenter). Также вы можете использовать следующие опции:

Interval - Количество минут, которое IPMonitor ждет до следующей проверки.

Timeout - Количество секунд, когда Scanner проверяет хост, который не отвечает.

Retries - Количество попыток проверки не отвечающего хоста.

4. Нажмите кнопку **OK**.

IPMonitor сразу начинает проверять хосты. Результаты проверки появляются в области **Details**.

5. Повторите пункты 3 и 4, чтобы еще добавить хосты к списку.

6. Нажмите кнопку **Close**.

Примечание: Вы можете два раза кликнуть по иконке IPMonitor в трее, чтобы появилось диалоговое окно Details. Вы можете нажать правой кнопкой в поле Details, чтобы появилось выпадающее меню.

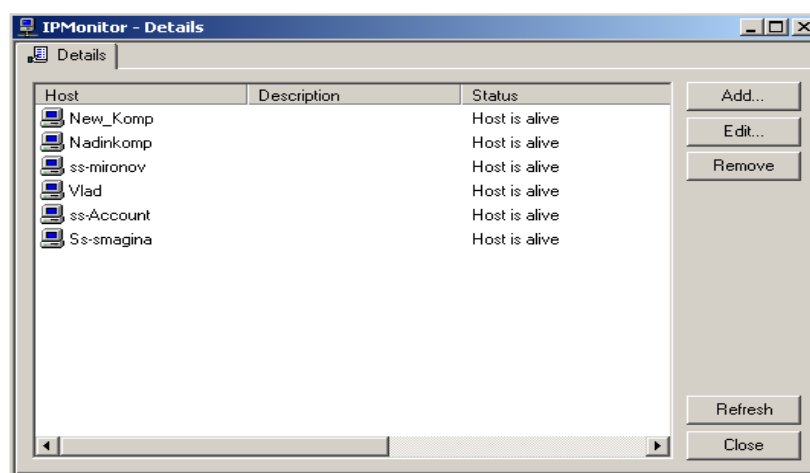


Рис.5.19. Пример использования утилиты IPMonitor:

5.5. Изучение пакетного анализатора Wireshark.

Пакетный анализатор **Wireshark** предназначен для отслеживания трафика, проходящего через сетевой интерфейс узла. Программа предоставляет возможность анализа широкого набора сетевых протоколов и позволяет динамически отображать на экране информацию о процессе поступления и отсылки пакетов. Помимо этих

возможностей программа предоставляет различные средства статистического анализа просмотренного трафика.

Основное окно программы

На рис.5.20 представлено основное окно программы. Оно разделено на несколько функциональных областей.

Область 1 содержит набор кнопок, управляющих работой программы. Наибольшее значение имеет первая группа кнопок.

Interfaces - открывает список доступных для анализа сетевых интерфейсов. В этом окне можно выбрать интерфейс, который будет просматриваться программой.

Options – открывает окно настройки анализа (см. далее).

Start – запуск анализа.

Stop – останов.

Restart – очистка результатов и перезапуск анализа.

