

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

А. В. Королькова, Д. С. Кулябов

Сетевые технологии

Лабораторные работы

Учебное пособие

Москва

Российский университет дружбы народов

2014

УДК

004.72+004.057.4(075.8)

ББК 32.973.202я73

К 68

Утверждено

РИС Ученого совета

Российского университета

дружбы народов

Рецензенты:

старший научный сотрудник Института проблем информатики РАН

кандидат физико-математических наук *Р. В. Разумчик*;

начальник сектора телекоммуникаций УИТО РУДН

кандидат физико-математических наук, доцент *К. П. Ловецкий*

Королькова, А. В.

К 68 Сетевые технологии. Лабораторные работы : учебное пособие /

А. В. Королькова, Д. С. Кулябов. — Москва : РУДН, 2014. —

106 с. : ил.

ISBN 978-5-209-05606-5

Пособие представляет собой лабораторный практикум по дисциплине «Сетевые технологии» и предназначено для студентов направлений «Математика и компьютерные науки», «Фундаментальная информатика и информационные технологии», «Бизнес-информатика».

УДК 004.72+004.057.4

ББК 32.973.202

ISBN 978-5-209-05606-5

© Королькова А. В., Кулябов Д. С., 2014

© Российский университет дружбы
народов, Издательство, 2014

Содержание

Лабораторный практикум по дисциплине «Сетевые технологии»	5
Глава 1. Методы кодирования и модуляция сигналов	7
1.1. Цели работы	7
1.2. Теоретические сведения	7
1.3. Высокоуровневый интерпретируемый язык программирования Octave	10
1.4. Порядок выполнения работы	16
Глава 2. Расчёт сети Fast Ethernet.	31
2.1. Цели работы	31
2.2. Теоретические сведения	31
2.3. Задание для выполнения.	36
Глава 3. Знакомство с Packet Tracer. Моделирование простой сети.	38
3.1. Цели работы	38
3.2. Предварительные сведения. Запуск, настройка, краткое описание интерфейса Packet Tracer	38
3.3. Моделирование простейшей сети. Протокол ICMP	41
3.4. Конфигурирование оборудования Cisco	48
Глава 4. Packet Tracer. Настройка маршрутизаторов.	56
4.1. Цели работы	56
4.2. Предварительные сведения	56
4.3. Моделирование сети со статической маршрутизацией	59
4.4. Задание для самостоятельной работы	67
4.5. Моделирование сети с динамической маршрутизацией	68
4.6. Задание для самостоятельной работы	72
Литература	74

Учебно-методический комплекс дисциплины «Сетевые технологии»	75
Программа дисциплины.	77
Цели и задачи дисциплины	77
Место дисциплины в структуре ООП	77
Требования к результатам освоения дисциплины.	78
Объем дисциплины и виды учебной работы.	79

Содержание дисциплины	80
Лабораторный практикум	83
Практические занятия (семинары)	83
Примерная тематика курсовых проектов (работ).	83
Учебно-методическое и информационное обеспечение дисциплины	83
Материально-техническое обеспечение дисциплины.	84
Методические рекомендации по организации изучения дисциплины	84
Фонды оценочных средств.	86
Примерные тестовые задания	86
Перечень тем для контроля знаний	98
Календарный план	99
Балльно-рейтинговая система	102
Сведения об авторах	106

Лабораторный практикум по дисциплине

«Сетевые технологии»

Глава 1. Методы кодирования и модуляция сигналов

1.1. Цели работы

Изучение методов кодирования и модуляции сигналов с помощью высокоуровневого языка программирования Octave. Определение спектра и параметров сигнала. Демонстрация принципов модуляции сигнала на примере аналоговой амплитудной модуляции. Исследование свойства самосинхронизации сигнала.

1.2. Теоретические сведения

1.2.1. Преобразование Фурье

Различают методы обработки сигналов во временной и в частотной области. Эквивалентность частотно-временных преобразований однозначно определяется через преобразование Фурье.

Преобразование Фурье — операция, сопоставляющая функции вещественной переменной другую функцию вещественной переменной, которая описывает коэффициенты («амплитуды») при разложении исходной функции на элементарные составляющие — гармонические колебания с разными частотами.

Для функции $f(x) \in \mathbb{R}$ преобразование Фурье имеет вид:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x) e^{-i\omega x} dx.$$

Обратное преобразование Фурье функции $\hat{f}(\omega)$:

$$F^{-1}(\hat{f}(\omega)) = f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega x} d\omega.$$

Функция $\hat{f}(\omega)$ называется *спектральной плотностью* или просто *спектром сигнала*.

Спектр периодического сигнала является дискретным и представляет набор гармонических колебаний, в сумме составляющий исходный сигнал. Разложение некоторого сигнала на составляющие называется *спектральным*. График зависимости параметров сигнала от частоты называется *спектральной диаграммой*. *Спектр сигнала* — совокупность простых составляющих сигнала с определенными амплитудами,

частотами и начальными фазами.

Изменение формы сигнала приводит к изменению его спектра (и наоборот).

1.2.2. Модуляция

Модуляция — процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала (сообщения).

Несущая — высокочастотное колебание, выполняющее роль переносчика информации, заложенной в управляющем (модулирующем) сигнале.

В качестве несущего колебания наиболее часто используют *гармоническое колебание*. В зависимости от того, какой из параметров несущего колебания — амплитуда, частота или начальная фаза несущего колебания изменяется по закону передаваемого сообщения, различают виды модуляции, соответственно, *амплитудная, частотная или фазовая*.

Сигнал, получаемый в процессе модуляции, называют *модулированным колебанием*, или радиосигналом.

В процессе *амплитудной модуляции* амплитуда U_0 несущего колебания $u_0(t) = U_0 \cos(\omega t + \varphi)$ перестает быть постоянной и изменяется по закону передаваемого сообщения. Амплитуда $U(t)$ несущего колебания может быть связана с передаваемым сообщением соотношением

$$U(t) = U_0 + k_A s(t),$$

где U_0 — амплитуда несущего колебания в отсутствии сообщения (немодулированное колебание); $s(t)$ — функция, зависящая от времени, соответствующая передаваемому сообщению (ее называют модулирующим сигналом); k_A — коэффициент пропорциональности, отражающий степень влияния модулирующего сигнала на величину изменения амплитуды результирующего сигнала (модулированного колебания).

Выражение для амплитудно-модулированного сигнала в общем случае имеет вид

$$u(t) = [U_0 + k_A s(t)] \cos(\omega_0 t + \varphi).$$

При *фазовой модуляции* значение фазового угла постоянной несущей частоты колебаний ω_0 пропорционально амплитуде модулирующего сигнала $s(t)$. Соответственно, уравнение ФМ-сигнала определяется выражением

$$u(t) = U \cos[\omega_0 t + k s(t)],$$

где k — коэффициент пропорциональности.

Частотная модуляция характеризуется линейной связью модулирующего сигнала с мгновенной частотой колебаний, при которой

мгновенная частота колебаний образуется сложением частоты высокочастотного несущего колебания ω_0 со значением амплитуды модулирующего сигнала с определенным коэффициентом пропорциональности:

$$\omega(t) = \omega_0 + ks(t).$$

1.2.3. Теорема Котельникова

Теорема Котельникова (теорема Найквиста–Шеннона или теорема отсчётов) гласит, что если аналоговый сигнал $x(t)$ имеет конечный (ограниченный по ширине) спектр, то он может быть восстановлен однозначно и без потерь по своим отсчётам, взятым с частотой, строго большей удвоенной верхней частоты f_c : $f > 2f_c$.

1.2.4. Кодирование сигнала

Одной из основных задач физического уровня модели OSI является преобразование данных в электромагнитные сигналы, и наоборот. Переход от электромагнитных импульсов к последовательности бит называют *кодированием сигнала*.

Способ представления исходного кода определёнными сигналами определяется форматом кода:

- *NRZ (Non Return to Zero)* — простейший двухуровневый код: логической единице соответствует верхний уровень, логическому нулю — нижний, переходы электрического сигнала происходят на границе битов;
- *AMI-код* (используется в телефонии): биты 0 представляются нулевым напряжением (0 В); биты 1 представляются поочерёдно значениями $-U$ или $+U$ (В);
- *RZ (Return to Zero)* — трёхуровневый код: обеспечивает возвращение к нулю после передачи каждого бита информации, в центре которого всегда есть переход, причём логической единице соответствует отрицательный импульс, логическому нулю — положительный;
- *манчестерский код* — двухуровневый код: логической единице соответствует переход вниз в центре бита, логическому нулю — переход вверх;
- *дифференциальный манчестерский код* — двухуровневый код: в течение битового интервала (времени передачи одного бита) уровень сигнала может меняться дважды, причём обязательно происходит изменение уровня в середине интервала и получается, что при передаче нуля в начале битового интервала происходит перепад уровней, а при передаче единицы такой перепад отсутствует.

1.3. Высокоуровневый интерпретируемый язык программирования Octave

1.3.1. Основы работы в Octave

Octave — высокоуровневый интерпретируемый язык программирования, предназначенный для решения задач вычислительной математики.

Интерпретатор Octave запускается из терминала операционной системы с помощью команды `octave` для работы с помощью консоли или `qt octave` для работы с оконным интерфейсом.

В окне интерпретатора Octave пользователь может вводить как отдельные команды языка Octave, так и группы команд, объединяемые в программы. Если строка заканчивается символом «;», то результаты на экран не выводятся. Если в конце строки символ «;» отсутствует, то результаты работы выводятся на экран. Текст в строке, который идет после символа %, является строкой комментария и интерпретатором не обрабатывается.

Octave имеет два режима работы: *терминальный* и *программный*. В терминальном режиме отдельные команды последовательно вводятся в окне интерпретатора. В программном режиме создается текстовый файл (с расширением `.m`), в котором хранятся последовательно выполняемые команды, впоследствии запускаемые на выполнение в среде Octave.

Простейшие арифметические операции в Octave:

- + — сложение;
- — вычитание;
- * — умножение;
- / — деление слева направо;
- \ — деление справа налево;
- ^ — возведение в степень.

Для определения переменной необходимо набрать *имя переменной*, символ «=» и *значение переменной*, где знак равенства — это оператор присваивания:

имя переменной = значение выражения

Система различает большие и малые буквы в именах переменных. Выражение в правой части оператора присваивания может быть числом, арифметическим выражением, строкой символов или символьным выражением. Если речь идет о символьной или строковой переменной, то выражение в правой части оператора присваивания следует брать в одинарные кавычки.

Если команда не содержит знака присваивания, то по умолчанию вычисленное значение присваивается специальной системной переменной `ans`. Причем полученное значение можно использовать в последу-

ющих вычислениях, но важно помнить, что значение **ans** изменяется после каждого вызова команды без оператора присваивания.

Системные переменные:

- **ans** – результат последней операции без знака присваивания;
- **i**, **j** – мнимая единица ($\sqrt{-1}$);
- **pi** – число π (3.141592653589793);
- **e** – число e (экспонента 2.71828183);
- **inf** – машинный символ бесконечности (∞);
- **NaN** – неопределенный результат;
- **realmin** – наименьшее число с плавающей точкой (2.2251e-308);
- **realmax** – наибольшее число с плавающей точкой (1.7977e+308).

Все перечисленные переменные можно использовать в математических выражениях.

Команда **clear** предназначена для уничтожения определения одной или нескольких переменных:

clear имя переменной

В общем виде обращение к функции в Octave имеет вид

имя переменной = имя функции(аргумент)

или

имя функции(аргумент)

Если имя переменной указано, то ей будет присвоен результат работы функции. Если же оно отсутствует, то значение вычисленного функцией результата присваивается системной переменной **ans**.

Примеры работы с тригонометрическими функциями:

```
>>> x=pi/2;    % Определение значения аргумента
>>> y=sin(x)    % Вызов функции
y = 1
>>> cos(pi/3) % Вызов функции
ans = 0.50000
```

Здесь >>> — знак приглашения Octave для ввода команд.

Встроенные функции Octave:

- **sin(x)** — синус числа x ;
- **cos(x)** — косинус числа x ;
- **tan(x)** — тангенс числа x ;
- **cot(x)** — котангенс числа x ;
- **sec(x)** — секанс числа x ;
- **csc(x)** — cosecant числа x ;
- **asin(x)** — арксинус числа x ;
- **acos(x)** — арккосинус числа x ;
- **atan(x)** — арктангенс числа x ;
- **acot(x)** — арккотангенс числа x ;
- **asec(x)** — арксеканс числа x ;
- **acsc(x)** — арккосеканс числа x ;
- **exp(x)** — экспонента числа x ;
- **log(x)** — натуральный логарифм числа x ;

- `sinh(x)` — гиперболический синус числа x ;
- `cosh(x)` — гиперболический косинус числа x ;
- `tanh(x)` — гиперболический тангенс числа x ;
- `coth(x)` — гиперболический котангенс числа x ;
- `sech(x)` — гиперболический секанс числа x ;
- `csch(x)` — гиперболический косеканс числа x ;
- `fix(x)` — округление числа x до ближайшего целого в сторону нуля;
- `floor(x)` — округление числа x до ближайшего целого в сторону отрицательной бесконечности;
- `ceil(x)` — округление числа x до ближайшего целого в сторону положительной бесконечности;
- `round(x)` — обычное округление числа x до ближайшего целого;
- `rem(x, y)` — вычисление остатка от деления x на y ;
- `sign(x)` — сигнум-функция числа x , выдаёт 0, если $x = 0$, -1 — при $x < 0$ и 1 при $x > 0$;
- `sqrt(x)` — корень квадратный из числа x ;
- `abs(x)` — модуль числа x ;
- `log10(x)` — десятичный логарифм от числа x ;
- `log2(x)` — логарифм по основанию два от числа x ;
- `pow2(x)` — возведение двойки в степень x ;
- `gcd(x, y)` — наибольший общий делитель чисел x и y ;
- `lcm(x, y)` — наименьшее общее кратное чисел x и y ;
- `rats(x)` — представление числа x в виде рациональной дроби;
- `real(Z)` — выдаёт действительную часть комплексного аргумента Z ;
- `imag(Z)` — выдаёт мнимую часть комплексного аргумента Z ;
- `angle(Z)` — вычисляет значение аргумента комплексного числа Z в радианах от $-\pi$ до π ;
- `conj(Z)` — выдаёт число, комплексно сопряжённое Z .

Операции отношения выполняют сравнение двух операндов и определяют, истинно выражение или ложно:

- `<` — меньше;
- `>` — больше;
- `=` — равно;
- `~=` — не равно;
- `<=` — меньше или равно;
- `>=` — больше или равно.

Синтаксис функции, определяемой пользователем:

```
function name1 [, name2, ...] = fun(var1 [, var2, ...])
```

Здесь `name1 [, name2, ...]` — список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений, `fun` — имя функции, `var1 [, var2, ...]` — входные параметры.

Все имена переменных внутри функции, а также имена из списка входных и выходных параметров воспринимаются системой как локальные, т.е. эти переменные считаются определенными только внутри

функции. Программы и функции в Octave могут быть созданы при помощи текстового редактора и сохранены в виде файла с расширением `.m` или `.M`. Но при создании и сохранении функции следует помнить, что ее имя должно совпадать с именем файла. Программу можно запустить на выполнение, указав имя файла, в котором она сохранена. Обращение к функции осуществляется так же, как и к любой другой встроенной функции системы, то есть с указанием входных и выходных параметров. Вызвать функцию можно из командной строки или использовать ее как один из операторов программы.

Массив — множественный тип данных, состоящий из фиксированного числа элементов одного типа. Как и любой другой переменной, массиву должно быть присвоено имя.

Самый простой способ задать одномерный массив в Octave имеет вид

имя массива = $X_n:dX:X_k$

Здесь X_n — значение первого элемента массива, X_k — значение последнего элемента массива, dX — шаг, с помощью которого формируется каждый следующий элемент массива, т.е. значение второго элемента составит X_n+dX , третьего $X_n+dX+dX$ и так далее до X_k .

Примеры создания массивов:

```
>>> A=1:5
A =
1 2 3 4 5
>>> B=2:2:10
B =
2 4 6 8 10
>>> xn=-3.5; xk=3.5; dx=0.5;
>>> X=xn:dx:xk
X =
Columns 1 through 8:
-3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5 0.0
Columns 9 through 15:
0.5 1.0 1.5 2.0 2.5 3.0 3.5
```

Обратиться к элементу вектора можно, указав имя массива и порядковый номер элемента в круглых скобках:

```
>>> x=[2 4 6 8 10];
>>> y=[-1.2 3.4 -0.8 9.1 5.6 -7.3];
>>> x(1)%значение первого элемента массива x
ans = 2
>>> y(5)%значение пятого элемента массива y
ans = 5.6000
>>> x(1)/2+y(3)^2-x(4)/y(5)
ans = 0.21143
```

Ввод элементов матрицы также осуществляется в квадратных скобках, при этом элементы строки отделяются друг от друга пробелом или запятой, а строки разделяются между собой точкой с запятой.

Обратиться к элементу матрицы можно, указав после имени матрицы, в круглых скобках, через запятую, номер строки и номер столбца, на пересечении которых элемент расположен:

```
>>> M=[2 4 6;1 3 5;7 8 9]
M =
  2 4 6
  1 3 5
  7 8 9
>>> M(1,2)
ans = 4
>>> M(3,1)
ans = 7
```

1.3.2. Построение графиков в Octave

Для того чтобы построить двумерный график функции $f(x)$, необходимо сформировать два массива x и y одинаковой размерности, а затем обратиться к функции `plot`.

Синтаксис функции `plot`:

```
plot(x1, y1, s1, x2, y2, s2, ..., xn, yn, sn)
```

Здесь x_1, x_2, \dots, x_n — массивы абсцисс графиков; y_1, y_2, \dots, y_n — массивы ординат графиков; s_1, s_2, \dots, s_n — строка форматов, определяющая параметры линии и, при необходимости, позволяющая вывести легенду.

В строке форматов могут участвовать символы, отвечающие за тип линии, маркер и его размер, цвет линии и вывод легенды. За сплошную линию отвечает символ «-». Цвет линии определяется буквой латинского алфавита: **y** — жёлтый, **m** — розовый, **c** — голубой, **r** — красный, **g** — зелёный, **b** — синий, **w** — белый. Некоторые символы маркера: **.** — точка, ***** — звёздочка, **x** — крестик, **+** — плюс, **o** — незакрашенный круг, **p** — незакрашенный квадрат.

Например, для построения графика функции $y = \sin x + \frac{1}{3} \sin 3x + \frac{1}{5} \sin 5x$ на интервале $[-10; 10]$ (рис. 1.1) можно использовать следующий листинг:

```
% Формирование массива x:
x=-10:0.1:10;
% Формирование массива y.
y=sin(x)+1/3*sin(3*x)+1/5*sin(5*x);
% Построение графика функции:
plot(x,y, "-ok; y=sin(x)+ (1/3)*sin(3*x)+(1/5)*sin(5*x);",
      "markersize",4)
% Отображение сетки на графике
grid on;
```

```
% Подпись оси X:
xlabel('x');
% Подпись оси Y:
ylabel('y');
Название графика:
title('y=sin x+ (1/3)sin(3x)+(1/5)sin(5x)');
% Экспорт рисунка в файл .eps:
print ("plot-sin.eps", "-mono", "-FArial:16", "-deps")
```

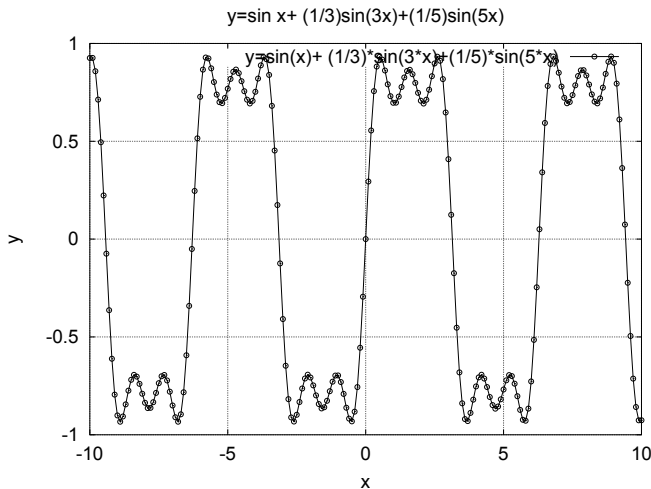


Рис. 1.1. График функции $y = \sin x + \frac{1}{3} \sin 3x + \frac{1}{5} \sin 5x$ на интервале $[-10; 10]$

Если повторно обратиться к функции `plot`, то в этом же окне будет стёрт первый график и нарисован второй. Для построения нескольких графиков в одной системе координат можно поступить одним из следующих способов:

- 1) обратиться к функции `plot` следующим образом:
`plot(x1,y1,x2,y2,... xn,yn)`,
 где x_1, y_1 — массивы абсцисс и ординат первого графика, x_2, y_2 — массивы абсцисс и ординат второго графика, ..., x_n, y_n — массивы абсцисс и ординат n -го графика;
- 2) каждый график изображать с помощью функции `plot(x,y)`, но перед обращением к функциям `plot(x2,y2)`, `plot(x3,y3)`, ..., `plot(xn,yn)` вызвать команду `hold on`, которая блокирует режим очистки окна.

1.4. Порядок выполнения работы

1. Ознакомиться с теоретическим материалом (разделы 1.1.–1.3).
2. Выполнить задания разделов 1.4.1–1.4.4. Получить соответствующие графики.
3. Составить отчёт о выполненной работе, включив в него листинги программ и полученные графики.

1.4.1. Разложение импульсного сигнала в частичный ряд Фурье

В цифровой технике основным типом сигналов является импульсный сигнал. Импульсный сигнал можно описать математически в синусоидальной форме. Такой тип сигнала называется меандром.

Меандр — бесконечный, периодический сигнал прямоугольной формы (импульсный сигнал), широко используемый в радиотехнике. Длительность импульса и длительность паузы в периоде такого сигнала равны.

Спектр меандра имеет вид

$$s(t) = \frac{A}{2} + \frac{2A}{\pi} \left(\cos\left(\frac{2\pi}{T}t\right) - \frac{1}{3} \cos\left(3\frac{2\pi}{T}t\right) + \frac{1}{5} \cos\left(5\frac{2\pi}{T}t\right) - \dots \right).$$

Гармоники, образующие меандр, имеют амплитуду, обратно пропорциональную номеру соответствующей гармоники.

Задание: разработать код m-файла, результатом выполнения которого являются графики меандра (рис. 1.2), реализованные с различным количеством гармоник.

Листинг программы в Octave:

```
% meandr.m
% количество отсчетов:
N=8;
% частота дискретизации:
t=-1:0.01:1;
% значение амплитуды:
A=1;
T=1;
nh=(1:N)*2-1;
% входной сигнал:
harmonics=cos(2*pi*nh'*t/T);
Am=2/pi./nh;
Am(2:2:end)=-Am(2:2:end);
s1=harmonics.*repmat(Am',1,length(t));
s2=cumsum(s1);
for k=1:N
```



```

subplot(4,2,k)
plot(t, s2(k,:))
end

```

Здесь функция `repmat(A,M,N)` формирует массив из частей; имеет три входных аргумента: массив `A`, количество строк `M` и столбцов `N` для вновь создаваемого массива; `cumsum` — суммирование элементов массива с накоплением.

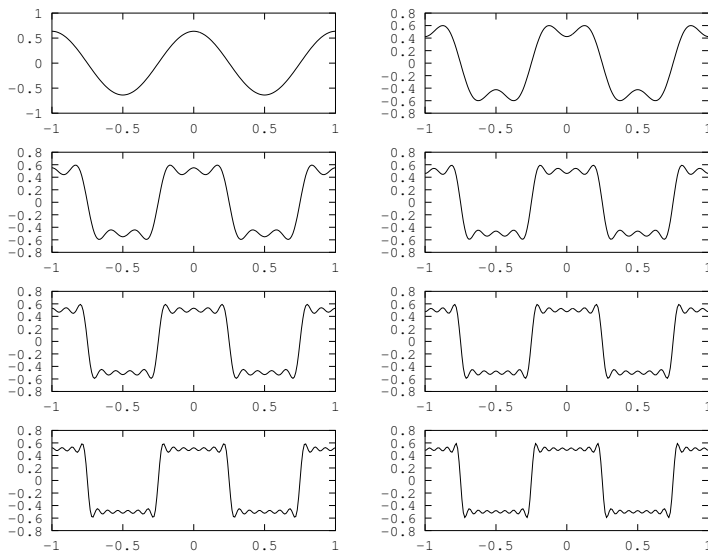


Рис. 1.2. Графики меандра, содержащего различное число гармоник

1.4.2. Определение спектра и параметров сигнала

Задание. Определить спектр двух отдельных сигналов и их суммы.

Частота дискретизации (количество отсчётов) выбирается на основе теоремы Котельникова как удвоенная ширина спектра исходного сигнала (таким образом, в следующем примере достаточно было взять частоту дискретизации 80 Гц).

Попробуйте выполнить задание с другой частотой дискретизации. Что будет, если взять частоту дискретизации меньше 80 Гц?

Для двух синусоидальных сигналов (рис. 1.3) требуется определить их спектр. В файле `spectre.m` задаем параметры сигналов:

```
% spectre.m
mkdir 'signal';
mkdir 'spectre';
tmax = 0.5;% Длина сигнала (с)
fd = 512;% Частота дискретизации (Гц) (количество отсчётов)
f1 = 10;% Частота первого сигнала (Гц)
f2 = 40;% Частота второго сигнала (Гц)
a1 = 1;% Амплитуда первого сигнала
a2 = 0.7;% Амплитуда второго сигнала
fd2 = fd/2;% Спектр сигнала
% Рассмотрим два сигнала (синусоиды) разной частоты
t = 0:1./fd:tmax;% Массив отсчётов времени
signal1 = a1*sin(2*pi*t*f1);
signal2 = a2*sin(2*pi*t*f2);
plot(signal1,'b'); % голубая
hold on
plot(signal2,'r'); % красная
hold off
title('Signal');
print 'signal/spectre.png';
```

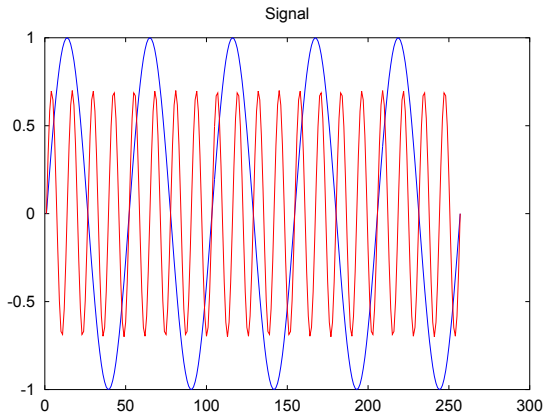


Рис. 1.3. Два синусоидальных сигнала разной частоты

С помощью быстрого преобразования Фурье найдем спектры сигналов (рис. 1.4), добавив в файл `spectre.m` следующий код.

```
% Посчитаем спектр
% Амплитуды преобразования Фурье сигнала 1
```

```

spectre1 = abs(fft(signal1,fd));
% Амплитуды преобразования Фурье сигнала 2
spectre2 = abs(fft(signal2,fd));
% Построение спектров сигналов
plot(spectre1,'b'); % голубая
hold on
plot(spectre2,'r'); % красная
hold off
title('Spectre');
print 'spectre/spectre.png';

```

Учитывая реализацию преобразования Фурье, скорректируем график спектра (рис. 1.5): отбрасываются дублирующие отрицательные частоты, а также учитывается то, что на каждом шаге вычисления быстрого преобразования Фурье происходит суммирование амплитуд сигналов. Добавляем в файл `spectre.m` следующий код.

```

% Исправление графика спектра
f = 1000*(0:fd2)./(2*fd); % Сетка частот
% Нормировка спектров по амплитуде
spectre1 = 2*spectre1/fd2;
spectre2 = 2*spectre2/fd2;
% Построение спектров сигналов
plot(f,spectre1(1:fd2+1),'b'); % голубая
hold on
plot(f,spectre2(1:fd2+1),'r'); % красная
hold off
xlim([0 100]);
title('Fixed spectre');
xlabel('Frequency (Hz)');
print 'spectre/spectre_fix.png';

```

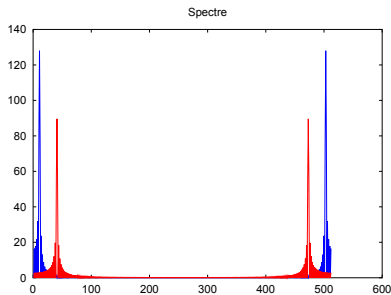


Рис. 1.4. График спектров синусоидальных сигналов

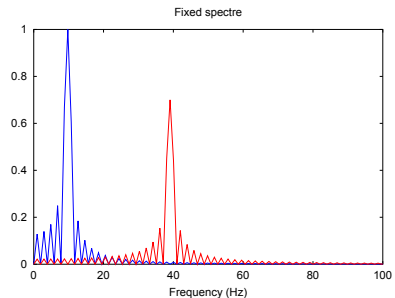


Рис. 1.5. Исправленный график спектров синусоидальных сигналов

Аналогично найдем спектр суммы рассмотренных сигналов (1.6), создав файл `spectre_sum.m` со следующим кодом.

```
% spectre_sum.m
mkdir 'signal';
mkdir 'spectre';
% Длина сигнала (с)
tmax = 0.5;
% Частота дискретизации (Гц) (количество отсчётов)
fd = 512;
% Частота первого сигнала (Гц)
f1 = 10;
% Частота второго сигнала (Гц)
f2 = 40;
% Амплитуда первого сигнала
a1 = 1;
% Амплитуда второго сигнала
a2 = 0.7;
% Спектр сигнала
fd2 = fd/2;

% Сумма двух сигналов (синусоиды) разной частоты

% Массив отсчётов времени:
t = 0:1./fd:tmax;
signal1 = a1*sin(2*pi*t*f1);
signal2 = a2*sin(2*pi*t*f2);
signal = signal1 + signal2;
plot(signal);
title('Signal');
print 'signal/spectre_sum.png';

% Подсчет спектра:
% Амплитуды преобразования Фурье сигнала
spectre = fft(signal,fd);
% Сетка частот
f = 1000*(0:fd2)./(2*fd);

% Нормировка спектра по амплитуде:
spectre = 2*sqrt(spectre.*conj(spectre))./fd2;

% Построение спектра сигнала
plot(f,spectre(1:fd2+1))
xlim([0 100]);
title('Spectre');
xlabel('Frequency (Hz)');
print 'spectre/spectre_sum.png';
```

В результате получим аналогичный предыдущему результат (рис. 1.7), т.е. спектр суммы сигналов равен сумме спектров сигналов, что вытекает из свойств преобразования Фурье.

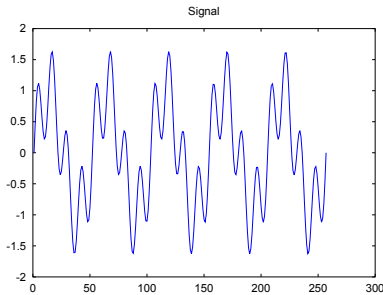


Рис. 1.6. Суммарный сигнал

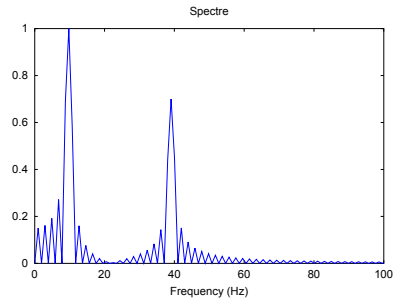


Рис. 1.7. Спектр суммарного сигнала

1.4.3. Демонстрация принципов модуляции сигнала на примере аналоговой амплитудной модуляции

Следующий код в файле `am.m` демонстрирует принципы модуляции сигнала на примере аналоговой амплитудной модуляции (рис.1.8).

```
% am.m
mkdir 'signal';
mkdir 'spectre';

% Модуляция синусоид с частотами 50 и 5
% Длина сигнала (с)
tmax = 0.5;
% Частота дискретизации (Гц) (количество отсчётов)
fd = 512;
% Частота сигнала (Гц)
f1 = 5;
% Частота несущей (Гц)
f2 = 50;
% Спектр сигнала
fd2 = fd/2;

% Построение графиков двух сигналов (синусоиды)
% разной частоты
% Массив отсчётов времени:
t = 0:1./fd:tmax;
```

```

signal1 = sin(2*pi*t*f1);
signal2 = sin(2*pi*t*f2);
signal = signal1 .* signal2;
plot(signal, 'b');
hold on

% Построение огибающей:
plot(signal1, 'r');
plot(-signal1, 'r');
hold off
title('Signal');
print 'signal/am.png';

% Расчет спектра:
% Амплитуды преобразования Фурье-сигнала
spectre = fft(signal,fd);
% Серка частот
f = 1000*(0:fd2)./(2*fd);

% Нормировка спектра по амплитуде:
spectre = 2*sqrt(spectre.*conj(spectre))./fd2;

% Построение спектра:
plot(f,spectre(1:fd2+1), 'b')
xlim([0 100]);
title('Spectre');
xlabel('Frequency (Hz)');
print 'spectre/am.png';

```

В результате получаем, что спектр произведения есть свёртка спектров (рис. 1.9).

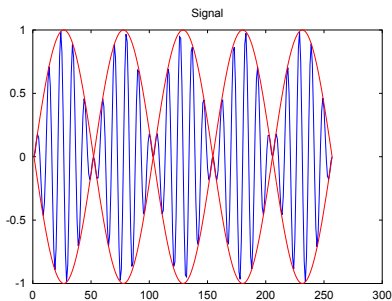


Рис. 1.8. Сигнал и огибающая при амплитудной модуляции

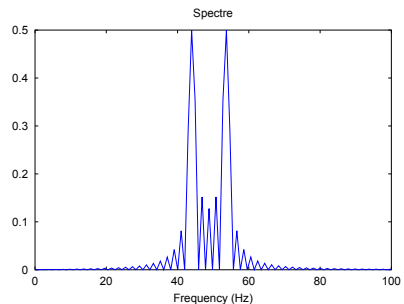


Рис. 1.9. Спектр сигнала при амплитудной модуляции

1.4.4. Кодирование сигнала. Исследование свойства самосинхронизации сигнала

По заданной выходной битовой последовательности требуется получить кодированный сигнал для нескольких кодов, проверить свойства самосинхронизируемости кода.

Создаём файл `main.m`:

```
% main.m
```

```
pkg load signal;
```

```
% Задаем входную кодовую последовательность:
```

```
data=[0 1 0 0 1 1 0 0 0 1 1 0];
```

```
% Задаем входную кодовую последовательность
```

```
% для проверки свойства самосинхронизации:
```

```
data_sync=[0 0 0 0 0 0 0 1 1 1 1 1 1];
```

```
% Построение графиков кодированного сигнала
```

```
mkdir 'signal';
```

```
axis("auto");
```

```
% Униполярное кодирование
```

```
wave=unipolar(data);
```

```
plot(wave);
```

```
ylim([-1 6]);
```

```
title('Unipolar');
```

```
print 'signal/unipolar.png';
```

```
% Кодирование ami
```

```
wave=ami(data);
```

```
plot(wave)
```

```
title('AMI');
```

```
print 'signal/ami.png';
```

```
% Кодирование NRZ
```

```
wave=bipolarnrz(data);
```

```
plot(wave);
```

```
title('Bipolar Non-Return to Zero');
```

```
print 'signal/bipolarnrz.png';
```

```
% Кодирование RZ
```

```
wave=bipolarrz(data);
```

```
plot(wave)
```

```
title('Bipolar Return to Zero');
```

```
print 'signal/bipolarrz.png';
```

```
% Манчестерское кодирование
wave=manchester(data);
plot(wave)
title('Manchester');
print 'signal/manchester.png';

% Дифференциальное манчестерское кодирование
wave=diffmanc(data);
plot(wave)
title('Differential Manchester');
print 'signal/diffmanc.png';

% Построение графиков кодированного сигнала
% для проверки свойства самосинхронизации
mkdir 'sync';
axis("auto");

% Униполярное кодирование
wave=unipolar(data_sync);
plot(wave);
ylim([-1 6]);
title('Unipolar');
print 'sync/unipolar.png';

% Кодирование AMI
wave=ami(data_sync);
plot(wave)
title('AMI');
print 'sync/ami.png';

% Кодирование NRZ
wave=bipolarnrz(data_sync);
plot(wave);
title('Bipolar Non-Return to Zero');
print 'sync/bipolarnrz.png';

% Кодирование RZ
wave=bipolarrz(data_sync);
plot(wave)
title('Bipolar Return to Zero');
print 'sync/bipolarrz.png';

% Манчестерское кодирование
wave=manchester(data_sync);
plot(wave)
title('Manchester');
```



```
print 'sync/manchester.png';

% Дифференциальное манчестерское кодирование
wave=diffmanc(data_sync);
plot(wave)
title('Differential Manchester');
print 'sync/diffmanc.png';
```

Следующая функция (в отдельном файле `maptowave.m`) по входному битовому потоку строит график сигнала:

```
% maptowave.m
function wave=maptowave(data)
    data=upsample(data,100);
    wave=filter(5*ones(1,100),1,data);
```

Каждая функция преобразования кодовой последовательности находится в отдельном файле. Например, униполярное кодирование реализуется с помощью следующей функции:

```
% unipolar.m
function wave=unipolar(data)
    wave=maptowave(data);
```

Кодирование АМІ реализуется с помощью следующей функции:

```
% ami.m
function wave=ami(data)
    am=mod(1:length(data(data==1)),2);
    am(am==0)=-1;
    data(data==1)=am;
    wave=maptowave(data);
```

Кодирование NRZ реализуется с помощью следующей функции:

```
% bipolarnrz.m
function wave=bipolarnrz(data)
    data(data==0)=-1;
    wave=maptowave(data);
```

Кодирование RZ реализуется с помощью следующей функции:

```
% bipolarrrz.m
function wave=bipolarrrz(data)
    data(data==0)=-1;
    data=upsample(data,2);
    wave=maptowave(data);
```

Манчестерское кодирование реализуется с помощью следующей функции:

```
% manchester.m
function wave=manchester(data)
    data(data==0)=-1;
    data=upsample(data,2);
    data=filter([-1 1],1,data);
    wave=maptowave(data);
```

Дифференциальное манчестерское кодирование реализуется с помощью следующей функции:

```
% diffmanc.m
function wave=diffmanc(data)
    data=filter(1,[1 1],data);
    data=mod(data,2);
    wave=manchester(data);
```

Для построения спектра сигнала реализуем следующую функцию (в отдельном файле calcspectre.m):

```
% calcspectre.m
function spectre = calcspectre(wave)
    Fd = 512; % Частота дискретизации (Гц)
    Fd2 = Fd/2;
    Fd3 = Fd/2 + 1;
    X = fft(wave,Fd);
    spectre = X.*conj(X)/Fd;
    f = 1000*(0:Fd2)/Fd;
    plot(f,spectre(1:Fd3));
    xlabel('Frequency (Hz)');
```

и в файл main.m добавляем следующий код:

```
% Построение спектра сигнала
mkdir 'spectre';
axis("auto");
data_spectre=[0 1 0 1 0 1 0 1 0 1 0 1 0 1];

% униполярное кодирование
wave=unipolar(data_spectre);
spectre=calcspectre(wave);
title('Unipolar');
print 'spectre/unipolar.png';

% кодирование AMI
wave=ami(data_spectre);
spectre=calcspectre(wave);
title('AMI');
print 'spectre/ami.png';
```

```
% кодирование NRZ
wave=bipolarnrz(data_spectre);
spectre=calcspectre(wave);
title('Bipolar Non-Return to Zero');
print 'spectre/bipolarnrz.png';

% кодирование RZ
wave=bipolarrz(data_spectre);
spectre=calcspectre(wave);
title('Bipolar Return to Zero');
print 'spectre/bipolarrz.png';

% манчестерское кодирование
wave=manchester(data_spectre);
spectre=calcspectre(wave);
title('Manchester');
print 'spectre/manchester.png';

% дифференциальное манчестерское кодирование
wave=diffmanc(data_spectre);
spectre=calcspectre(wave);
title('Differential Manchester');
print 'spectre/diffmanc.png';
```

Запускаем главный скрипт `main.m`. В каталоге `signal` получаем поведение кодированного сигнала (рис. 1.10–1.15).

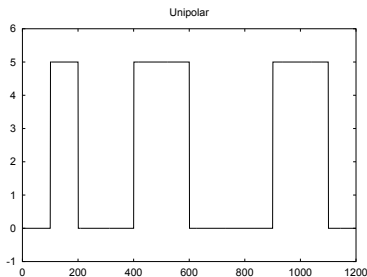


Рис. 1.10. Униполярное кодирование

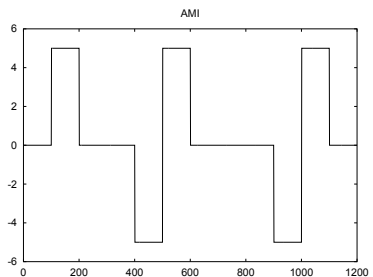


Рис. 1.11. Кодирование AMI

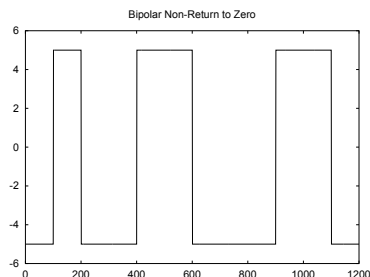


Рис. 1.12. Кодирование NRZ

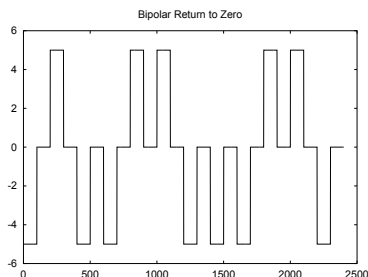


Рис. 1.13. Кодирование RZ

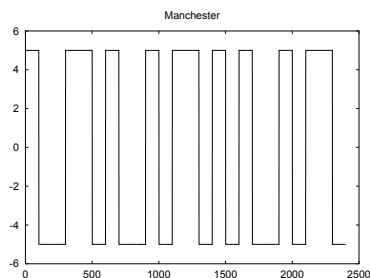


Рис. 1.14. Манчестерское кодирование

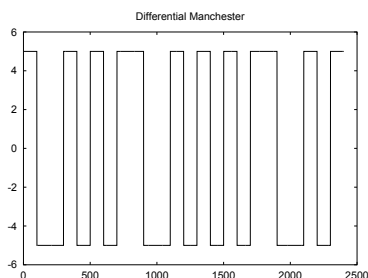


Рис. 1.15. Дифференциальное манчестерское кодирование

В каталоге `sync` иллюстрируются свойства самосинхронизации (рис. 1.16–1.21).