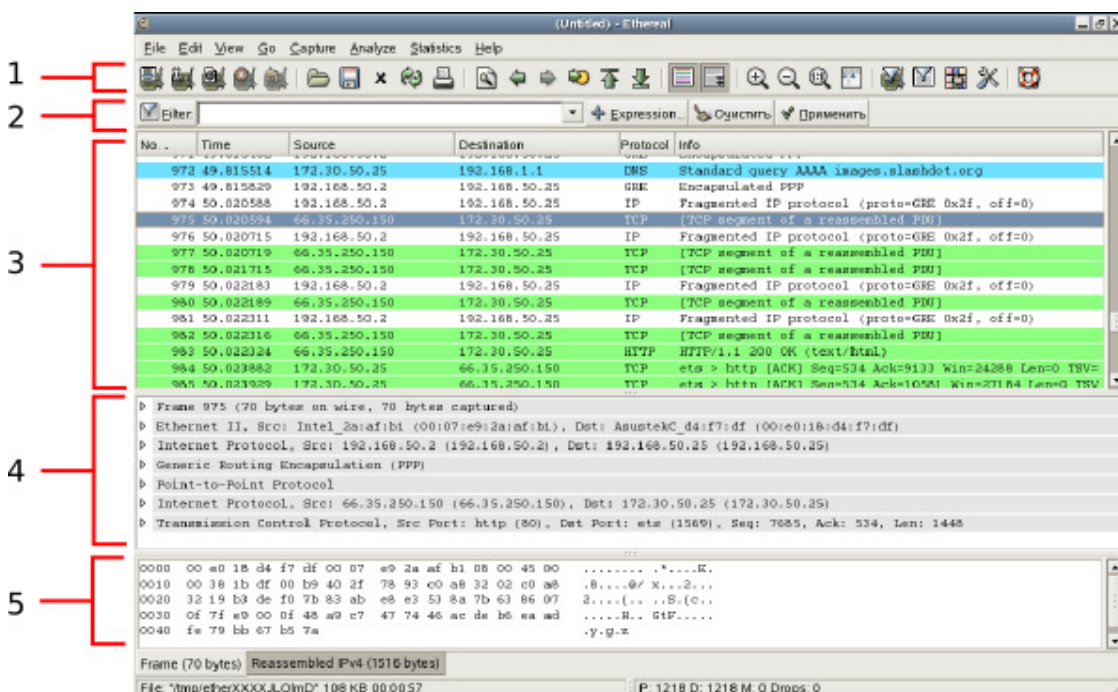


Рис.5.20. Главное окно Wireshark.

Область 2 – строка настройки фильтра. Wireshark позволяет выделять из потока только пакеты интересующих пользователя протоколов. Для обозначения правил фильтрации используется развитый язык выражений, позволяющий выбирать необходимые протоколы, вводить ограничения на содержимое пакетов, комбинировать условия с помощью логических операций и пр. (см. далее).

В указанной области находится кнопка Filters, открывающая окно выбора одного из предопределённых фильтров и создания собственных фильтров. Далее следует строка ввода, позволяющая быстро применить новое условие фильтрации к текущему анализу. Кнопка Expression открывает окно конструктора выражений фильтрации. Здесь представлен широкий набор возможных выражений и операций над ними. Кнопки Apply и Clear служат для активации и очистки выражения в строке ввода.

В область 3 выводится основная информация. Здесь отображается список отслеженных пакетов, прошедших через сетевой интерфейс. Для каждого пакета указывается информация о времени прохождения, адресах источника и приёмника, протоколе и краткое описание его содержимого.

При выделении в списке одного из пакетов в области 4 отображается информация о его содержимом. Она структурирована по всем протоколам, инкапсулирующим передаваемые данные. В раскрывающихся списках содержатся названия и значения полей заголовков протоколов. Выделение одного из полей в таком списке сопровождается подсветкой соответствующей информации в области 5. Здесь отображается необработанное содержимое пакета в шестнадцатеричном виде.

Настройка параметров анализа

Рассмотрим процесс настройки работы анализатора (рис.21).

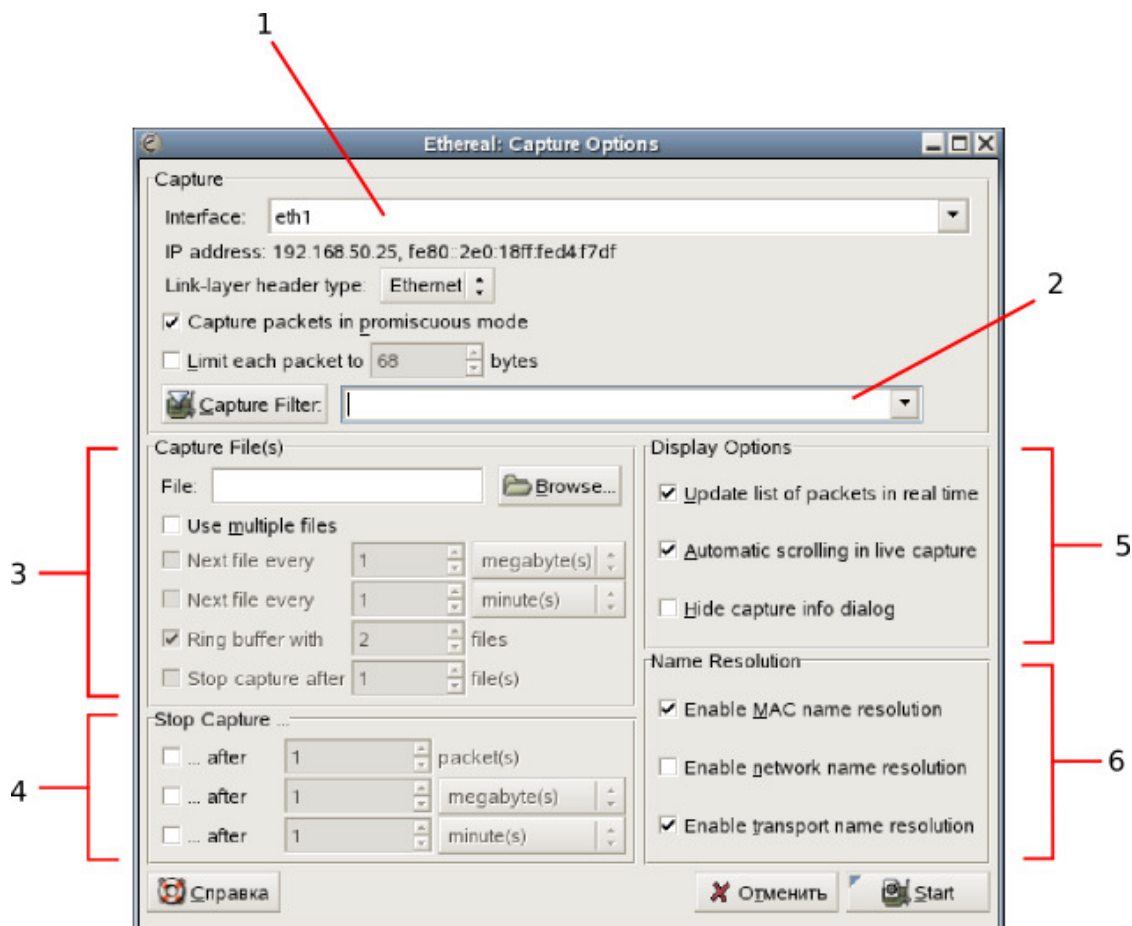


Рис. 5.21. Окно настройки параметров анализа.

Поле 1 позволяет выбрать просматриваемый интерфейс. В поле 2 можно задать необходимый фильтр (это можно сделать и позже, в процессе анализа). В области 3 можно задать имя файла для сохранения результатов анализа и настроить параметры автоматического сохранения. Область 4 позволяет задать условия прекращения работы программы. Область 5 содержит следующие опции:

Update list of packets in real time – включить динамическое обновление списка просмотренных пакетов;

Automatic scrolling in live capture – автоматическая прокрутка списка пакетов во время работы анализатора;

Hide capture info dialog – скрывать окно, содержащее статистику по количеству просмотренных пакетов.

Область 6 позволяет настроить преобразование имён узлов в символическое представление.

В нижней части окна располагается кнопка **Start**, запускающая анализатор.

Фильтрация пакетов

Wireshark обладает развитым механизмом задания параметров фильтрации. Требуемые параметры задаются с помощью строки, содержащей выражение особой формы. Выражение состоит из одного или более условий, связанных логическими операциями. В качестве условий могут выступать названия протоколов и ограничения на значения отдельных полей пакетов. Использование названия протокола означает разрешение на просмотр всех пакетов данного протокола, в том числе и инкапсулирующих данные других протоколов. Для задания ограничений на отдельное поле пакета требуется указать его имя в форме **<имя_протокола>.<имя_поля>** и определить отношение его значения к определённому значению с помощью знаков **>**, **<**, **>=**, **<=**, **==**, **!=**, **contains**, **matches** **present**.

Условия комбинируются с помощью логических операторов **and** и **or**. Употребление ключевого слова **not** перед условием позволяет инвертировать его значение. Допускается использование скобок.

Пример: **tcp and (not http) and (ip.ttl >= 10)**

(все пакеты TCP, не содержащие пакетов HTTP и имеющие значения поля TTL заголовка IP не менее 10).