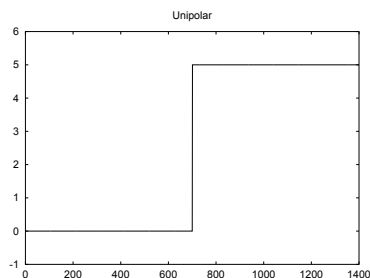


Рис. 1.16. Униполярное кодирование: нет самосинхронизации

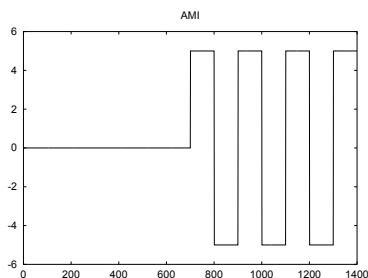


Рис. 1.17. Кодирование AMI: самосинхронизация при наличии сигнала

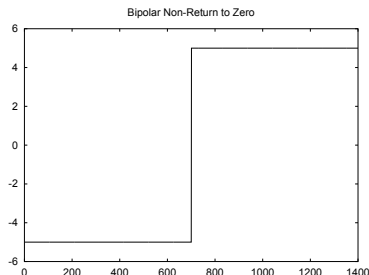


Рис. 1.18. Кодирование NRZ:
нет самосинхронизации

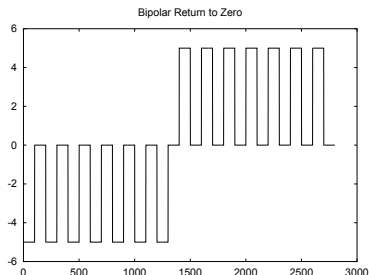


Рис. 1.19. Кодирование RZ:
есть самосинхронизация

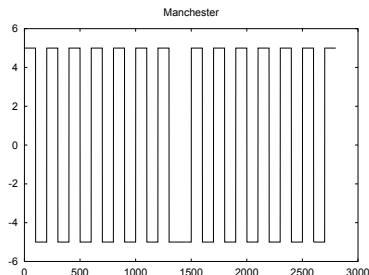


Рис. 1.20. Манчестерское кодирование:
есть самосинхронизация

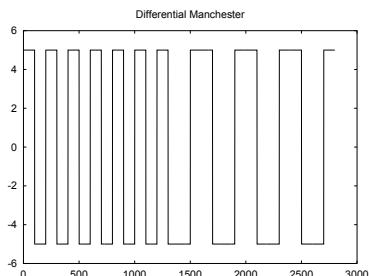


Рис. 1.21. Дифференциальное манчестерское кодирование:
есть самосинхронизация

В каталоге `spectre` получаем спектр сигнала (рис. 1.22–1.27).

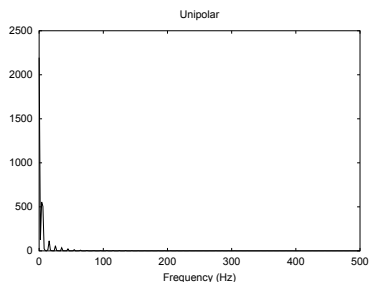


Рис. 1.22. Униполярное кодирование: спектр сигнала

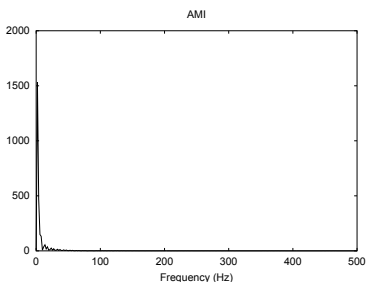
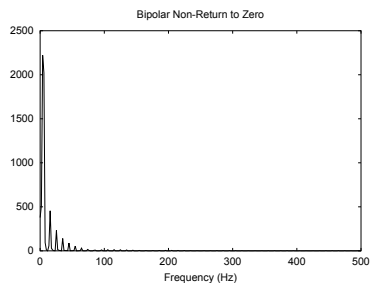
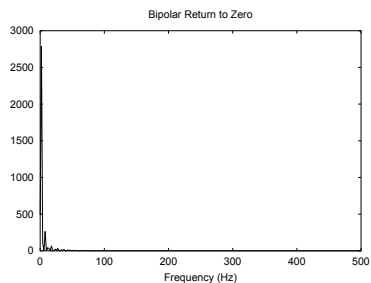


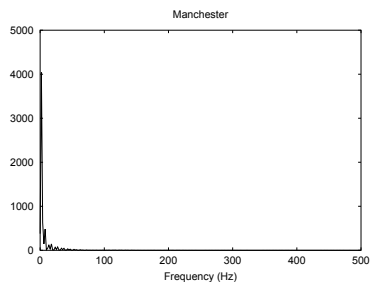
Рис. 1.23. Кодирование AMI: спектр сигнала



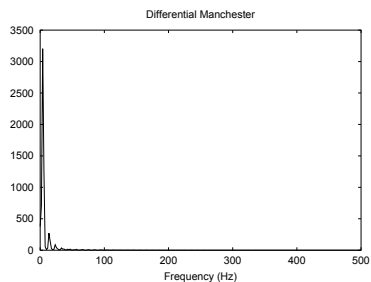
**Рис. 1.24. Кодирование NRZ:
спектр сигнала**



**Рис. 1.25. Кодирование RZ:
спектр сигнала**



**Рис. 1.26. Манчестерское
кодирование: спектр сигнала**



**Рис. 1.27. Дифференциальное
манчестерское кодирование:
спектр сигнала**

Глава 2. Расчёт сети Fast Ethernet

2.1. Цели работы

Цель данной работы — изучение принципов технологий Ethernet и Fast Ethernet и практическое освоение методик оценки работоспособности сети, построенной на базе технологии Fast Ethernet.

2.2. Теоретические сведения

2.2.1. Технология Ethernet

Спецификация сети Ethernet была предложена фирмами DEC, Intel и Xerox (DIX) в 1980 г., и несколько позже на её основе появился стандарт IEEE 802.3.

Первые версии Ethernet v1.0 и Ethernet v2.0 в качестве среды передачи использовали только коаксиальный кабель. Стандарт IEEE 802.3 позволяет в качестве среды передачи использовать также витую пару и оптоволокно. В 1995 г. был принят стандарт IEEE 802.3u (Fast Ethernet) со скоростью 100 Мбит/с, а в 1997 г. — IEEE 802.3z (Gigabit Ethernet — 1000 Мбит/с). Осенью 1999 г. принят стандарт IEEE 802.3ab — Gigabit Ethernet на витой паре категории 5.

В обозначениях Ethernet (10BASE2, 100BASE-TX и др.) первый элемент обозначает скорость передачи данных в Мбит/с; второй элемент BASE означает, что используется прямая (немодулированная) передача; третий элемент обозначает округлённое значение длины кабеля в сотнях метров (10BASE2 — 185 м, 10BASE5 — 500 м) или тип среды передачи (Т, TX, T2, T4 — витая пара; FX, FL, FB, SX и LX — оптоволокно; CX — твинаксиальный кабель для Gigabit Ethernet).

В основе Ethernet лежит *метод множественного доступа к среде передачи с прослушиванием несущей и обнаружением коллизий* — CSMA/CD (*Carrier Sense with Multiple Access and Collision Detection*), реализуемый адаптерами каждого узла сети на аппаратном или микропрограммном уровне:

- все адаптеры имеют устройство доступа к среде (MAU) — трансивер, подключённый к общей (разделяемой) среде передачи данных;
- каждый адаптер узла перед передачей информации прослушивает линию до момента отсутствия сигнала (несущей);
- затем адаптер формирует кадр (frame), начинающийся с синхронизирующей преамбулы, за которой следует поток двоичных данных в самосинхронизирующемся (манчестерском) коде;

- другие узлы принимают посланный сигнал, синхронизируются по преамбуле и декодируют его в последовательность бит;
- окончание передачи кадра определяется обнаружением приёмником отсутствия несущей;
- в случае обнаружения *коллизии* (столкновения двух сигналов от разных узлов) передающие узлы прекращают передачу кадра, после чего через случайный промежуток времени (каждый через свой) осуществляют повторную попытку передачи после освобождения линии; пр5. Communications in Computer and Information Scienceи очередной неудаче делается следующая попытка (и так до 16 раз), причём интервал задержки увеличивается;
- коллизия обнаруживается приёмником по нестандартной длине кадра, которая не может быть меньше 64 байт, не считая преамбулы;
- между кадрами должен обеспечиваться временной зазор (*межкадровый или межпакетный промежуток, IPG – inter-packet gap*) длительностью 9,6 мкс — узел не имеет права начать передачу раньше, чем через интервал IPG, после определения момента пропадания несущей.

ОПРЕДЕЛЕНИЕ 1. *Домен коллизий* — группа узлов, связанных общей средой (кабелями и повторителями) передачи.

Протяжённость домена коллизий ограничивается временем распространения сигнала между наиболее удалёнными друг от друга узлами.

ОПРЕДЕЛЕНИЕ 2. *Диаметр домена коллизий* — расстояние между двумя наиболее удалёнными друг от друга оконечными устройствами.

ОПРЕДЕЛЕНИЕ 3. *Битовый интервал* — время, необходимое для передачи одного бита.

Битовый интервал в Ethernet (при скорости 10 Мбит/с) составляет 0,1 мкс.

2.2.2. Технология Fast Ethernet

В технологии Fast Ethernet величина битового интервала составляет 0,01 мкс, что даёт десятикратное увеличение скорости передачи данных. При этом формат кадра, объём переносимых кадром данных и механизм доступа к каналу передачи данных остались без изменения по сравнению с Ethernet.

В Fast Ethernet используется среда передачи данных для работы на скорости 100 Мбит/с, которая в спецификации IEEE 802.3u имеет обозначения «100BASE-T4» и «100BASE-TX» (витая пара); «100BASE-FX» и «100BASE-SX» (оптоволокно).

2.2.2.1. Правила построения сети

Первая модель сети Fast Ethernet. Модель представляет собой, по сути, набор правил построения сети (табл. 2.1):

- длина каждого сегмента витой пары должна быть меньше 100 м;
- длина каждого оптоволоконного сегмента должна быть меньше 412 м;
- если используются кабели МП (Media Independent Interface), то каждый из них должен быть меньше 0,5 м;
- задержки, вносимые кабелем МП, не учитываются при оценке временных параметров сети, так как они являются составной частью задержек, вносимых оконечными устройствами (терминалами) и повторителями.

Стандартом определены два класса повторителей:

- повторители класса I выполняют преобразование входных сигналов в цифровой вид, а при передаче снова перекодируют цифровые данные в физические сигналы; преобразование сигналов в повторителе требует некоторого времени, поэтому в домене коллизий допускается только один повторитель класса I;
- повторители класса II немедленно передают полученные сигналы без всякого преобразования, поэтому к ним можно подключать только сегменты, использующие одинаковые способы кодирования данных; можно использовать не более двух повторителей класса II в одном домене коллизий.

Таблица 2.1

Предельно допустимый диаметр домена коллизий в Fast Ethernet

Тип повторителя	Все сегменты TX или T4	Все сегменты FX	Сочетание сегментов (T4 и TX/FX)	Сочетание сегментов (TX и FX)
Сегмент, соединяющий два узла без повторителей	100	412,0	–	–
Один повторитель класса I	200	272,0	231,0	260,8
Один повторитель класса II	200	320,0	–	308,8
Два повторителя класса II	205	228,0	–	216,2

Вторая модель сети Fast Ethernet. Вторая модель содержит последовательность расчётов временных параметров сети при полудуплексном режиме обмена данными. Диаметр домена коллизий и количество сегментов в нём ограничены временем двойного оборота, необходимым для правильной работы механизма обнаружения и разрешения коллизий (табл. 2.2).

Время двойного оборота рассчитывается для наихудшего (в смысле распространения сигнала) пути между двумя узлами домена коллизий. Расчёт выполняется путём суммирования временных задержек в сегментах, повторителях и терминалах.

Таблица 2.2

Временные задержки компонентов сети Fast Ethernet

Компонент	Удельное время двойного оборота (би/м)	Максимальное время двойного оборота (би)
Пара терминалов TX/FX	–	100
Пара терминалов T4	–	138
Пара терминалов T4 и TX/FX	–	127
Витая пара категории 3	1,14	114 (100 м)
Витая пара категории 4	1,14	114 (100 м)
Витая пара категории 5	1,112	111,2 (100 м)
Экранированная витая пара	1,112	111,2 (100 м)
Оптоволокно	1,0	412 (412 м)
Повторитель класса I	–	140
Повторитель класса II, имеющий порты типа TX/FX	–	92
Повторитель класса II, имеющий порты типа T4	–	67

Для вычисления времени двойного оборота нужно умножить длину сегмента на величину удельного времени двойного оборота соответствующего сегмента. Определив времена двойного оборота для всех сегментов наихудшего пути, к ним нужно прибавить задержку, вносимую парой оконечных узлов и повторителями. Для учёта непредви-

денных задержек к полученному результату рекомендуется добавить ещё 4 битовых интервала (би) и сравнить результат с числом 512. Если полученный результат не превышает 512 би, то сеть считается работоспособной.

2.2.2.2. Пример расчёта конфигурации сети Fast Ethernet

На рис. 2.1 приведён пример одной из предельно допустимых конфигураций сети Fast Ethernet.

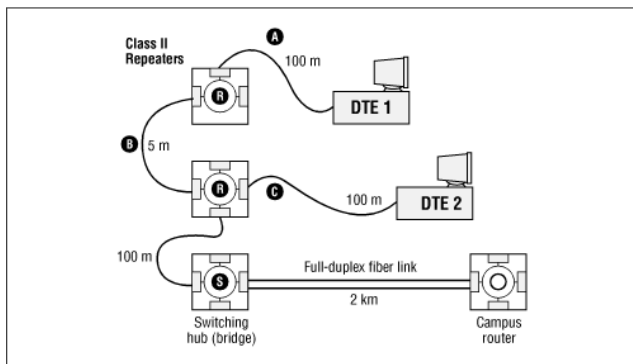


Рис. 2.1. Пример допустимой конфигурации сети Fast Ethernet

Диаметр домена коллизий вычисляется как сумма длин сегментов А (100 м), В (5 м) и С (100 м) и равен 205 м. Длина сегмента, соединяющего повторители, может быть более 5 м, если при этом диаметр домена коллизий не превышает допустимый для данной конфигурации предел. Коммутатор (switching hub), входящий в состав сети, изображённой на рис. 2.1, считается окончательным устройством, так как коллизии через него не распространяются. Поэтому 2-километровый сегмент оптоволоконного кабеля, соединяющий этот коммутатор с маршрутизатором (router), не учитывается при расчёте диаметра домена коллизий сети Fast Ethernet. Сеть удовлетворяет правилам первой модели.

Проверим теперь её по второй модели. Наихудшие пути в домене коллизий: от DTE1 к DTE2 и от DTE1 к коммутатору (switching hub). Оба пути состоят из трёх сегментов на витой паре, соединённых двумя повторителями класса II. Два сегмента имеют предельно допустимую длину 100 м. Длина сегмента, соединяющего повторители, равна 5 м.

Предположим, что все три рассматриваемых сегмента являются сегментами 100BASE-TX и в них используется витая пара категории 5. В табл. 2.3 приведены величины времени двойного оборота для рассматриваемых путей. Сложив числа из второго столбца этой

таблицы, получим 511,96 би – это и будет время двойного оборота для наихудшего пути.

Таблица 2.3

Время двойного оборота сети рис. 2.1

Компонент пути	Время двойного оборота, би
Пара терминалов с интерфейсами TX	100
Сегмент на витой паре категории 5 (100 м)	111,2
Сегмент на витой паре категории 5 (100 м)	111,2
Сегмент на витой паре категории 5 (5 м)	5,56
Повторитель класса II	92
Повторитель класса II	92

Следует заметить, что в данном случае нет страхового запаса в 4 би, так как в этом примере используются наихудшие значения задержек, приведённые в табл. 2.2. Реальные временные характеристики компонентов Fast Ethernet могут отличаться в лучшую сторону.

2.3. Задание для выполнения

Требуется оценить работоспособность 100-мегабитной сети Fast Ethernet в соответствии с первой и второй моделями.

Конфигурации сети приведены в табл. 2.4. Топология сети представлена на рис. 2.2–2.3.

Таблица 2.4

Варианты заданий

No	Сегмент 1	Сегмент 2	Сегмент 3	Сегмент 4	Сегмент 5	Сегмент 6
1.	100BASE-TX, 100 м	100BASE-TX, 95 м	100BASE-TX, 80 м	100BASE-TX, 5 м	100BASE-TX, 100 м	100BASE-TX, 100 м
2.	100BASE-TX, 15 м	100BASE-TX, 5 м	100BASE-TX, 5 м	100BASE-FX, 400 м	100BASE-TX, 10 м	100BASE-TX, 4 м
3.	100BASE-TX, 60 м	100BASE-TX, 95 м	100BASE-TX, 10 м	100BASE-TX, 10 м	100BASE-TX, 90 м	100BASE-TX, 95 м

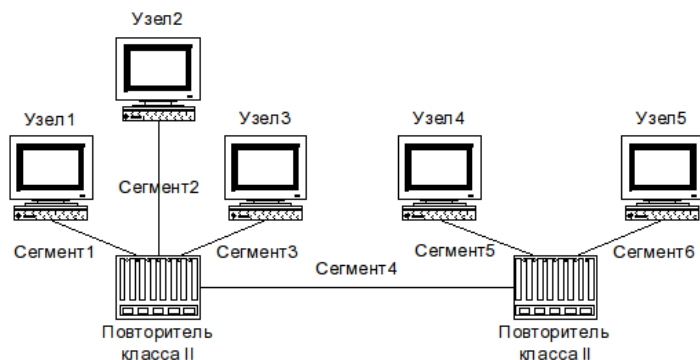


Рис. 2.2. Топология сети 1

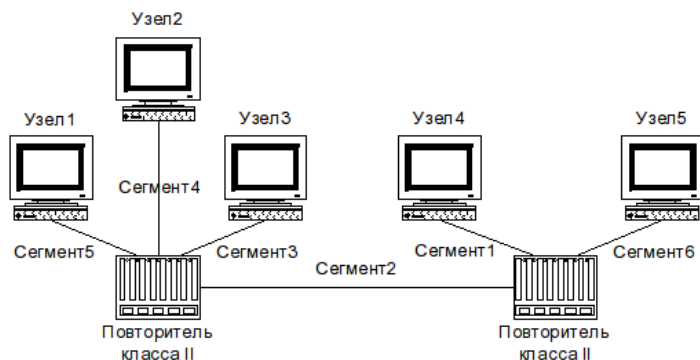


Рис. 2.3. Топология сети 2

Глава 3. Знакомство с Packet Tracer. Моделирование простой сети

3.1. Цели работы

Изучение принципов построения сетей передачи данных и принципов настройки сетевого оборудования.

3.2. Предварительные сведения. Запуск, настройка, краткое описание интерфейса Packet Tracer

Packet Tracer — симулятор сети передачи данных, выпускаемый фирмой Cisco Systems.

С помощью данного симулятора можно строить модели сетей передачи данных, изучать настройки и принципы функционирования сетевого оборудования производителя, проводить диагностику работоспособности моделируемой сети.

Для запуска Packet Tracer, установленного под ОС Linux, достаточно в командной строке терминала ввести команду
`packettracer &`

Рабочее пространство Packet Tracer представлено на рис. 3.1.

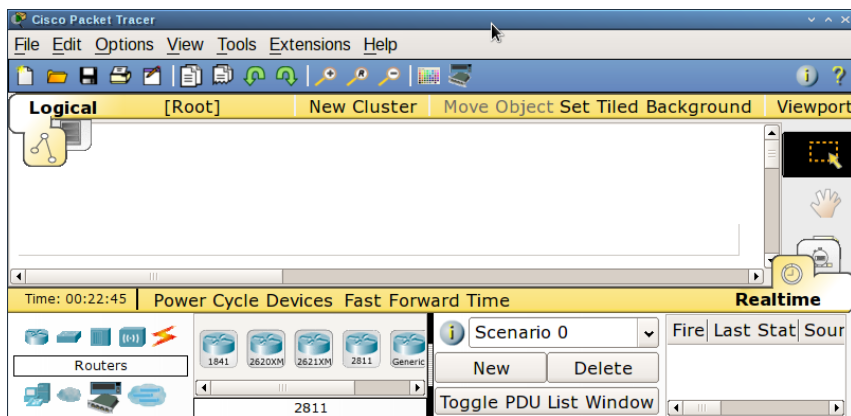


Рис. 3.1. Рабочее пространство Packet Tracer

По умолчанию интерфейс Packet Tracer — английский. Для изменения его на русский необходимо в меню выбрать «Options», «Preferences», вкладку «Interface» и указать язык «RUSSIANxx.ptl», где xx обозначает версию симулятора (рис. 3.2). Затем нужно нажать «Change Language» и перезапустить симулятор.

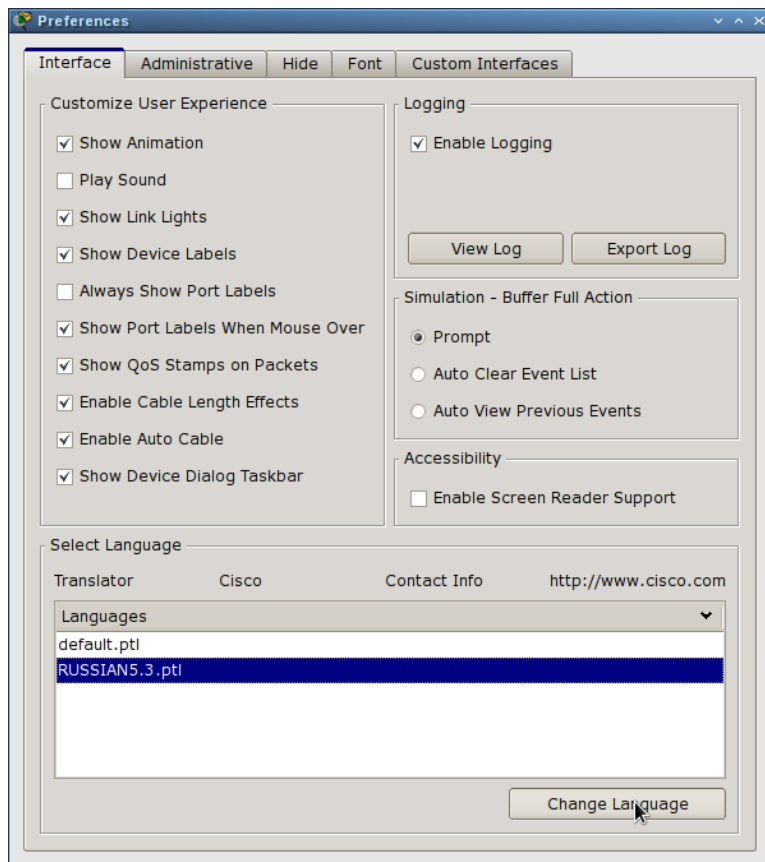


Рис. 3.2. Изменения языка интерфейса Packet Tracer

Основное окно программы (рис. 3.3) содержит программное меню (1), правое графическое окно с пиктограммами инструментов для работы с проектом и его объектами (2), меню выбора типа устройства (3), меню выбора устройства определённого типа (4).

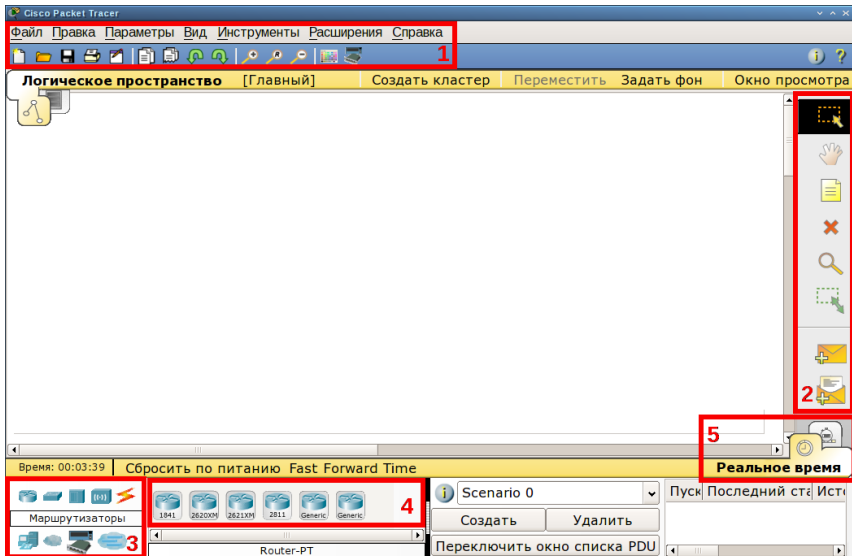


Рис. 3.3. Меню Packet Tracer

Меню (1) позволяет создать, открыть, сохранить или распечатать проект, скопировать и вставить элемент, масштабировать рабочее пространство проекта.

Меню (2) содержит инструменты выделения одного или нескольких объектов проекта, прокрутки проекта, добавления подписи к элементу проекта, удаления элемента проекта. Инструмент, напоминающий лупу, используется для просмотра содержимого таблиц ARP, NAT, таблицы маршрутизации и т.д. После этого инструмента расположены инструмент изменения размеров объекта, инструменты для эмуляции отправки с последующим отслеживанием произвольного пакета данных внутри проекта.

В меню (3) можно выбрать тип устройства: концентратор, коммутатор, маршрутизатор, тип соединения, оконечные и пользовательские устройства.

В меню (4) в зависимости от выбранного в меню (3) типа оборудования можно выбрать конкретное устройство.

Вкладка (5) позволяет переключаться с режима работы в реальном времени в режим симуляции и бывает полезна, если нужно более детально изучить, например, движение передаваемых от устройства к устройству данных, форматы конкретных пакетов.

3.3. Моделирование простейшей сети. Протокол ICMP

3.3.1. Постановка задачи

Требуется построить топологию сети из двух коммутаторов и четырёх пользовательских устройств (компьютеров), подсоединённых по два к каждому коммутатору. На пользовательских узлах нужно задать статическую адресацию из одного адресного пространства. Затем требуется изучить работу протокола ICMP.

3.3.2. Порядок выполнения работы

Создать новый проект (например, lab03-01.pkt).

В рабочем пространстве разместить 2 коммутатора, например, Cisco-2950, 4 оконечных устройства (рис. 3.4). Подключить оконечные устройства к коммутаторам, используя прямое соединение, и коммутаторы между собой, используя кроссовое соединение.

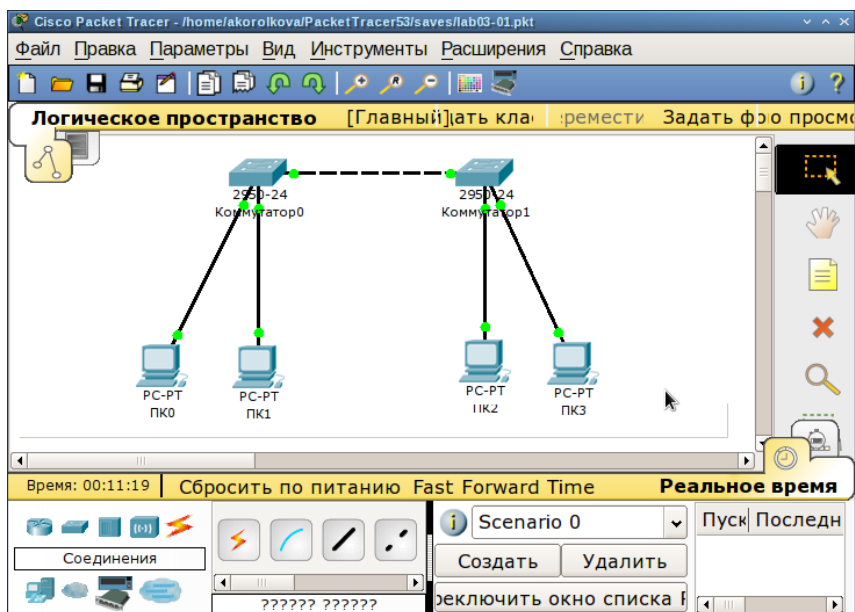


Рис. 3.4. Модель простой сети

Щёлкнув последовательно на каждом оконечном устройстве, задайте статический адрес из диапазона 192.168.1.2 – 192.168.1.5 (рис. 3.5).

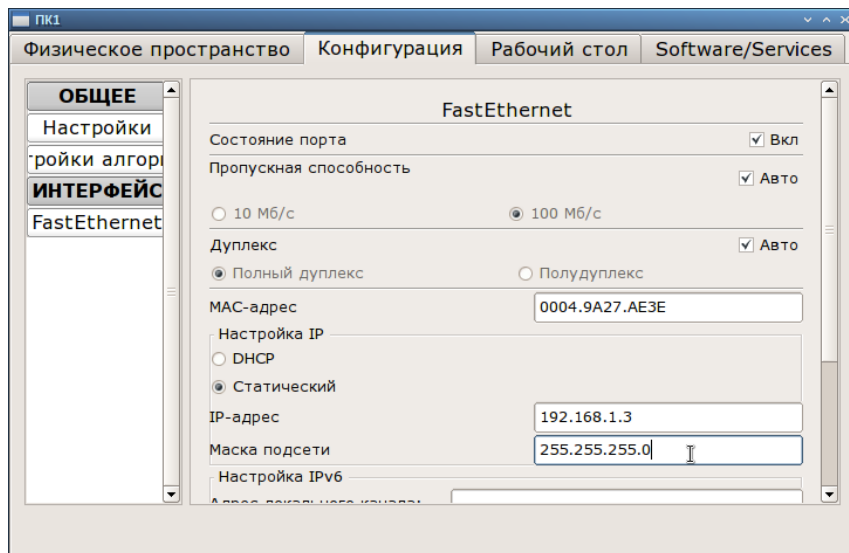


Рис. 3.5. Статическая адресация на оконечном устройстве

На одном из оконечных устройств запустите командную строку (рис. 3.6) и с помощью команды `ipconfig /all` посмотрите его сетевые настройки (рис. 3.7). Проверьте доступность другого узла сети с помощью команды `ping`.

В основном окне проекта перейдите из режима реального времени в режим симуляции. В командной строке одного из оконечных устройств повторите команду `ping`. С помощью кнопки «Захват / Вперёд» проследите движение пакета ICMP от одного оконечного устройства к другому (рис. 3.8).

Щёлкнув на значке пакета, откройте окно информации о PDU на устройстве и изучите его детали (рис. 3.9, 3.10). Используя кнопку «Проверь себя», ответьте на вопросы. Прodelайте эту операцию на каждом этапе следования пакета и дайте пояснения об изменениях. Изучите информацию о PDU, передаваемых между коммутаторами (рис. 3.11, 3.12). Изучите изменения формата передаваемых пакетов других протоколов (например, telnet).

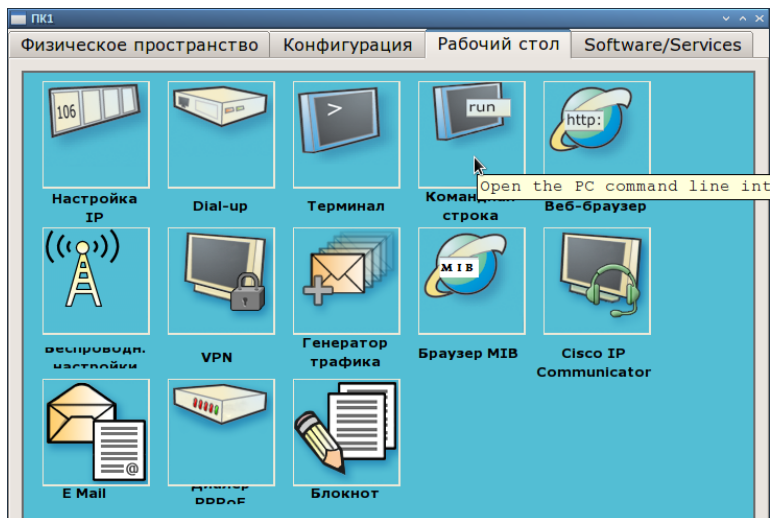


Рис. 3.6. Запуск командной строки на оконечном устройстве

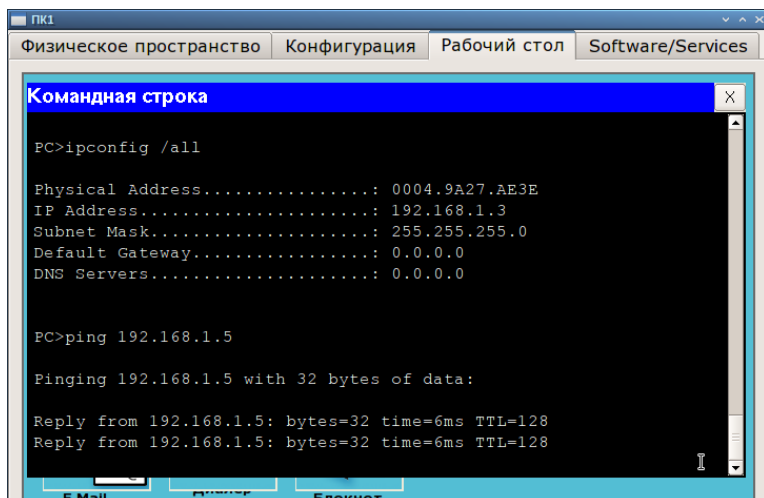


Рис. 3.7. Выполнение команды ipconfig на оконечном устройстве

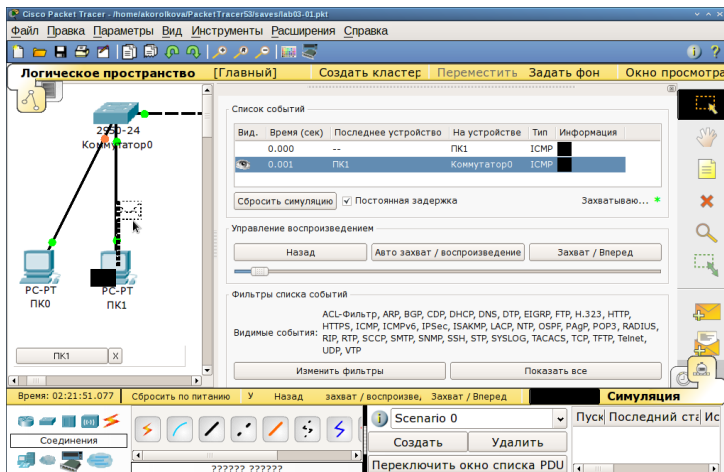


Рис. 3.8. Выполнение команды ping в режиме симуляции

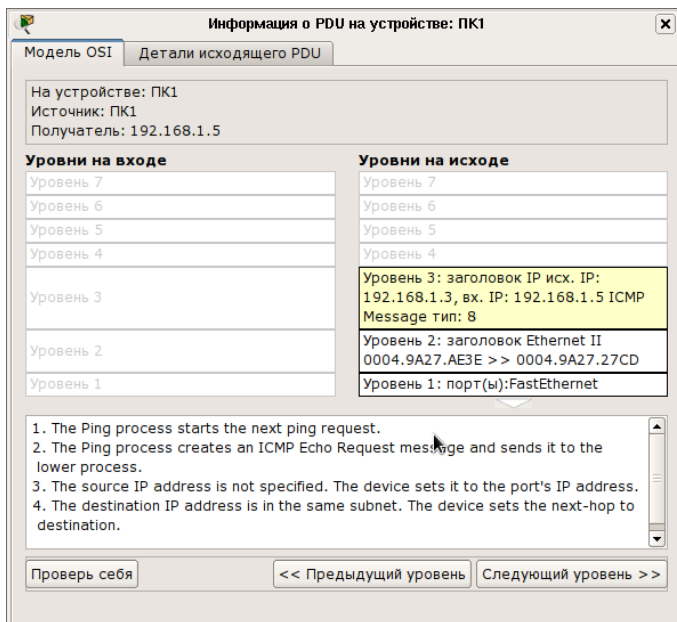


Рис. 3.9. Информация о PDU: уровень OSI

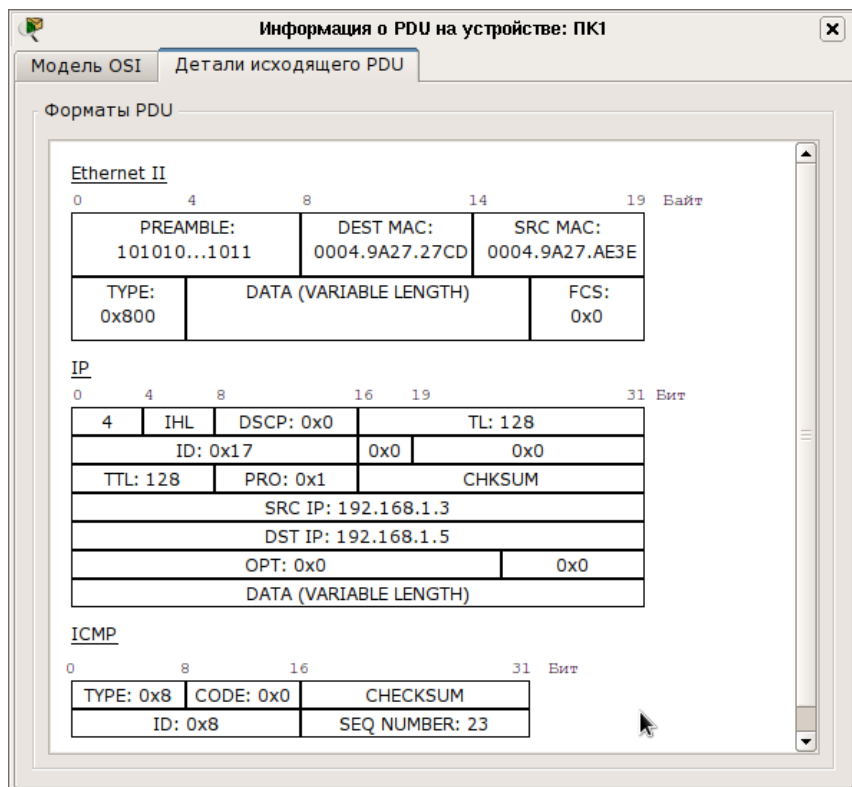


Рис. 3.10. Информация о PDU: форматы пакетов

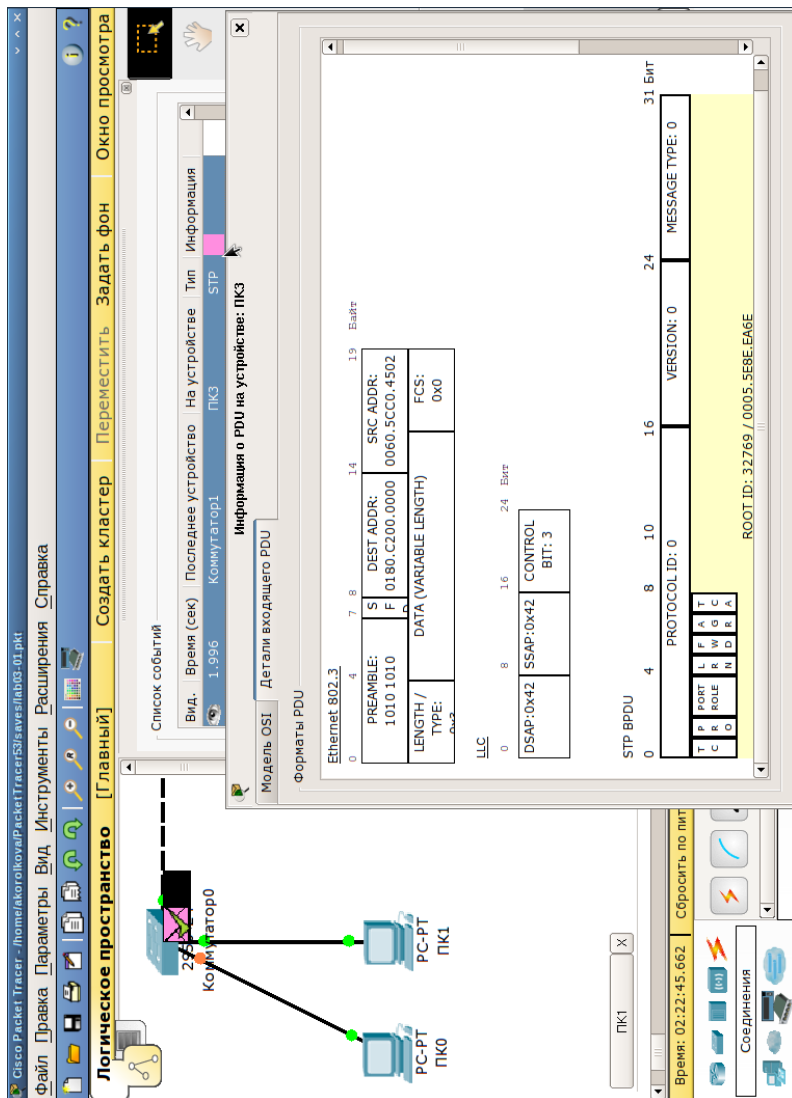


Рис. 3.11. Информация о PDU протокола STP

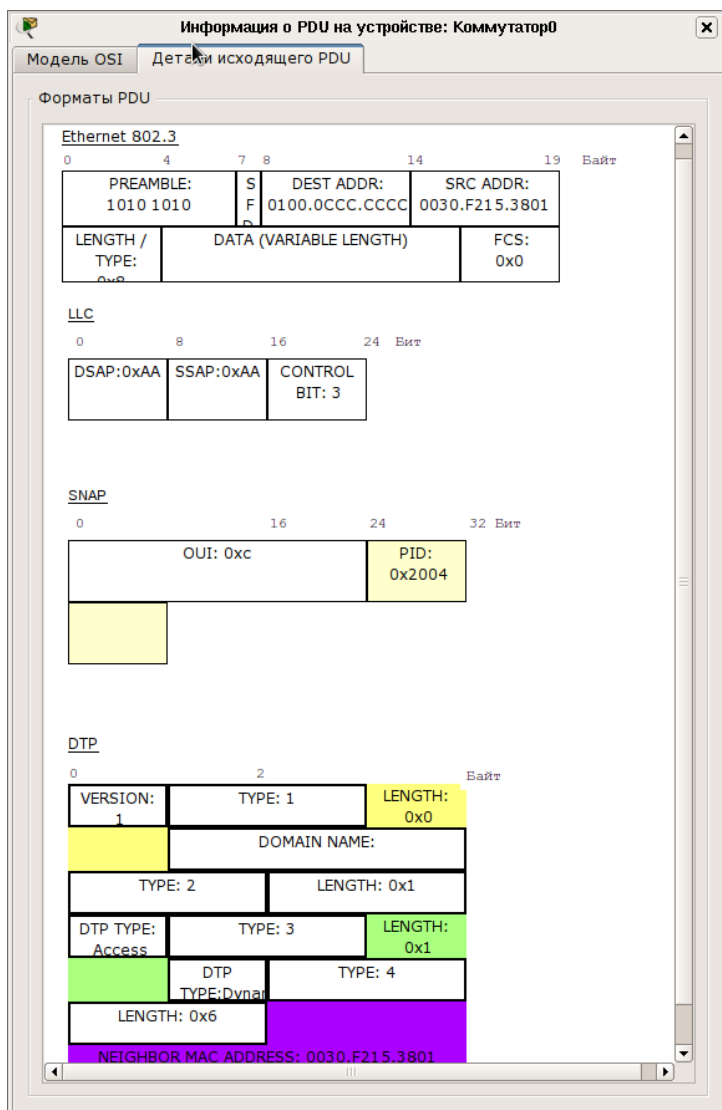


Рис. 3.12. Информация о PDU, передаваемых между коммутаторами

3.4. Конфигурирование оборудования Cisco

Для конфигурирования оборудования Cisco предусмотрен консольный кабель, на одном конце которого имеется разъём RJ45, подключаемый в соответствующий порт на оборудовании Cisco, а на другом — RS-232, подключаемый к COM-порту компьютера, с которого будет производиться настройка. Для управления через COM-порт в Windows системах можно использовать HyperTerminal, а в Linux, например, minicom.

После подключения оборудования и включения питания на экране появляется приглашение пользовательского режима (отображается символом >), например

```
Router>
```

или

```
Switch>
```

В Packet Tracer соответственно, чтобы войти в режим конфигурирования оборудования, необходимо разместить это оборудование в рабочем пространстве, щёлкнуть на нём и перейти на вкладку CLI (рис. 3.13) командного интерфейса IOS (Internetwork Operating System).