Граф анализа

Программа предоставляет пользователю набор средств автоматизированного анализа полученной информации, среди которых инструмент **Flow Graph** (меню **Statistics**). Он позволяет построить диаграмму перемещения пакетов по узлам.. На рис.22 представлен

пример такой диаграммы, полученный по результатам анализа работы утилиты **traceroute**.

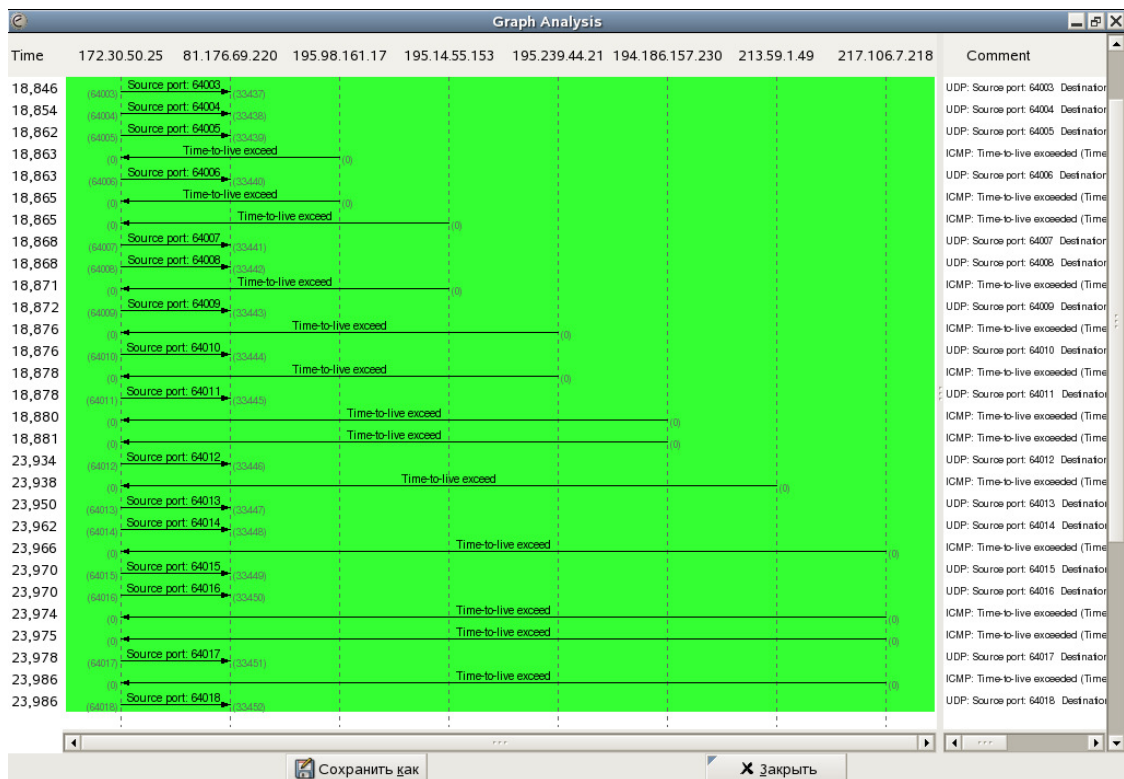


Рис. 5.22. Граф анализа работы программы *traceroute*.

Порядок работы с приложением.

Обычно для выполнения анализа трафика следует выполнить следующие действия.

1. Открыть окно настройки параметров анализа;
2. Выбрать требуемый сетевой интерфейс из списка обнаруженных;
3. Ввести необходимые условия фильтрации.
4. Если требуется, выбрать режим автоматического сохранения информации, условия останова работы анализатора;
5. Установить флаги Update list of packets in real time и Automatic scrolling in live capture;
6. Нажать кнопку Start.
7. Отслеживать обрабатываемый трафик в основном окне.

8. При необходимости остановить анализ, сменить параметры фильтрации и продолжить работу.
9. Остановить анализ, когда вся необходимая информация представлена на экране.
10. Анализировать информацию, просматривая содержимое пакетов и используя встроенные средства анализа.

5.6. Порядок выполнения работы.

Лабораторная работа №8. Исследование протоколов сетевого уровня IP-сетей.

В соответствии с вариантом либо послать ICMP эхо-запрос на удалённый хост либо определить количество «хопов» до удалённого хоста. При помощи пакетного анализатора проанализировать все пакеты приходящие на сетевой интерфейс. Для нечётных вариантов просматривать ICMP трафик, а для чётных ICMP и UDP трафик. Результаты анализа представить в отчете.

Просканировать порты удалённого хоста и зафиксировать и описать службы на открытых портах.

Сделать запросы, соответствующие варианту. Объяснить реакцию хоста на каждый запрос.

Отчет по лабораторной работе должен содержать сценарии выполнения вышеуказанных процедур, результаты анализа пакетов, описание служб на открытых портах удалённого хоста, описание реакции на запросы.

Ответить на контрольные вопросы.

Защитить лабораторную работу.

Лабораторная работа №9. Исследование протоколов транспортного уровня IP-сетей.

Инициировать TCP сеанс с хостом, указанным в варианте, и проанализировать его открытие, поддержку и закрытие при помощи пакетного анализатора. Перенести в отчет описание процедур открытия, поддержания и закрытия TCP соединения посредством обмена заголовками протокола TCP для этого сеанса. Найти и перенести в отчет все пакеты, отвечающие за вышеуказанные события.

Ответить на контрольные вопросы.

Защитить лабораторную работу.

5.7. Варианты заданий.

ИУ5-51

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	президент.рф	+		+	who-is, HTML
2	правительство.рф		+	+	who-is, HTML
3	яндекс.рф	+		+	lookup, HTML
4	гугл.рф		+	+	who-is, lookup
5	рамблер.рф	+		+	who-is. HTML
6	известия.рф		+	+	lookup, HTML
7	кремль.рф	+		+	who-is, lookup
8	лента.рф		+	+	who-is. HTML
9	почта.рф	+		+	who-is, lookup
10	жж.рф		+	+	lookup, HTML
11	вконтакте.рф	+		+	who-is. HTML
12	хабрахабр.рф		+	+	who-is, lookup
13	ульяновск.рф	+		+	lookup, HTML
14	псков.рф		+	+	who-is, lookup
15	ростов-на-дону.рф	+		+	who-is, lookup
16	салехард.рф		+	+	who-is. HTML
17	самара.рф	+		+	who-is, lookup
18	внуково.рф		+	+	who-is, HTML
19	газпром.рф	+		+	lookup, HTML
20	тнк.рф		+	+	who-is, lookup

21	ржд.рф	+		+	who-is. HTML
22	лדпр.рф		+	+	lookup, HTML
23	минфин.рф	+		+	who-is, lookup
24	мчс.рф		+	+	who-is, HTML
25	фсб.рф	+		+	lookup, HTML

ИУ5-52

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	фонд-культуры-екатерина.рф	+		+	who-is, HTML
2	консультант-плюс.рф		+	+	who-is, HTML
3	лаборатория-касперского.рф	+		+	lookup, HTML
4	платиновая-унция.рф		+	+	who-is, lookup
5	фармацевтический-вестник.рф	+		+	who-is. HTML
6	уралсиб-банк.рф		+	+	lookup, HTML
7	а-айсберг.рф	+		+	who-is, lookup
8	антивирус-касперского.рф		+	+	who-is. HTML
9	найдется-все.рф	+		+	who-is, lookup
10	вим-авиа.рф		+	+	lookup, HTML
11	мой-круг.рф	+		+	who-is. HTML
12	спорт-экспресс.рф		+	+	who-is, lookup
13	авангардбанк.рф	+		+	lookup, HTML
14	жж.рф		+	+	who-is, lookup
15	балтийскийбанк.рф	+		+	who-is, lookup
16	автолайн.рф		+	+	who-is. HTML
17	инвестбанк.рф	+		+	who-is, lookup
18	билайн.рф		+	+	lookup, HTML
19	мегафон.рф	+		+	who-is. HTML
20	антигриппин.рф		+	+	who-is, lookup
21	айпод.рф	+		+	lookup, HTML
22	ростелеком.рф		+	+	lookup, HTML
23	есть-домены.рф	+		+	who-is, lookup
24	ру-центр.рф		+	+	who-is, HTML
25	альфа-банк.рф	+		+	lookup, HTML