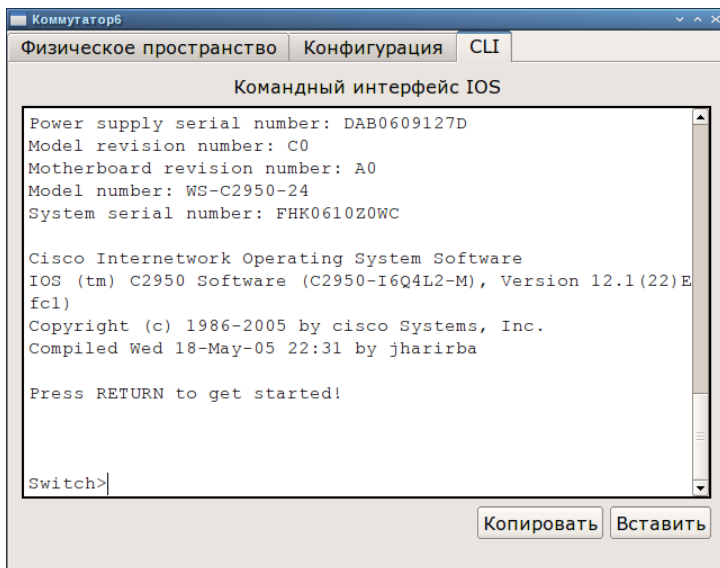


Рис. 3.13. Командный интерфейс IOS в Packet Tracer

3.4.1. Основные команды конфигурирования оборудования Cisco

Пользовательский режим на оборудовании Cisco предназначен только для просмотра конфигурации устройства и ввода простейших команд диагностики, например `ping`.

Для просмотра доступных команд используется команда `?` знака вопроса и нажатие клавиши `Enter` (рис. 3.14).

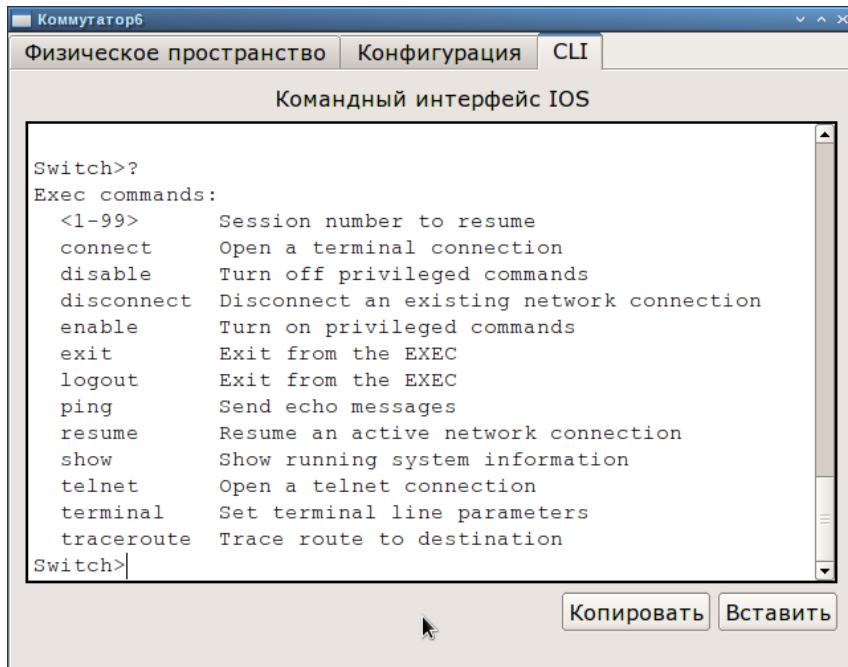


Рис. 3.14. Вызов списка команд, доступных в пользовательском режиме

Чтобы перейти в привилегированный режим и иметь возможность настраивать оборудование, вводится команда `enable` и приглашение заменяется на `#`, например:

Router#

или

Switch#

Список доступных в привилегированном режиме команд продемонстрирован на рис. 3.15 для коммутаторов и на рис. 3.16 для маршрутизаторов.

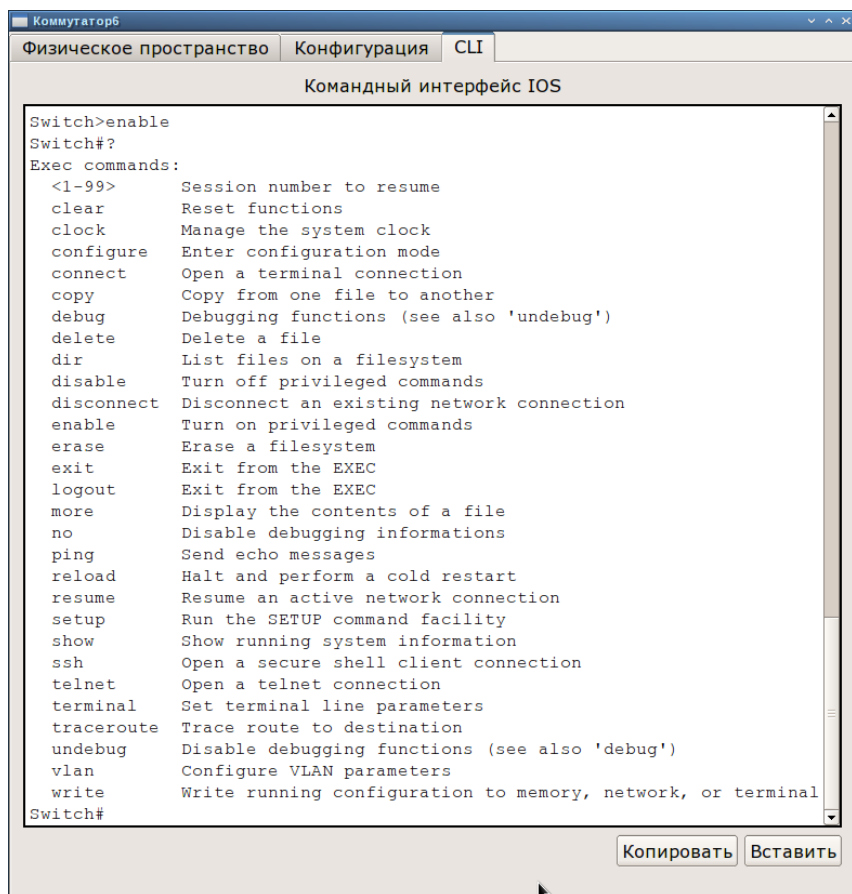


Рис. 3.15. Вызов списка команд коммутатора, доступных в привилегированном режиме

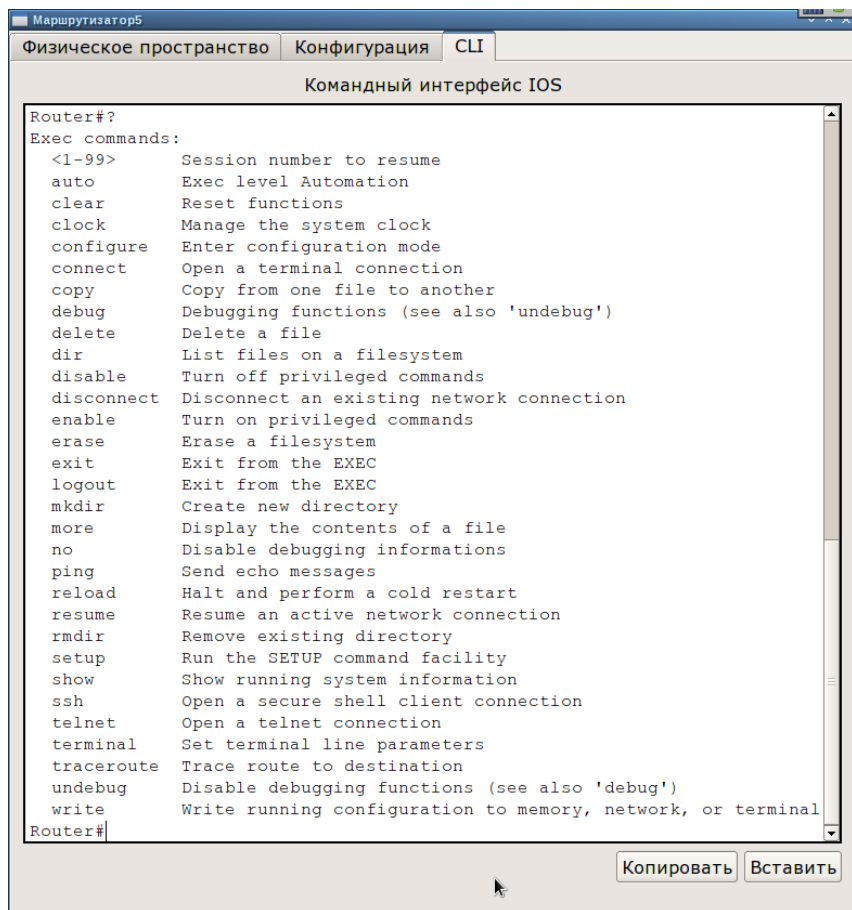


Рис. 3.16. Вызов списка команд маршрутизатора, доступных в привилегированном режиме

Для упрощения администрирования в устройствах Cisco применяется концепция VLAN-доменов.

VLAN (Virtual Local Area Network) — логическая («виртуальная») локальная компьютерная сеть, представляющая собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широкополосному домену, независимо от их физического местонахождения.

VLAN можно назначить:

- определённому порту устройства;
- определённому MAC-адресу;
- определённому протоколу;
- по данным аутентификации пользователя или устройства.

По умолчанию на каждом порту устройства Cisco имеется сеть VLAN1 или VLAN управления. Сеть управления не может быть удалена, однако могут быть созданы дополнительные сети VLAN и этим альтернативным VLAN могут быть дополнительно назначены порты.

Операционная система IOS понимает как укороченное написание команд, так и их дописывание. Клавиша **Tab** автоматически дополняет команду, которую вы начали писать, что позволяет существенно уменьшить количество опечаток и повысить скорость работы. Клавиша **Tab** дополнит команду, а знак вопроса выведет на экран список дальнейших возможных действий, а также небольшое описание к ним.

Рассмотрим некоторые общие и часто используемые команды:

- **show** — просмотр параметров устройства:
 - **show running-config** — просмотр текущей конфигурации настраиваемого оборудования;
 - **show startup-config** — просмотр стартовой конфигурации;
 - **show mac-address-table** — просмотр таблицы MAC-адресов;
- **configure terminal** — переход в режим конфигурации и настройки:
 - **hostname имя** — задать имя устройства;
 - **service password-encryption** — включить режим хранения паролей в файле конфигурации устройства в зашифрованном виде;
 - **line console номер** — перейти в настройку консольного подключения:
 - **login** — разрешение проверки пароля;
 - **password пароль** — задать пароль;
 - **line vti 0 4** — настройка определённых виртуальных терминалов (в данном случае с 0 до 4 терминала):
 - **login** — разрешение проверки пароля;
 - **password пароль** — задать пароль;
 - **enable password пароль** — назначить пароль привилегированного уровня;
 - **service password-encryption** — зашифровать систему паролей;

- **interface** тип/номер — настройка определённого интерфейса (например, **interface fastEthernet0/1**):
 - **ip address** ip-адрес маска — задать ip-адрес и маску;
 - **no shutdown** — поднять интерфейс;
 - **description** текст — задать текстовое пояснение назначения интерфейса;
- **vlan** номер — настройка определённого vlan;
- **interface vlan** номер — настройка определённого vlan на интерфейсах (например, **interface vlan 1**):
 - **ip address** адрес маска — задать ip-адрес и маску на интерфейсе vlan;
 - **no shutdown** — поднять интерфейс vlan;
 - **description** текст — задать текстовое пояснение назначения интерфейса vlan;
- **exit** — выход из режима конфигурирования;
- **write memory** — запись внесённых изменений в память оборудования;
- **disable** — выход из привилегированного режима в пользовательский;
- **logout** — выйти из режима консоли.

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.

Если, например, на коммутаторе 2-го уровня требуется задать ip-адрес, по которому можно обратиться к нему, то это можно реализовать, задав адрес на VLAN1:

```
Switch> enable
Switch# configure terminal
Switch(config)# interface vlan 1
Switch(config-if)# ip address ip-адрес маска
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# exit
Switch# write memory
```

Для проверки связи между устройствами сети можно использовать команду **ping**. Результаты выполнения команды **ping** представлены в табл. 3.1.

Таблица 3.1

Результаты выполнения команды ping

Символ	Значение
!	успешный приём эхо-ответа
.	превышено время ожидания
U	пункт назначения недостижим
C	перегрузка сети
I	выполнение команды прервано администратором
?	неизвестный тип пакета
&	пакет превысил значение параметра времени жизни TTL пакета

3.4.2. Задание

Схема моделируемой сети представлена рис. 3.17.

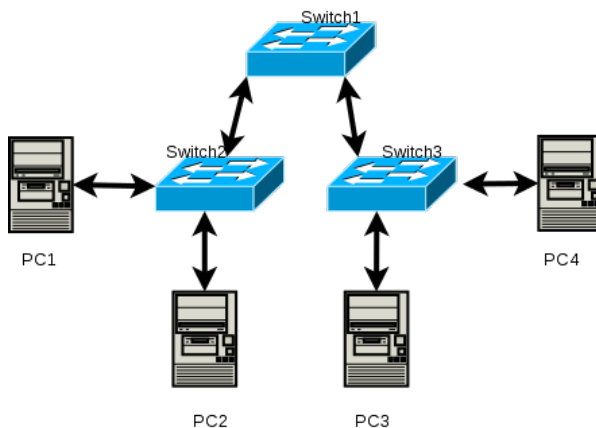


Рис. 3.17. Схема моделируемой сети

- Изменить имя коммутаторам Cisco.
- Обеспечить парольный доступ к привилегированному режиму на коммутаторах.

- Задать ip-адреса и маски коммутаторам (172.16.1.11/24, 172.16.1.12/24, 172.16.1.13/24).
- Задать ip-адреса и маски сетей персональным компьютерам (172.16.1.1/24, 172.16.1.2/24, 172.16.1.3/24, 172.16.1.4/24).
- Убедиться в достижимости всех объектов сети по протоколу IP.
- Переключившись в режим симуляции (его описание приведено в методических указаниях к предыдущей лабораторной работе) рассмотреть и пояснить процесс обмена данными по протоколу ICMP между устройствами (выполнив команду `ping` с одного компьютера на другой), пояснить роль протокола ARP в этом процессе. Детальное пояснение включить в отчёт.
Структура отчета по работе:
- Титульный лист.
- Задание.
- Схема сети.
- Ход работы: раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) и копий экранов (должна быть видна набранная команда и реакция системы, если она есть).
- Выводы.

Глава 4. Packet Tracer. Настройка маршрутизаторов

4.1. Цели работы

Изучение принципов маршрутизации в IP-сетях и принципов настройки сетевого оборудования.

4.2. Предварительные сведения

4.2.1. Понятия DTE и DCE

Оконечное оборудование данных (Data Terminal Equipment, DTE) — термин для обозначения устройства, обеспечивающего интерфейс с пользователем. Обычно таким оборудованием является терминал или ЭВМ. На DTE исполняются пользовательские прикладные программы.

Оконечное оборудование канала данных (Data Circuit-terminating Equipment, DCE) обеспечивает подключение DTE к связному каналу для передачи преобразования и усиления сигнала, сгенерированного DTE.

Маршрутизатор с последовательным интерфейсом можно считать как устройством DCE (маршрутизатор только транслирует данные на сетевом уровне), так и устройством DTE (цепочки бит на выходе маршрутизатор сам генерирует и сам обрабатывает принятые данные). Соответственно, корректнее говорить о типе портов (DTE/DCE), при этом имея в виду следующие соглашения:

- для порта DTE сигнал, обозначаемый как TxD (данные передатчика), является выходным, сигнал RxD (данные приемника) — входным;
- для порта DCE сигнал, обозначаемый как TxD, является входным (устройство должно передать данные в канал связи), а сигнал RxD — выходным (данные из канала, которые должны попасть на вход приемника DTE).

Из соглашений следует, что при соединении порта DTE с портом DCE одноименные сигнальные цепи должны соединяться через последовательное соединение точка-точка. При соединении однотипных портов (DTE-DTE или DCE-DCE) сигнальные цепи должны соединяться перекрестно (TxD-RxD, RxD-TxD), а управляющие — в соответствии с логикой протокола.

Для асинхронного режима передачи посимвольная синхронизация осуществляется старт-битами и внешняя подача синхронизации

не требуется (скорости и допуски отклонения согласованы, и любое устройство имеет свой внутренний генератор). В синхронном режиме все последовательные интерфейсы используют внешние сигналы синхронизации, причем отдельные для передаваемых ST (Send Timing, Send Clock) и принимаемых RT (Receive Timing, Receive Clock) данных.

Для устройств DTE сигналы синхронизации, по которым работают сдвигающие регистры их приемников и передатчиков, являются входными. Источником синхронизации (сигнал RT) для принимаемых данных (RxD) практически однозначно является DCE (синхросигнал обычно выделяется из самосинхронизирующего сигнала линии связи).

Если синхронный канал сам навязывает свою синхронизацию, то для DTE первичным источником синхронизации будет DCE. DCE может потребовать и внешней синхронизации. Для этих целей порт DTE снабжают дополнительным выходом ТТ (Terminal Timing), от которого может синхронизироваться DCE. Но сам передатчик DTE будет синхронизироваться от входа ST, на который сигнал поступит либо от DCE, либо по перемычке в разъеме. Возможен вариант конфигурирования, когда сигнал ТТ заводится на передатчик внутри порта (не с разъема). В высокоскоростном интерфейсе для синхронных каналов (HSSI) источником синхронизации является только DCE, а сигнал ТТ является буферизованным принятым сигналом ST. Такой «разворот» синхросигнала позволяет учесть задержки распространения сигнала в интерфейсном кабеле (сигналы TxD и ТТ идут параллельными путями, и фазовый перекокс между ними будет небольшим). Для компенсации перекокса при высоких скоростях может применяться и инверсия сигнала синхронизации, которую включают при настройке порта.

В названиях сигналов синхронизации передатчика бывает путаница, и внешним (external) сигналом синхронизации называют как сигнал от DTE, так и сигнал, по которому DTE в действительности синхронизирует свой приемник. Понять, что есть что, позволяет указание на источник (DTE или DCE), т.е., если источником является DCE, то DTE синхронизирует свой приёмник по сигналу от DCE, который и является в этом случае внешним.

4.2.2. Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) позволяет устройствам обмениваться основной конфигурационной информацией. CDP работает на втором (канальном) уровне модели OSI, поэтому он не является маршрутизируемым протоколом и работает только с непосредственно подключенными устройствами. Протокол CDP связывает физическую среду передачи данных более низкого уровня с протоколами более высокого сетевого уровня. Поэтому устройства, поддерживающие разные протоколы третьего уровня, могут узнавать друг друга.

При запуске устройства протокол CDP запускается автоматически, после чего он может автоматически определить соседние устройства, на которых также работает протокол CDP, и выдать информацию, например, об идентификаторах устройств и портов, аппаратной платформе.

4.2.3. Команда `traceroute`

Команды `traceroute` показывает адреса промежуточных интерфейсов (хопов) на пути пакетов в пункт назначения.

Для определения промежуточных маршрутизаторов `traceroute` отправляет узлу назначения серию ICMP-пакетов (по умолчанию 3 пакета), с каждым шагом увеличивая значение поля TTL («время жизни») на 1. Это поле обычно указывает максимальное количество маршрутизаторов, которое может быть пройдено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и поэтому первый же маршрутизатор возвращает обратно ICMP-сообщение «time exceeded in transit», указывающее на невозможность доставки данных. Traceroute фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся на монитор компьютера). Затем `traceroute` повторяет отправку серии пакетов, но уже с TTL, равным 2, что заставляет первый маршрутизатор уменьшить TTL пакетов на единицу и направить их ко второму маршрутизатору. Второй маршрутизатор, получив пакеты с TTL=1, также возвращает «time exceeded in transit». Процесс повторяется до тех пор, пока пакет не достигнет узла назначения. При получении ответа от этого узла процесс трассировки считается завершённым.

На конечном хосте IP-дейтаграмма с TTL = 1 не отбрасывается и не вызывает ICMP-сообщения типа «срок истёк», а должна быть отдана приложению. Достижение пункта назначения определяется следующим образом: отсылаемые `traceroute` дейтаграммы содержат UDP-пакет с заведомо неиспользуемым номером порта на адресуемом хосте. Номер порта будет равен $33434 + (\text{максимальное количество транзитных участков до узла}) - 1$. В пункте назначения UDP-модуль, получая подобные дейтаграммы, возвращает ICMP-сообщения об ошибке «порт недоступен». Таким образом, чтобы узнать о завершении работы, программе `traceroute` достаточно обнаружить, что поступило ICMP-сообщение об ошибке этого типа.

4.2.4. Address Resolution Protocol

Address Resolution Protocol (ARP) — протокол сетевого уровня, предназначенный для определения MAC-адреса по известному IP-адресу.

Типы сообщений ARP: запрос ARP (ARP request) и ответ ARP (ARP reply). Система-отправитель при помощи запроса ARP запрашивает физический адрес системы-получателя. Ответ (физический адрес узла-получателя) приходит в виде ответа ARP.

Отправитель определяет IP-адрес приёмника, просматривает свою ARP таблицу и определяет MAC-адрес приёмника. Если MAC- и IP-адреса приёмника присутствуют в ARP-таблице отправителя, то между ними устанавливается соответствие и приёмник использует его в ходе инкапсуляции IP-пакетов во фреймы канального уровня. MAC-адреса фреймов канального уровня берутся из ARP-таблиц. После этого фрейм по физическому каналу отправляется от отправителя к адресату.