ИУ5-53

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	тамбов.рф	+		+	who-is, HTML
2	тверь.рф		+	+	who-is, HTML
3	улан-удэ.рф	+		+	lookup, HTML
4	липецк.рф		+	+	who-is, lookup
5	майкоп.рф	+		+	who-is. HTML
6	москва.рф		+	+	lookup, HTML
7	мурманск.рф	+		+	who-is, lookup
8	нижнийновгород.рф		+	+	who-is. HTML
9	новосибирск.рф	+		+	who-is, lookup
10	омск.рф		+	+	lookup, HTML
11	пенза.рф	+		+	who-is. HTML
12	петрозаводск.рф		+	+	who-is, lookup
13	горно-алтайск.рф	+		+	lookup, HTML
14	екатеринбург.рф		+	+	who-is, lookup
15	ижевск.рф	+		+	who-is, lookup
16	казань.рф		+	+	who-is. HTML
17	курган.рф	+		+	who-is, lookup
18	курск.рф		+	+	lookup, HTML
19	абакан.рф	+		+	who-is. HTML
20	великийновгород.рф		+	+	who-is, lookup
21	владивосток.рф	+		+	lookup, HTML
22	кинопоиск.рф		+	+	time, whois
23	гарант-парк-телеком.рф	+		+	who-is, lookup
24	шереметьево.рф		+	+	who-is. HTML
25	невский-акционерное-общество.рф	+		+	who-is, lookup

ИУ5-54

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	натурпродукт.рф	+		+	who-is, HTML
2	народ.рф		+	+	who-is, HTML
3	технопарк.рф	+		+	lookup, HTML
4	башинформ.рф		+	+	who-is, lookup
5	марий-эл.рф	+		+	who-is. HTML

6	владимирскаяобласть .рф		+	+	lookup, HTML
7	псковскаяобласть.рф	+		+	who-is, lookup
8	югра.рф		+	+	who-is. HTML
9	чувашия.рф	+		+	who-is, lookup
10	алтайскийкрай.рф		+	+	lookup, HTML
11	курганскаяобласть.рф	+		+	who-is. HTML
12	владимирскаяобласть .рф		+	+	who-is, lookup
13	татарстан.рф	+		+	lookup, HTML
14	центральныйбанк.рф		+	+	who-is, lookup
15	пенсионныйфонд.рф	+		+	who-is, lookup
16	росграница.рф		+	+	who-is. HTML
17	фас.рф	+		+	who-is, lookup
18	росавтодор.рф		+	+	lookup, HTML
19	минприроды.рф	+		+	who-is. HTML
20	ропатент.рф		+	+	who-is, lookup
21	вольво.рф	+		+	who-is. HTML
22	ведомости.рф		+	+	lookup, HTML
23	эксперт.рф	+		+	who-is, lookup
24	россия.рф		+	+	who-is, HTML
25	прокуратура.рф	+		+	lookup, HTML

5.8. Контрольные вопросы.

1. Какое назначение имеет протокол IP?
2. Как работает утилита ping?
3. При помощи какой утилиты можно получить информацию обо всех пользователях сетевого узла?
4. По каким флагам заголовка протокола TCP можно идентифицировать фазу TCP соединения?
5. Как осуществляется настройка фильтрации пакетов в пакетном анализаторе?
6. Как работает утилита traceroute?
7. Сколько фаз проходит TCP соединение?

6. Литература.

1. Галкин В.А., Григорьев Ю.А. Телекоммуникации и сети: Учеб. пособие для вузов. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2003.
2. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы 3-е издание. Учебное пособие, СПб.:Питер 2007
3. Степанов А.Н. Архитектура вычислительных систем и компьютерных сетей, СПб.: Питер 2007
4. Л. Клейрок Теория массового обслуживания. Перевод с англ. /Пер. И. И. Грушко; ред. В. И. Нейман – М.: Машиностроение, 1979. – 432с., ил.