Если записи в кэше ARP нет, то выполняется широковещательный запрос ARP. Запрос принимают все сетевые устройства в сегменте сети, но только устройство, имеющее запрашиваемый IP-адрес, реагирует на него, посылая отправителю информацию о MAC-адресе своего сетевого интерфейса со своим IP-адресом. Отправитель записывает MAC-адрес и IP-адрес в свою ARP-таблицу.

4.3. Моделирование сети со статической маршрутизацией

4.3.1. Постановка задачи

1. Построить топологию сети из трёх маршрутизаторов (рис. 4.1). Соединение между первым и вторым маршрутизаторами — Ethernet, соединение между первым и третьим маршрутизаторами — последовательное соединение точка-точка (serial cable), причём первый маршрутизатор должен выполнять функции DCE-устройства, т.е. задавать синхронизацию, с тактовой частотой 64 Кбит/с, а третий маршрутизатор — DTE.

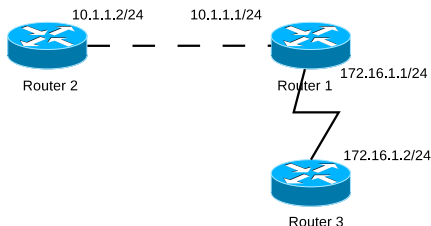


Рис. 4.1. Модель сети

2. Задать имена маршрутизаторам и IP-адреса их интерфейсам. Первый маршрутизатор (Router1) должен иметь 2 адреса: 10.1.1.1/24, 172.16.10.1/24. Второй маршрутизатор (Router2) должен иметь адрес 10.1.1.2/24, третий (Router3) — 172.16.10.2/24 (см. рис. 4.1).
3. Изучите состояния всех интерфейсов, на которых работает CDP.
4. На Router1 настройте возможность работы по протоколу telnet. С Router2 зайдите по на Router1 по telnet. Выведите информацию о подключённых на Router1 пользователях. На Router2 выведите информацию о запущенных сессиях, возобновите telnet-сессию, а затем закройте её.
5. Настройте сначала статическую маршрутизацию, а затем статическую маршрутизацию по умолчанию с Router2 на Router3 и с Router3 на Router2.
6. Оформите отчёт, зафиксировав в нём производимые вами действия.

4.3.2. Порядок выполнения работы

Создайте новый проект (например, lab04-01.pkt).

В рабочем пространстве разместите 3 маршрутизатора типа Generic и соедините их согласно требованиям задания (рис. 4.2).

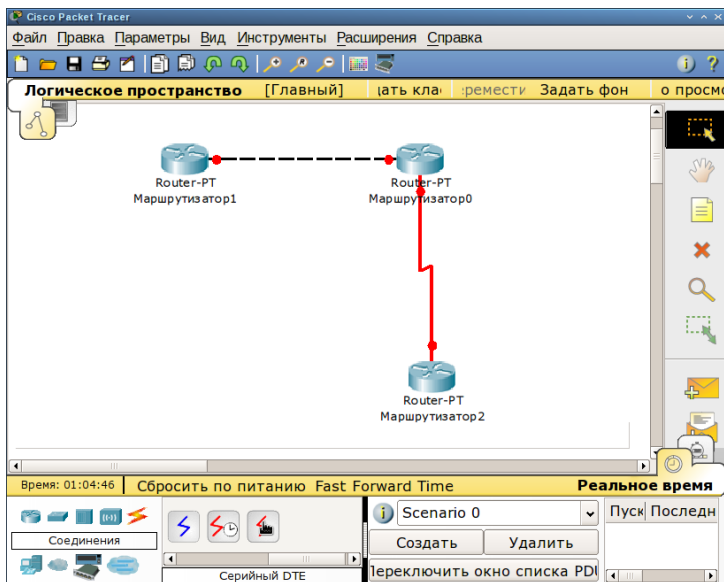


Рис. 4.2. Модель сети: 3 маршрутизатора типа Generic

Переименуйте маршрутизаторы, назвав их Router1, Router2, Router3. Для этого при конфигурации каждого маршрутизатора используйте следующую последовательность команд (например, для маршрутизатора 0):

```
Continue with configuration dialog? [yes/no]: no
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#exit
Router1#write memory
```

Соответственно, во вкладке «Конфигурация» замените «Отображаемое имя» (рис. 4.3).

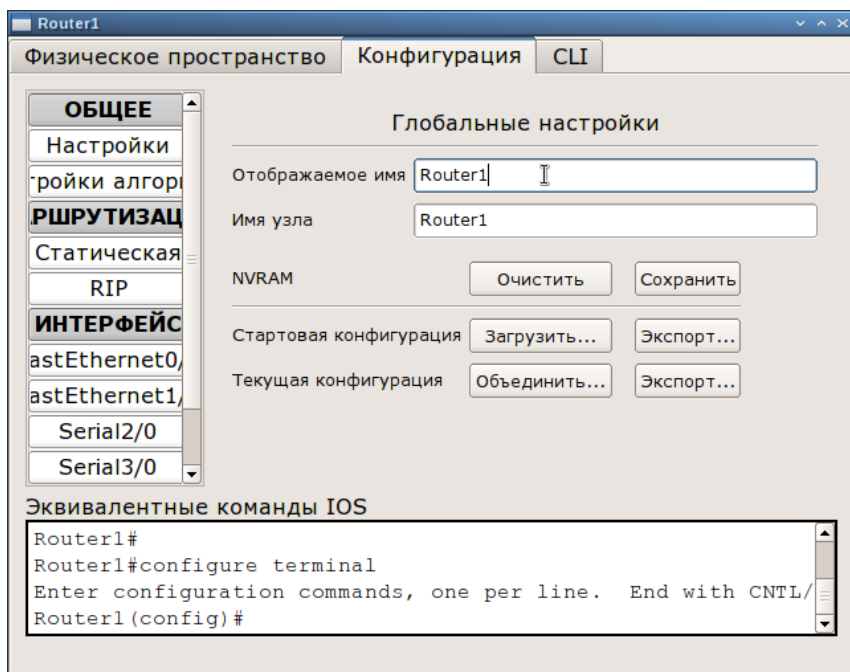


Рис. 4.3. Замена отображаемого имени маршрутизатора на вкладке «Конфигурация»

С помощью значка боковой панели «Сделать пометку» обозначьте планируемое распределение адресного пространства сети. В результате получите следующую топологию сети (рис. 4.4).

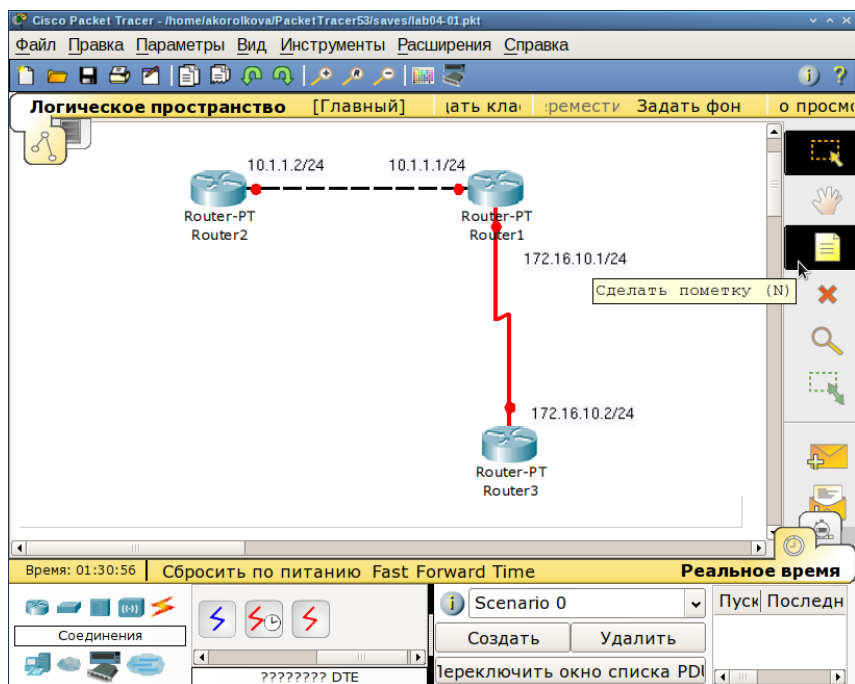


Рис. 4.4. Модель сети с переименованными маршрутизаторами

Далее необходимо в соответствии с заданием настроить интерфейсы Ethernet маршрутизаторов, подняв их и задав им описание. Например, для Router1:

```
Router1#configure terminal
Router1(config)#interface f0/0
Router1(config-if)#no shutdown
Router1(config-if)#description Ethernet on Router1
Router1(config-if)#exit
Router1#write memory
Router1#show running-config
```

По аналогии настройте интерфейс на Router2.

Далее необходимо в соответствии с заданием настроить интерфейсы последовательного соединения маршрутизаторов, подняв их и задав им описание и тактовую частоту на DCE в 64 Кбит/с. Например, для Router1 (по заданию является DCE):

```
Router1#show controllers s2/0
```

```
Router1#configure terminal
Router1(config)#interface s2/0
Router1(config-if)#clock rate 64000
Router1(config-if)#no shutdown
Router1(config-if)#description Serial on Router1
Router1(config-if)#exit
Router1(config)#exit
Router1#write memory
Router1#show running-config
```

По аналогии настройте интерфейс на Router3, учитывая, что он является DTE (тактовую частоту задавать не требуется).

На Router1 введите команду для вывода состояния всех интерфейсов, на которых работает CDP:

```
Router1#show cdp interface
```

и убедитесь, что интерфейсы f0/0 и s2/0 подняты и посылают CDP-пакеты.

Затем получите краткую и полную информацию о непосредственно подключённых к Router1 устройствах:

```
Router1#show cdp neighbors
Router1#show cdp neighbors detail
```

Получите информацию об устройстве Router3:

```
Router1#show cdp entry Router3
```

Задайте ip-адреса маршрутизаторам и с помощью команд `ping` и `tracert` проверьте доступность всех маршрутизаторов. Убедитесь, что с Router3 команды покажут недоступность адресов 10.1.1.1 и 10.1.1.2.

На Router1 настройте возможность работы по протоколу telnet. Необходимо, чтобы сетевое устройство принимало telnet-сессии и было защищено паролем. Каждая так называемая линия (line vty) в сетевом устройстве потенциально представляет активную telnet-сессию, которую устройство может поддерживать. Наши сетевые устройства поддерживают до 5 линий, назначенные на виртуальные терминалы vty:

```
Router1#configure terminal
Router1(config)#line vty 0 4
Router1(config-line)#password Cisco
Router1(config-line)#login
Router1(config-line)#exit
Router1(config)#service password-encryption
Router1(config)#exit
Router1#write memory
```

C Router2 зайдите на Router1 по telnet:

```
Router2>telnet 10.1.1.1
```

Затем выведите информацию о подключённых на Router1 пользователях, введя

```
Router1#show users
```

Нажмите одновременно клавиши control-shift-6, потом отпустите и сразу нажмите клавишу x. Имя сетевого устройства должно поменяться на Router2. Выведите информацию о запущенных сессиях:

```
Router2>show sessions
```

Вернитесь в telnet-сессию на Router1, используя клавиши control-shift-6 и 1 вместо x. Посмотрите конфигурацию Router1.

Вернитесь на Router2 и возобновите telnet-сессию, а затем закройте её:

```
Router2>resume 1
[Resuming connection 1 to 10.1.1.1 ... ]
```

```
Router1>
```

```
Router2>disconnect 1
```

Настройте статическую маршрутизацию.

Для просмотра таблицы маршрутов следует использовать команду **show ip route**. Маршрут на непосредственно подсоединённые сети отображается на интерфейс маршрутизатора, к которому они присоединены. Таблица маршрутов отображает сетевые префиксы (адреса сетей) на выходные интерфейсы.

Для направления пакетов к другим адресатам необходимо в таблицу маршрутизации включить дополнительные маршруты (статически или динамически). Статические маршруты не меняются самим маршрутизатором. Динамические маршруты изменяются самим маршрутизатором автоматически при получении информации о смене маршрутов от соседних маршрутизаторов.

Для конфигурации статической маршрутизации в маршрутизаторах Cisco используют две версии команды **ip route**:

1. **ip route АдресСетиНазначения МаскаСетиНазначения Интерфейс**
Команда указывает маршрутизатору, что все пакеты, предназначенные для *АдресСетиНазначения-МаскаСетиНазначения*, следует направлять на свой интерфейс *Интерфейс*. Если интерфейс *Интерфейс* имеет тип Ethernet, то MAC-адреса исходящих пакетов будут широковещательными.
2. **ip route АдресСетиНазначения МаскаСетиНазначения Адрес**
Команда указывает маршрутизатору, что все пакеты, предназначенные для *АдресСетиНазначения-МаскаСетиНазначения*, следует

направлять на тот свой интерфейс, из которого достигим IP-адрес *Адрес*. Как правило, *Адрес* — это адрес следующего хопа по пути к *АдресСетиНазначения*. Выходной интерфейс и физические адреса исходящих пакетов определяются маршрутизатором по своим ARP-таблицам на основании IP-адреса *Адрес*.

Для сетей типа Ethernet рекомендуется всегда использовать форму (2) команды `ip route`. Ethernet-интерфейс на маршрутизаторе, как правило, соединён с несколькими Ethernet-интерфейсами других устройств в сети. Указание в команде `ip route` IP-адреса позволит маршрутизатору правильно сформировать физический адрес выходного пакета по своим ARP-таблицам.

Подсоединитесь к маршрутизатору Router2 и посмотрите таблицу маршрутов. Добавьте маршрут к сети 172.16.10.0/24 через адрес 10.1.1.1 ближайшего хопа на пути к этой сети:

```
Router2#show ip route
Router2#configure terminal
Router2(config)#ip route 172.16.10.0 255.255.255.0 10.1.1.1
Router2(config)#exit
Router2#write memory
Router2#show ip route
Router2#ping 172.16.10.1
```

Подсоединитесь к маршрутизатору Router3 и посмотрите таблицу маршрутов. Добавьте маршрут к сети 10.1.1.0/24 через адрес 172.16.10.1 ближайшего хопа на пути к этой сети:

```
Router3#show ip route
Router3(config)#ip route 10.1.1.0 255.255.255.0 172.16.10.1
Router3(config)#exit
Router3#write memory
Router3#show ip route
Router3#ping 10.1.1.2
```

Сетевые устройства Router2 и Router3 имеют только по одному выходу во внешний мир: через интерфейсы с адресами 10.1.1.1 и 172.16.10.1 соответственно. Поэтому можно не определять? на какие подсети маршрутизируются пакеты, и использовать так называемую *маршрутизацию по умолчанию*.

Маршруты по умолчанию используются, когда маршрутизатор не может поставить в соответствие сети назначения строку в таблице маршрутов. В этом случае маршрутизатор должен использовать маршрут по умолчанию для отсылки пакетов другому маршрутизатору. Следующий маршрутизатор будет иметь маршрут к этой сети назначения или иметь свой маршрут по умолчанию к третьему маршрутизатору и т.д. В конечном счёте пакет будет маршрутизирован на маршрутизатор, имеющий маршрут к сети назначения.

Маршрут по умолчанию может быть статически введен администратором или динамически получен из протокола маршрутизации. Так как все IP-адреса принадлежат сети 0.0.0.0 с маской 0.0.0.0, то в простейшем случае надо использовать команду

```
ip route 0.0.0.0 0.0.0.0 адрес_следующего_хопа
```

или

```
ip route 0.0.0.0 0.0.0.0 выходной_интерфейс
```

Ручное задание маршрута по умолчанию на каждом маршрутизаторе подходит для простых сетей. В сложных сетях необходимо организовать динамический обмен маршрутами по умолчанию.

Сначала удалите старые маршруты на Router2:

```
Router2#configure terminal
Router2(config)#no ip route 172.16.10.0 255.255.255.0 10.1.1.1
```

и на Router3:

```
Router3#configure terminal
Router3(config)#no ip route 10.1.1.0 255.255.255.0 172.16.10.1
```

Назначьте маршруты по умолчанию:

```
Router2#configure terminal
Router2(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1
Router2(config)#exit
Router2#write memory
Router2#show ip route
```

```
Router3#configure terminal
Router3(config)#ip route 0.0.0.0 0.0.0.0 172.16.10.1
Router3#write memory
Router3#show ip route
Router3#ping 10.1.1.2
```

Проверьте доступность всех узлов сети.

Сохраните свой проект и оформите первую часть отчёта, зафиксировав в нём производимые вами действия.

4.4. Задание для самостоятельной работы

1. Создайте новый проект (например, lab04-02.pkt). Постройте в Packet Tracer топологию, представленную на рис. 4.5.

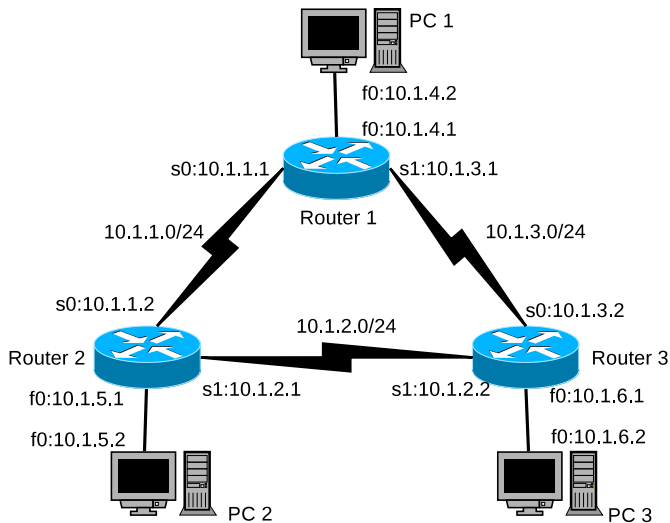


Рис. 4.5. Модель сети для самостоятельной работы

2. На каждом маршрутизаторе настройте используемые интерфейсы и определите соседей.
3. Назначьте интерфейсам сети адреса согласно рис. 4.5.
4. Назначьте шлюзы по умолчанию для компьютеров.
5. Проверьте факт назначения адресов (на каждом маршрутизаторе выполните `show running-config` и `show ip interface brief`). Для компьютеров используйте команду `ipconfig`.
6. Проверьте правильность назначения адресов путём выполнения на каждом маршрутизаторе команды `ping` к непосредственным соседям.
7. Настройте на маршрутизаторах статическую маршрутизацию. На каждом маршрутизаторе посмотрите таблицу маршрутизации. Сохраните конфигурацию.
8. На каждом компьютере выполните команду трассировки `tracert` других компьютеров.
9. Сохраните проект. Подготовьте отчёт, зафиксировав в нём производимые вами действия.

4.5. Моделирование сети с динамической маршрутизацией

4.5.1. Предварительные сведения

Динамическая маршрутизация может быть реализована с помощью одного или нескольких протоколов. Протоколы для работы внутри автономных систем называют *внутренними протоколами шлюзов* (*Interior Gateway Protocols, IGP*), а протоколы для работы между автономными системами называют *внешними протоколами шлюзов* (*Exterior Gateway Protocols, EGP*). К протоколам IGP относятся RIP, RIPv2, IGRP, EIGRP, OSPF и IS-IS. Протоколы EGP3 и BGP4 относятся к EGP.

Маршрутизаторы используют метрики для оценки или измерения маршрутов. Например, протокол RIP использует в качестве метрики количество переходов (хопов), а EIGRP — сложную комбинацию факторов, включающую полосу пропускания канала и его надёжность.

Результаты работы маршрутизирующих протоколов заносятся в таблицу маршрутов, которая постоянно изменяется при смене ситуации в сети. Простые протоколы RIP и IGRP распространяют информацию о таблицах маршрутов через все интерфейсы маршрутизатора в широковещательном режиме без уточнения точного адреса конкретного соседнего маршрутизатора. Соседний маршрутизатор, получая широковещательное сообщение, сравнивает информацию со своей текущей таблицей маршрутов, добавляет в неё маршруты к новым сетям или маршруты к известным сетям с лучшей метрикой, удаляет несуществующие маршруты, добавляет свои собственные значения к метрикам полученных маршрутов. Новая таблица маршрутизации снова распространяется по соседним маршрутизаторам.

Если маршрутизатор работает с протоколом OSPF, то он строит полную базу данных всех состояний связи в своей области. Каждый маршрутизатор затем самостоятельно выполняет *алгоритм поиска кратчайшего пути* (*Shortest Path First, SPF*) на своём собственном отображении сети или базе данных состояний связи для определения лучшего пути, который заносится в таблицу маршрутов. Эти пути к другим сетям формируют дерево с вершиной в виде локального маршрутизатора.

Маршрутизаторы извещают о состоянии своих связей все маршрутизаторы в области. Такое извещение называют *LSA (Link-State Advertisements)*. Имеет место начальный наплыв LSA пакетов для построения базы данных состояний связи. Обновление маршрутов происходит только при изменении состояний связи или в течение определённого интервала времени. Если состояние связи изменилось, то частичное обновление пересылается немедленно. Оно содержит только состояния связей, которые изменились, а не всю таблицу маршрутов.

4.5.2. Порядок выполнения работы

Загрузите проект лабораторной работы раздела 4.3.2 (lab04-01.pkt), пересохраните его под другим именем (например, lab04-03.pkt).

Посмотрите таблицы маршрутизации на Router2 и Router3 и отключите статическую маршрутизацию по умолчанию:

```
Router2#enable
Router2#show ip route
Router2#ping 172.16.10.2
Router2#configure terminal
Router2(config)#no ip route 0.0.0.0 0.0.0.0 10.1.1.1
Router2(config)#exit
Router2#ping 172.16.10.2
Router2#write memory
```

```
Router3#enable
Router3#show ip route
Router3#configure terminal
Router3(config)#no ip route 0.0.0.0 0.0.0.0 172.16.10.1
Router3(config)#exit
Router3#write memory
```

4.5.2.1. Настройка RIP на маршрутизаторах

Включите RIP на всех маршрутизаторах и посмотрите изменение в их конфигурации:

```
Router1#configure terminal
Router1(config)#router rip
Router1(config-router)#network 172.16.10.0
Router1(config-router)#network 10.1.1.0
Router1(config-router)#exit
Router1(config)#exit
Router1#write memory
Router1#show running-config
```

```
Router2#configure terminal
Router2(config)#router rip
Router2(config-router)#network 10.1.1.0
Router2(config-router)#exit
Router2(config)#exit
Router2#write memory
Router2#show running-config
```

```
Router3#configure terminal
Router3(config)#router rip
Router3(config-router)#network 172.16.10.0
Router3(config-router)#exit
Router3(config)#exit
Router3#write memory
Router3#show running-config
```

Сеть 10.1.1.0/24 должна быть воспринята маршрутизаторами как сеть 10.0.0.0/8, а сеть 172.16.10.0/24 — как сеть 172.16.0.0/16. Это связано с классами IP-адресов.

С помощью команды `show ip route` определите параметры, по которым работает протокол RIP. С помощью команды `show ip route` посмотрите таблицы маршрутов. С помощью команд `ping` и `tracert` проверьте доступность других маршрутизаторов.

С помощью команды `debug ip rip` включите трассировку на маршрутизаторе Router1. Посмотрите, как маршрутизаторы обмениваются маршрутной информацией, поясните выводимую информацию. Отключите трассировку с помощью команды `no debug ip rip`. Командой `no router rip` отключите RIP на всех маршрутизаторах, сохраните конфигурацию.

4.5.2.2. Настройка EIGRP на маршрутизаторах

Включите EIGRP на всех маршрутизаторах, образуя автономную систему с номером 100:

```
Router1#configure terminal
Router1(config)#router eigrp 100
Router1(config-router)#network 172.16.10.0
Router1(config-router)#network 10.1.1.0
Router1(config-router)#exit
Router1(config)#exit
Router1#write memory
```

```
Router2#configure terminal
Router2(config)#router eigrp 100
Router2(config-router)#network 10.1.1.0
Router2(config-router)#exit
Router2(config)#exit
Router2#write memory
```

```
Router3#configure terminal
Router3(config)#router eigrp 100
Router3(config-router)#network 172.16.10.0
Router3(config-router)#exit
```

```
Router3(config)#exit
Router3#write memory
```

Посмотрите изменение в конфигурации каждого маршрутизатора. Сеть 10.1.1.0/24 должна быть воспринята как сеть 10.0.0.0/8, а сеть 172.16.10.0/24 — как сеть 172.16.0.0/16. Это связано с классами IP адресов.

С помощью команды `show ip route` определите параметры, по которым работает протокол EIGRP. С помощью команды `show ip route` посмотрите таблицы маршрутов. С помощью команд `ping` и `traceroute` проверьте доступность других маршрутизаторов.

С помощью команды `debug eigrp packets` посмотрите, какими пакетами обмениваются маршрутизаторы. Остановите вывод этой информации командой `no debug eigrp packets`.

Командой `no router eigrp 100` отключите EIGRP на всех маршрутизаторах, сохраните конфигурацию.

4.5.2.3. Настройка OSPF на маршрутизаторах

Включите OSPF на всех маршрутизаторах, присвоив процессу номер 100 с областью действия номер 0:

```
Router1#configure terminal
Router1(config)#router ospf 100
Router1(config-router)#network 172.16.10.0 0.0.0.255 area 0
Router1(config-router)#network 10.1.1.0 0.0.0.255 area 0
Router1(config-router)#exit
Router1(config)#exit
Router1#show running-config
Router1#write memory
```

```
Router2#configure terminal
Router2(config)#router ospf 100
Router2(config-router)#network 10.1.1.0 0.0.0.255 area 0
Router2(config-router)#exit
Router2(config)#exit
Router2#show running-config
Router2#write memory
```

```
Router3#configure terminal
Router3(config)#router ospf 100
Router3(config-router)#network 172.16.10.0 0.0.0.255 area 0
Router3(config-router)#exit
Router3(config)#exit
Router3#show running-config
Router3#write memory
```

Посмотрите изменение в конфигурации каждого маршрутизатора.

С помощью команды `show ip route` определите параметры, по которым работает протокол OSPF. С помощью команды `show ip route` посмотрите таблицы маршрутов. С помощью команд `ping` и `tracert` проверьте доступность других маршрутизаторов.

С помощью команд `ip ospf interface`, `show ip ospf database` и `debug ip ospf events` посмотрите все параметры протокола OSPF.

Сохраните конфигурацию.

4.6. Задание для самостоятельной работы

1. Загрузите проект из самостоятельной работы раздела 4.4 и пересохраните его (например, `lab04-04.pkt`).
2. Отключите на всех маршрутизаторах статическую маршрутизацию. Проверьте с помощью команды просмотра конфигурации маршрутизатора.
3. Настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу RIP. На каждом маршрутизаторе посмотрите таблицу маршрутизации. На каждом компьютере выполните команды трассировки `tracert` других компьютеров.
4. Отключите на маршрутизаторе Router1 последовательный интерфейс Serial 0. Через пару минут, когда в сети пройдут обновления маршрутной информации, на каждом маршрутизаторе посмотрите таблицу маршрутизации. Определите, через какую сеть будут маршрутизироваться пакеты? На каждом компьютере выполните команды трассировки `tracert` других компьютеров. Сохраните файлы конфигурации маршрутизаторов.
5. Отключите RIP и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу IGRP. На каждом маршрутизаторе посмотрите таблицу маршрутизации. На каждом компьютере выполните команды трассировки `tracert` других компьютеров.
6. Отключите на маршрутизаторе Router1 последовательный интерфейс Serial 0. Через пару минут, когда в сети пройдут обновления маршрутной информации, на каждом маршрутизаторе посмотрите таблицу маршрутизации. Определите, через какую сеть будут маршрутизироваться пакеты? На каждом компьютере выполните команды трассировки `tracert` других компьютеров. Сохраните файлы конфигурации маршрутизаторов.
7. Отключите IGRP и настройте на каждом маршрутизаторе динамическую маршрутизацию по протоколу OSPF. На каждом маршрутизаторе посмотрите таблицу маршрутизации. На каждом компьютере выполните команды трассировки `tracert` других компьютеров.
8. Отключите на маршрутизаторе Router1 последовательный интерфейс Serial 0. Через пару минут, когда в сети пройдут обновления

маршрутной информации, на каждом маршрутизаторе посмотрите таблицу маршрутизации. Определите, через какую сеть будут маршрутизироваться пакеты? На каждом компьютере выполните команды трассировки **tracert** других компьютеров. Сохраните файлы конфигурации маршрутизаторов.

Литература

1. Королькова А. В., Кулябов Д. С. Архитектура и принципы построения современных сетей и систем телекоммуникаций. — Москва : Издательство РУДН, 2008. — С. 463.
2. Самуйлов К. Е., Абаев П. О., Гайдамака Ю. В. и др. Мультисервисные сети связи. — Москва : РУДН, 2013. — С. 363. — ISBN: 978-5-209-05014-8.
3. A J. Packet Tracer Network Simulator. — Packt Publishing, 2014. — ISBN: 9781782170433.

Учебно-методический комплекс дисциплины

«Сетевые технологии»

Рекомендуется для направления подготовки
010300.62 «Фундаментальная информатика
и информационные технологии»

Квалификация (степень) выпускника: бакалавр

Программа дисциплины

Цели и задачи дисциплины

Целью курса является введение учащихся в предметную область современных систем и сетей телекоммуникаций.

В процессе преподавания курса решаются следующие задачи:

- изучение и анализ принципов построения и архитектур сетей, функционирующих в режимах коммутации пакетов;
- изучение эталонной модели взаимодействия открытых систем;
- исследование принципов построения и архитектур основных типов современных компьютерных систем.

Место дисциплины в структуре основной образовательной программы

Цикл, к которому относится дисциплина: вариативная часть профессионального цикла Б.3., дисциплина по выбору

Требования к входным знаниям, умениям и компетенциям студента: требуется пройти обучение по дисциплинам: «Архитектура вычислительных систем», «Операционные системы», «Компьютерные сети».

Студент должен:

знать:

- концепции, базовые алгоритмы, принципы разработки и функционирования современных операционных систем (ПК-20);
- теоретические и методические основы, понимание функциональных возможностей конфигурирования и использования операционных систем (ПК-25).

владеть:

- основными методами, способами и средствами получения, хранения, переработки информации, иметь навыки работы с компьютером как средством управления информацией (ОК-12);
- методами и навыками использования и конфигурирования операционных систем и платформенных окружений (ПК-24).

Дисциплины, для которых данная дисциплина является предшествующей: «Прикладные протоколы Интернет и www», «Администрирование локальных сетей», «Информационная безопасность», курсовая работа, выпускная квалификационная работа.

Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций: ОК: 12; ПК: 1, 4, 6–8, 20, 23–29

В результате изучения дисциплины студент должен:

знать:

- концепции, базовые алгоритмы, принципы разработки и функционирования современных операционных систем (ПК-20);
- теоретические и методические основы, функциональные возможности следующих предметных областей: архитектура и организация компьютеров, конфигурирование и использование операционных систем, Разработка и принципы сетевых технологий (ПК-25);
- теоретические основы и общие принципы использования следующих профессиональных областей: системное администрирование, управление информационными коммуникациями (ПК-26);

уметь:

- применять в профессиональной деятельности современные электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты сетевых технологий (ПК-1);
- применять в исследовательской и прикладной деятельности фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области сетевых технологий (ПК-4);
- уметь осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в сети Интернет, способность взаимодействовать и сотрудничать с профессиональными сетевыми сообществами и международными консорциумами, отслеживать динамику развития направлений области сетевых технологий (ПК-6);
- уметь применять на практике международные и профессиональные стандарты информационных технологий, современные парадигмы и методологии, инструментальные средства, относящиеся к сетевым технологиям (ПК-7);
- уметь квалифицированно применять в профессиональной деятельности современные электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий (П-27);
- уметь решать задачи производственной и технологической деятельности на высоком профессиональном уровне, включая: создание информационных ресурсов глобальных сетей (П-28);
- уметь оценивать процессы жизненного цикла информационных систем, сервисов систем информационных технологий (П-29);

владеть:

- владеть основными методами, способами и средствами получения, хранения, переработки информации, иметь навыки работы с компьютером как средством управления информацией (ОК-12);
- базовыми сетевыми технологиями, эффективно применять их для решения научно-технических задач и прикладных задач, связанных с развитием и использованием информационных и сетевых технологий (ПК-8);
- методами и навыками использования и конфигурирования сетевых технологий (ПК-23);
- методами и навыками использования и конфигурирования операционных систем и платформенных окружений (ПК-24).

Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 3 зачетные единицы.

№	Вид учебной работы	Всего часов	Се- мест- ры
			5
1.	Аудиторные занятия (всего)	36	36
	<i>В том числе:</i>		
1.1	Лекции	18	18
1.2	Прочие занятия	18	18
	<i>В том числе:</i>		
1.2.1	<i>Практические занятия (ПЗ)</i>	-	-
1.2.2	<i>Семинары (С)</i>	-	-
1.2.3	<i>Лабораторные работы (ЛР)</i>	18	18
1.2.4	<i>Из них в интерактивной форме (ИФ)</i>	18	18
2	Самостоятельная работа студентов	72	72
	<i>В том числе:</i>		
2.1	Курсовой проект (работа)	-	-
2.2	Расчетно-графические работы	-	-
2.3	Реферат	-	-
2.4	Подготовка и прохождение промежуточной аттестации	18	18

2.5	<i>Другие виды самостоятельной работы:</i>		
2.5.1	Самостоятельная проработка дополнительных материалов по дисциплине	36	36
2.5.2	Выполнение домашних заданий	18	18
3.	Общая трудоёмкость (ак. часов)	108	108
4.	Общая трудоёмкость (зач. ед.)	3	3

Содержание дисциплины

Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Общие принципы построения компьютерных сетей	1.1 Понятие протокола. Иерархия протоколов. Интерфейсы и службы. Обмен данными. 2. Общие принципы построения модели взаимодействия открытых систем.
2.	Физический и канальный уровни модели ISO/OSI	2.1. Кодирование сигнала, среда передачи, кабельная система, стандарты кабельной системы. 2.2. Протоколы доступа к среде (протокол CSMA, полнодуплексный доступ, маркерное кольцо). 2.3. Стандарты серии IEEE 802. Подуровни LLC и MAC. Стандарт IEEE 802.2. Форматы кадров. 2.4. Метод доступа CSMA/CD, спецификация физической среды. Развитие технологий Ethernet, Fast Ethernet, Gigabit Ethernet. Полнодуплексная передача. 2.5. Другие технологии локальных сетей. Технология 100VG-AnyLAN. Token Ring.

3.	Сетевой и транспортный уровни модели ISO/OSI	<p>3.1. Стек протоколов TCP/IP. Соответствие эталонной модели OSI.</p> <p>3.2. Сетевой уровень. Протоколы сетевого уровня. Межсетевой уровень стека протоколов TCP/IP. IP, ICMP, ARP.</p> <p>3.3. Формат кадра IP. IP-адресация. Взаимодействие межсетевого уровня с физическим. Фрагментация IP.</p> <p>3.4. Транспортный уровень. Протоколы TCP, UDP; концепция портов, сессии TCP. Передача пакетов TCP, параметры передачи, MTU, окно. Надёжная доставка.</p>
----	--	---

Разделы дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами

№ п/п	Наименование обеспечиваемых (последующих) дисциплин	№ № разделов данной дисциплины, необходимых для изучения обеспечиваемых (последующих) дисциплин		
		1	2	3
1.	Прикладные протоколы Интернет и www	+		+
2.	Администрирование локальных сетей	+	+	+
3.	Информационная безопасность	+	+	+
4.	Курсовая работа	+	+	+
5.	Выпускная квалификационная работа	+	+	+

Разделы дисциплин и виды занятий

№ п/п	Наименование раздела дисциплины	Лекц.	Практические занятия и лабораторные работы			СРС	Всего час.
			ПЗ/С	ЛР	Из них в ИФ		
1.	Общие принципы построения компьютерных сетей	1				3	4
2.	Физический и канальный уровни модели ISO/OSI	5		4	4	39	48
3.	Сетевой и транспортный уровни модели ISO/OSI	12		14	14	30	56
Итого:		18	0	18	18	72	108

Описание интерактивных занятий

№ п/п	№ р/д	Тема интерактивного занятия	Вид занятия	Труд. (час.)
1	2	Кодирование и модуляция сигнала	Лаб. раб., выполняемая малой группой (2–3 чел.)	2
2	2, 3	Проектирование сетей Ethernet и Fast Ethernet	Лаб. раб., выполняемая малой группой (2–3 чел.)	2
3	3	Знакомство с Packet Tracer. Моделирование простой сети	Лаб. раб., выполняемая малой группой (2–3 чел.)	6
4	3	Packet Tracer. Соединение двух сетей. Маршрутизация	Лаб. раб., выполняемая малой группой (2–3 чел.)	7

Лабораторный практикум

№ п/п	№ р/д	Наименование лабораторных работ	Труд. (час.)
1.	2	Кодирование и модуляция сигнала	2
2.	2, 3	Проектирование сетей Ethernet и Fast Ethernet	2
3.	3	Знакомство с Packet Tracer. Моделирование простой сети	6
4.	3	Packet Tracer. Соединение двух сетей. Маршрутизация	7
Итого:			17

Практические занятия (семинары)

Практические занятия (семинары) не предусмотрены.

Примерная тематика курсовых проектов (работ)

Курсовые работы не предусмотрены.

Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература

1. Кулябов Д.С., Королькова А.В. Архитектура и принципы построения современных сетей и систем телекоммуникаций. — М.: 2008. URL: <http://lib.rudn.ru/polnotekstovye-knigi/61-Kulyabov.pdf>
2. Таненбаум Э. Компьютерные сети, 4 изд. // Спб.: Изд-во «Питер», 2007.

б) Дополнительная литература

1. Семенов Ю. А. Алгоритмы телекоммуникационных сетей. Часть 1. Алгоритмы и протоколы каналов и сетей передачи данных. Интернет-университет информационных технологий — ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007, 640 с. URL: <http://www.intuit.ru/departement/network/algoprotnet/>.

2. Семенов Ю. А. Алгоритмы телекоммуникационных сетей. Часть 2. Протоколы и алгоритмы маршрутизации в INTERNET. Интернет-университет информационных технологий — ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007, 832 с. URL: <http://www.intuit.ru/department/network/pami/>
 3. Семенов Ю. А. Алгоритмы телекоммуникационных сетей. Часть 3. Процедуры, диагностика, безопасность. Интернет-университет информационных технологий — ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007, 512 с. URL: <http://www.intuit.ru/department/network/pdsi/>
 4. Олифер В. Г., Олифер Н. А. Основы компьютерных сетей. — СПб: Питер, 2009.
 5. Парк Д., Маккей С., Райт Э. Передача данных в системах контроля и управления, 2007.
 6. Вишнеvский В., Портной С., Шахнович И. Энциклопедия WiMAX. Путь к 4G, 2009.
 7. Гулевич Д. С. Сети связи следующего поколения, 2009.
 8. Таненбаум Э. Современные операционные системы. — СПб.: Питер. — 2010.
- в) **Программное обеспечение:** ОС Linux, ОС Windows, flash, java, утилиты traceroute, ping, PacketTracer.
- г) **Базы данных, информационно-справочные и поисковые системы:** не требуются.

Материально-техническое обеспечение дисциплины

Москва, ул. Орджоникидзе, д.3, корп. 1, 5, лаборатория «Управление инфокоммуникациями»: ауд. 110: комплект жидко-кристаллический дисплей Sharp PNL702B, Монитор 24" Acer V243HAOBD, системный блок (процессор Intel Core i7-2600 OEM <3.40GHz, 8Mb, 95W, LGA1155(Sandy Bridge)>, 16GB ОП, HDD 2 TB), ноутбук Toshiba Satellite 17/300GB Intel Core2 2.4 GHz (9 шт.); ауд. 116: проектор DMS800 с интерактивной доской Board 1077, HP xw7800, Intel Core2 2.4 GHz (8 шт.); дисплейные классы ДК3, ДК4, ДК5, ДК6, ДК7, Intel Core i3-550 3.2 GHz – 60 шт.

Методические рекомендации по организации изучения дисциплины

Учебным планом на изучение дисциплины отводится один семестр. Промежуточный контроль знаний предусматривает: проведение тестирования в середине семестра, подготовку и сдачу домашнего задания, отчётов по лабораторному практикуму. В качестве итогового контроля знаний предусмотрен зачёт в форме контрольного тестирования.

Дисциплина разбита на 2 модуля:

Первый модуль трудоемкостью в 1 кредит составляют:

- 1) теоретический материал на темы: «Понятие протокола», «Иерархия протоколов», «Интерфейсы и службы», «Обмен данными», «Общие принципы построения модели взаимодействия открытых систем», «Физический и канальный уровни модели ISO/OSI»;
- 2) выполнение домашнего задания по разделу 2;
- 3) выполнение лабораторных работ в интерактивной форме по разделам 2 и 3.

В конце этого модуля проводится промежуточный контроль знаний.

Второй модуль трудоемкостью в 2 кредита составляют:

- 1) теоретический материал на темы: «Сетевой и транспортный уровни модели ISO/OSI», «Стек протоколов TCP/IP»;
- 2) выполнение лабораторных работ в интерактивной форме по разделу 3.

В конце этого модуля проводится итоговый контроль знаний.

Примерный перечень тематики вопросов текущего промежуточного контроля знаний:

1. Модель ISO/OSI. Услуги и функции уровней модели ISO/OSI.
2. Обмен данными.
3. Доступ к среде. Протоколы множественного доступа: протоколы семейства ALOHA, протоколы семейства CSMA.
4. Эталонная модель IEEE 802. Стандарты IEEE 802. Протоколы MAC и LLC.

Примерный перечень тематики вопросов итоговых семестровых испытаний:

1. Сетевой, транспортный и прикладной уровни модели ISO/OSI.
2. Технологии локальных сетей: Ethernet, 100VG–AnyLAN, Token Ring.
3. Стек протоколов TCP/IP.
4. Схема адресации протокола IP. Подсети. Маска подсети.
5. Протоколы транспортного уровня: TCP и UDP.

Фонды оценочных средств

Примерные тестовые задания

- 1.** 10BaseT — это
 - (a) Кабель на основе неэкранированной витой пары
 - (b) Одномодовый оптоволоконный кабель
 - (c) Коаксиальный кабель диаметром 0,5 дюйма
- 2.** ESE — это
 - (a) конец кадра
 - (b) протокол кодирования данных
 - (c) множественные коллизии
- 3.** FSE — это
 - (a) конец кадра
 - (b) ложная несущая
 - (c) сетевой идентификатор
- 4.** IPG — это
 - (a) тип сетевого экрана
 - (b) протокол кодировки данных
 - (c) межпакетная щель
- 5.** IPv4-адрес имеет длину
 - (a) 128 бит
 - (b) 128 байт
 - (c) 32 бита
 - (d) 32 байта
 - (e) 64 бита
- 6.** MAC-адрес Ethernet 00:25:56:ab:f0:39 является
 - (a) индивидуальным
 - (b) групповым
 - (c) локально администрируемым
 - (d) глобально администрируемым

7. PDV — это

- (a) двойное время прохождения сигнала между абонентами сети
- (b) последовательность для усиления коллизии
- (c) усеченная двоичная экспоненциальная отсрочка

8. RIP основан на

- (a) маршрутизации с использованием вектора пути
- (b) маршрутизации с использованием вектора расстояния маршрута
- (c) маршрутизации по состоянию линии
- (d) алгоритме Дейкстры

9. TCP предназначен для соединения:

- (a) процесс - процесс
- (b) хост - хост
- (c) сеть - сеть
- (d) порт - порт

10. UDP принимает входящие данные от

- (a) прикладного уровня
- (b) транспортного уровня
- (c) сетевого уровня

11. Адрес 0:0:0:0:ffff:1.1.1.1 является

- (a) IPv4-совместимым адресом
- (b) IPv4-сопоставленным адресом
- (c) Адресом 6to4

12. Адрес 2002:836B:1::/48 является

- (a) IPv4-совместимым адресом
- (b) IPv4-сопоставленным адресом
- (c) Адресом 6to4

13. В TCP за контроль перегрузки отвечает

- (a) параметр Размер окна (Window)
- (b) параметр Окно перегрузки (Congestion Window, CWnd)
- (c) параметр Порог медленного старта (Slow Start Threshold, SSThresh)

- 14.** В TCP параметр Окно перегрузки (Congestion Window, CWnd)
- (a) отслеживает заполнение входного буфера получателя
 - (b) осуществляет контроль доставки данных
 - (c) понижает интенсивность трафика
 - (d) изменяет размер скользящего окна
- 15.** В какой сети сетевая нагрузка не разделяется: с коммутатором или с концентратором?
- (a) с коммутатором
 - (b) с концентратором
 - (c) разделяется в обоих вариантах
- 16.** В каком типе модуляции для представления данных выполняется смещение несущего сигнала?
- (a) в фазовой модуляции
 - (b) в частотной модуляции
 - (c) в амплитудной модуляции
- 17.** В каком типе модуляции представлению нуля или единицы соответствует наличие или отсутствие соответственно несущей частоты при постоянной амплитуде?
- (a) в амплитудной модуляции
 - (b) в частотной
 - (c) в фазовой
- 18.** Длина сегмента кабеля 100BASE-T ограничена
- (a) 100 метрами
 - (b) 200 метрами
 - (c) 500 метрами
- 19.** Длина сегмента сети 100BASE-FX может достигать
- (a) 400 метров
 - (b) 5 км
 - (c) 10 км
- 20.** Длительность передачи одного бита в CSMA/CD называется
- (a) IT
 - (b) VT
 - (c) IS

21. В чем отличие концентратора класса I от концентратора класса II?

- (a) Он имеет меньшую величину задержки
- (b) Он допускает подключение сегментов только одного типа
- (c) Он преобразует принятые сигналы перед передачей в цифровую форму
- (d) Он позволяет соединять две разные сети
- (e) К нему можно подключать только сегменты на электрическом кабеле

22. В соответствии с основной функцией концентратора — повторением сигнала — его относят к устройствам, работающим на физическом уровне модели OSI. Выберите дополнительные функции концентратора, для выполнения которых концентратору требуется информация протоколов более высоких уровней.

- (a) Поддержка управления по протоколу SNMP, блокировка порта при подключении узла с несанкционированным MAC-адресом, доставка данных в неискаженном виде только узлу назначения
- (b) Поддержка резервных связей
- (c) Подключение к концентратору компьютеров с неизвестными MAC-адресами

23. Для вертикальной подсистемы структурированной кабельной системы подходит:

- (a) Оптоволоконный кабель, толстый коаксиал, широкополосный кабель (используемый в кабельном телевидении)
- (b) Экранированная и неэкранированная витая пара
- (c) Оптоволоконный кабель, тонкий коаксиал, экранированная витая пара

24. Для перехода сети Ethernet на FDDI требуются

- (a) мосты
- (b) коннекторы
- (c) ресиверы

25. Для полного дуплекса длина оптоволоконного кабеля составляет

- (a) до 100—400 м
- (b) до 1—1,5 км
- (c) до 2—10 км

- 26.** Для сети Ethernet 10 Мбит/с PDV не должна превышать
- (a) 512 ВТ
 - (b) 1024 ВТ
 - (c) 2048 ВТ
- 27.** Для сети Fast Ethernet двойная задержка концентратора класса II для T4 составляет:
- (a) 15
 - (b) 28
 - (c) 67
- 28.** Для сети Fast Ethernet двойная задержка концентратора класса II для TX/FX составляет:
- (a) 92
 - (b) 111
 - (c) 128
- 29.** Для сокращения PDV в сети Ethernet следует
- (a) увеличивать число роутеров
 - (b) уменьшать длину кабелей
 - (c) увеличивать нагрузку в сети
- 30.** Для чего минимальный размер кадра в Gigabit Ethernet увеличен до 512 байт?
- (a) для ускорения передачи информации
 - (b) для уменьшения количества коллизий
 - (c) для увеличения допустимой области коллизий
 - (d) для отказа от метода доступа CSMA/CD
 - (e) для увеличения эффективности полнодуплексного режима
- 31.** Для чего служит алгоритм Spanning Tree?
- (a) для удаления замкнутых путей в сети
 - (b) для разбиения сети на широковещательные области
 - (c) для разбиения сети на области коллизии
 - (d) для согласования скоростей передачи отдельных сегментов
 - (e) для предотвращения внутрисегментных передач пакетов

32. Для чего используется jam-последовательность?

- (a) для усиления коллизии
- (b) для определения интервала между пакетами
- (c) для определения IP-адреса

33. Домен коллизий — это

- (a) Часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части этой сети коллизия возникла
- (b) Часть сети Ethernet, все узлы которой распознают коллизию в зависимости от того, в какой части этой сети коллизия возникла
- (c) Часть сети Ethernet, все узлы которой образуют домен

34. Если имеется IP-адрес 172.16.10.5/25, то какой широковещательный адрес должен использовать этот хост?

- (a) 255.255.255.255
- (b) 172.16.10.127
- (c) 172.16.10.255
- (d) 172.16.10.128

35. Имеется маска 255.255.255.252. Какое значение имеет CIDR?

- (a) /16
- (b) /24
- (c) /30
- (d) /32

36. Имеется сеть 192.168.55.0. Требуется разбить ее на несколько подсетей так, чтобы в каждой подсети можно было использовать по 25 хостов при максимально возможном числе таких подсетей. Какую маску подсети следует применить?

- (a) 255.255.255.192
- (b) 255.255.255.224
- (c) 255.255.255.240
- (d) 255.255.255.248

37. Имеются ли отличия в работе сетевых адаптеров, соединяющих компьютер с коммутатором или с мостом, или с концентратором?

- (a) Да, сетевой адаптер, соединенный с коммутатором, может работать в дуплексном режиме, а в остальных случаях — нет
- (b) Нет, сетевой адаптер, соединенный с коммутатором, может работать в любом случае
- (c) Да, сетевой адаптер, соединенный с мостом, может работать в дуплексном режиме, а в остальных случаях — нет

38. Какая ошибка не регистрируется и не исправляется репитерными концентраторами?

- (a) ложная несущая
- (b) множественные коллизии
- (c) затянувшаяся передача
- (d) ошибка в контрольной сумме пакета
- (e) все перечисленные ошибки исправляются

39. Какие из следующих адресов находятся в пределах области IP-адресов многоадресной рассылки?

- (a) 242.127.1.1
- (b) 224.0.0.1
- (c) 239.255.255.254
- (d) 225.128.1.1

40. Какие линии связи имеют высокую пропускную способность и помехозащищенность?

- (a) телефонная пара
- (b) коаксиальный кабель
- (c) витая пара
- (d) волоконно-оптические линии связи
- (e) радиоканал
- (f) спутниковый канал

41. Какие отличия от модели ISO/OSI имеет стандарт LAN IEEE 802?

- (a) число сетевых уровней увеличивается до 8
- (b) число сетевых уровней уменьшается до 5
- (c) на физическом уровне применяются только проводные линии связи
- (d) канальный и физический уровни делятся на подуровни, применяются специальные методы кодирования физических сигналов

42. Какие последствия влечет за собой увеличение скорости сети?

- (a) увеличение нагрузки на сеть
- (b) увеличение количества коллизий
- (c) увеличение пропускной способности сети

43. Какова длина jam-последовательности?

- (a) 32 битовых интервала
- (b) 512 битовых интервалов
- (c) 16 битовых интервалов
- (d) 96 битовых интервалов
- (e) 49 битовых интервалов

44. Какой адрес пакета читает маршрутизатор, чтобы определить следующий участок?

- (a) логический
- (b) физический
- (c) протокола ARP

45. Какой адрес требует передача пакета от узла с IP-адресом 198.123.46.20 для всех маршрутизаторов в сети 198.123.46.0?

- (a) индивидуальный
- (b) групповой рассылки
- (c) широковещательной передачи

46. Какой диапазон допустимых адресов хостов для сети 10.1.0.1/16?

- (a) с 10.1.0.1 по 10.1.255.254
- (b) с 10.1.0.1 по 10.1.255.255
- (c) с 10.1.1.1 по 10.1.1.254
- (d) с 10.1.1.1 по 10.1.1.255

47. Какой из кадров использует поле EtherType Protocol в заголовке LLC для определения протокола более высокого уровня?

- (a) 802.2
- (b) SNAP
- (c) Ethernet II
- (d) 802.3

48. Какой из сетевых подуровней стандарта IEEE 802 определяет конфигурацию LAN и метод доступа к среде передачи данных?

- (a) управление логическим каналом LLC
- (b) управление доступом к передающей среде MAC
- (c) передача физических сигналов PS
- (d) интерфейс с устройством доступа AUI
- (e) подключение к физической среде PMA

49. Какой концентратор можно применять для объединения сегментов 100BASE-TX и 100BASE-T4?

- (a) любой тип концентратора
- (b) концентратор не может соединить эти сегменты
- (c) концентратор класса I
- (d) концентратор класса II
- (e) два концентратора класса II

50. Какой метод нельзя применять для преодоления ограничений на размер сети (зоны конфликта) Ethernet?

- (a) выбор марки кабеля с меньшей задержкой сигнала
- (b) использование коммутаторов для разделения сети
- (c) переход на полнодуплексный режим обмена
- (d) переход на локальную сеть FDDI
- (e) применение дополнительных концентраторов

51. Какой таймер необходим для того, чтобы управлять объявлением при нулевом размере окна TCP?

- (a) повторная передача
- (b) настойчивость
- (c) дежурный
- (d) время ожидания

52. Какой таймер предотвращает слишком долгое свободное соединение между двумя TCP?

- (a) повторная передача
- (b) настойчивость
- (c) дежурный
- (d) время ожидания

53. Какой тип передачи данных реализует рация?

- (a) симплексный
- (b) полудуплексный
- (c) дуплексный

54. К методам контроля ошибок передающим абонентом следует отнести?

- (a) побитовую проверку в процессе передачи пакета
- (b) составление контекстных резервных копий пакетов
- (c) сравнение переданного пакета и пакета, возвращенного принимающим абонентом

55. Когда необходимо использовать перекрестный кабель в сети 10BASE-T?

- (a) при подсоединении компьютера к концентратору
- (b) при присоединении концентратора к порту Uplink другого концентратора
- (c) во всех перечисленных случаях
- (d) при непосредственном соединении между собой двух компьютеров
- (e) ни в одном из перечисленных случаев

56. Код MLT-3

- (a) двухуровневый
- (b) трёхуровневый
- (c) самосинхронизирующий
- (d) даёт выигрыш в скорости
- (e) кодирует сигнал
- (f) кодирует данные

57. Код NRZ

- (a) двухуровневый
- (b) трёхуровневый
- (c) самосинхронизирующий
- (d) даёт выигрыш в скорости
- (e) кодирует сигнал
- (f) кодирует данные

58. Концентраторы всегда работают в режиме

- (a) полудуплекса
- (b) дуплекса
- (c) симплекса

59. К промежуточным сетевым устройствам Ethernet следует отнести

- (a) репитер
- (b) концентратор
- (c) валидатор

60. К функциям концентраторов Ethernet следует отнести

- (a) пересылку пакетов из порта в порты
- (b) уменьшение коллизий
- (c) отключение портов в аварийных ситуациях

- 61.** Максимальная задержка повтора передачи в методе CSMA/CD составляет
- (a) 128 ST
 - (b) 256 ST
 - (c) 1024 ST
- 62.** Максимальная скорость передачи данных в сети Fast Ethernet в полудуплексе составляет
- (a) 10 Мбит/с
 - (b) 50 Мбит/с
 - (c) 100 Мбит/с
- 63.** Пакеты какой сети не включают в себя MAC-адреса?
- (a) Ethernet
 - (b) FDDI
 - (c) Token-Ring
 - (d) 100VG-AnyLAN
 - (e) все пакеты включают в себя MAC-адрес
- 64.** Маршрутизатор работает:
- (a) на физическом уровне
 - (b) на канальном уровне
 - (c) на сетевом уровне
 - (d) на транспортном уровне
 - (e) на сеансовом уровне
 - (f) на уровне представления
 - (g) на прикладном уровне
 - (h) на всех вышеперечисленных уровнях
 - (i) ни на одном из вышеперечисленных уровней
- 65.** Почему для соединения концентраторов между собой используются специальные порты?
- (a) Для исключения необходимости использования перекрестных кабелей
 - (b) Для использования перекрёстных кабелей
 - (c) Для изоляции нескольких концентраторов, выполненных в виде отдельного корпуса

66. Сетевой адаптер совместно с драйвером выполняет:

- (a) Передачу и приём кадра
- (b) Передачу кадра
- (c) Приём кадра

67. Укажите коды без синхронизации

- (a) NRZ
- (b) NRZI
- (c) RZ
- (d) Манчестерский код
- (e) MLT-3

68. Что случится, если во время работы моста/коммутатора произойдет реконфигурация сети, например, будут подключены новые компьютеры?

- (a) Мост/коммутатор автоматически учтет их существование при отправке новыми компьютерами первого кадра в сеть
- (b) Мост/коммутатор не учтёт существования новых

69. Что содержит таблица маршрутизации?

- (a) идентификаторы соседних сетей
- (b) счетчик участков для достижения соседних сетей
- (c) идентификатор следующего участка

70. Что такое коллизия?

- (a) Ситуация, когда две рабочие станции одновременно передают данные в разделяемую передающую среду
- (b) Ситуация, когда две рабочие станции одновременно передают пакет и происходит блокирование
- (c) Ситуация, когда станция, желающая передать пакет, обнаруживает, что в данный момент другая станция уже заняла передающую среду

71. Максимальное количество абонентов в сети Ethernet составляет

- (a) 1024
- (b) 4096
- (c) более 10000

72. Что такое коммутаторы третьего уровня (L3)?

- (a) коммутаторы SAF
- (b) коммутаторы, поддерживающие алгоритм Spanning Tree
- (c) управляемые коммутаторы
- (d) стекируемые коммутаторы
- (e) коммутаторы, выполняющие маршрутизацию

Перечень тем для контроля знаний

1. Модель ISO/OSI. Услуги и функции уровней модели ISO/OSI.
2. Обмен данными.
3. Доступ к среде. Протоколы множественного доступа: протоколы семейства ALOHA, протоколы семейства CSMA.
4. Эталонная модель IEEE 802. Стандарты IEEE 802. Протоколы MAC и LLC.
5. Сетевой, транспортный и прикладной уровни модели ISO/OSI.
6. Технологии локальных сетей: Ethernet, 100VG–AnyLAN, Token Ring.
7. Стек протоколов TCP/IP.
8. Схема адресации протокола IP. Подсети. Маска подсети.
9. Протоколы транспортного уровня: TCP и UDP.

Календарный план

Неделя	Лекции	Число часов	Лабораторные занятия	Число часов
1	Базовые понятия в области систем и сетей телекоммуникаций. Общая характеристика проблемной области. Стандартизирующие организации. Общие принципы построения модели взаимодействия открытых систем (ISO/OSI), иерархия протоколов различных стеков протоколов	1	Кодирование и модуляция сигнала.	2
2	Физический уровень модели ISO/OSI. Обзор возможных сред передачи, СКС, методы кодирования сигнала и сферы их применения	1		
3	Канальный уровень модели ISO/OSI. Протоколы доступа к среде (семейство протоколов ALONA, протокол CSMA, полнодуплексный доступ). Стандарты серии IEEE 802. Подуровни LLC и MAC. Стандарт IEEE 802.2. Форматы кадров	1	Проектирование сетей Ethernet и Fast Ethernet	2

4	Метод доступа CSMA/CD, спецификация физической среды. Развитие технологий Ethernet, Fast Ethernet, Gigabit Ethernet. Полнодуплексная передача	1		
5	Технологии маркерного доступа (TokenBus, TokenRing), FDDI. Технология 100VG-AnyLAN	1	Знакомство с Packet Tracer. Моделирование простой сети	6
6	Промежуточный контроль знаний	1		
7-8	Сетевой уровень модели ISO/OSI. Стек протоколов TCP/IP. Соответствие эталонной модели OSI. Фрагментация IP. Формат кадра IPv4, IPv6	2		
9-10	IPv4-адресация. Планирование сетей IPv4. Разрешение имён на основе DNS	2		
11-12	IPv6-адресация. Планирование сетей IPv6	2	Packet Tracer. Соединение двух сетей. Маршрутизация	7
13-15	Маршрутизация. Статическая, динамическая. Ядерная маршрутизация. Фильтрация пакетов. Протоколы маршрутизации RIP, OSPF, BGP	3		

16, 17	Транспортный уровень. Протоколы TCP, UDP; концепция портов, сессии TCP. Передача пакетов TCP, параметры передачи, MTU, окно. Надёжная доставка	2		
18	Итоговый контроль знаний	1		1

Балльно-рейтинговая система

Рейтинговая система оценки знаний студентов

Раздел	Тема	Формы контроля освоения ООП				Баллы темы	Баллы раздела
		ЛР	ДЗ	тест 1	тест 2		
Общие принципы построения компьютерных сетей	Базовые понятия в области систем и сетей телекоммуникаций. Общая характеристика проблемной области. Стандартизирующие организации. Общие принципы построения модели взаимодействия открытых систем (ISO/OSI), иерархия протоколов различных стеков протоколов			3		3	3
Физический и канальный уровни модели ISO/OSI	Обзор возможных сред передачи, СКС, методы кодирования сигнала и сферы их применения	5		3		8	32
	Протоколы доступа к среде (семейство протоколов ALOHA, протокол CSMA, полнодуплексный доступ). Стандарты серии IEEE 802. Подуровни LLC и MAC. Стандарт IEEE 802.2. Форматы кадров		5	5		10	

	Метод доступа CSMA/CD, спецификация физической среды. Развитие технологий Ethernet, Fast Ethernet, Gigabit Ethernet. Полнодуплексная передача	5		5		10	
	Технологии маркерного доступа (TokenBus, TokenRing), FDDI. Технология 100VG-AnyLAN			4		4	
Сетевой и транспортный уровни модели ISO/OSI	Стек протоколов TCP/IP. Соответствие эталонной модели OSI. Фрагментация IP. Формат кадра IPv4, IPv6				4	4	65
	IPv4-адресация. Планирование сетей IPv4. Разрешение имён на основе DNS	20	5		4	29	
	IPv6-адресация. Планирование сетей IPv6				4	4	
	Маршрутизация. Статическая, динамическая. Ядерная маршрутизация. Фильтрация пакетов. Протоколы маршрутизации RIP, OSPF, BGP	20			4	24	
	Транспортный уровень. Протоколы TCP, UDP; концепция портов, сессии TCP. Передача пакетов TCP, параметры передачи, MTU, окно. Надёжная доставка				4	4	
Итого:		50	10	20	20	100	100

Таблица соответствия баллов и оценок

Баллы БРС	Традиционные оценки РФ	Оценки ECTS
95 - 100	5	A
86 - 94		B
69 - 85	4	C
61 - 68	3	D
51 - 60		E
31 - 50	2	FX
0 - 30		F
	Зачет	Passed

Правила применения БРС

1. Раздел (тема) учебной дисциплины считаются освоенными, если студент набрал более 50 % от возможного числа баллов по этому разделу (теме).
2. Студент не может быть аттестован по дисциплине, если он не освоил все темы и разделы дисциплины, указанные в сводной оценочной таблице дисциплины.
3. По решению преподавателя и с согласия студентов, не освоивших отдельные разделы (темы) изучаемой дисциплины, в течение учебного семестра могут быть повторно проведены мероприятия текущего контроля успеваемости или выданы дополнительные учебные задания по этим темам или разделам. При этом студентам за данную работу засчитывается минимально возможный положительный балл (51 % от максимального балла).
4. При выполнении студентом дополнительных учебных заданий или повторного прохождения мероприятий текущего контроля полученные им баллы засчитываются за конкретные темы. Итоговая сумма баллов не может превышать максимального количества баллов, установленного по данным темам (в соответствии с приказом Ректора № 564 от 20.06.2013). По решению преподавателя предыдущие баллы, полученные студентом по учебным заданиям, могут быть аннулированы.
5. График проведения мероприятий текущего контроля успеваемости формируется в соответствии с календарным планом курса. Студенты обязаны сдавать все задания в сроки, установленные преподавателем.

6. Время, которое отводится студенту на выполнение мероприятий текущего контроля успеваемости, устанавливается преподавателем. По завершении отведённого времени студент должен сдать работу преподавателю, вне зависимости от того, завершена она или нет.
7. Использование источников (в том числе конспектов лекций и лабораторных работ) во время выполнения контрольных мероприятий возможно только с разрешения преподавателя.
8. Отсрочка в прохождении мероприятий текущего контроля успеваемости считается уважительной только в случае болезни студента, что подтверждается наличием у него медицинской справки, заверенной круглой печатью в поликлинике № 25, предоставляемой преподавателю не позднее двух недель после выздоровления. В этом случае выполнение контрольных мероприятий осуществляется после выздоровления студента в срок, назначенный преподавателем. В противном случае отсутствие студента на контрольном мероприятии признается неуважительным.
9. Студент допускается к итоговому контролю знаний с любым количеством баллов, набранных в семестре, но при условии, что у студента имеется теоретическая возможность получить за весь курс не менее 31 балла.
10. Итоговая контроль знаний оценивается из 20 баллов независимо от числа баллов за семестр.
11. Если в итоге за семестр студент получил менее 31 балла, то ему выставляется оценка F и студент должен повторить эту дисциплину в установленном порядке. Если же в итоге студент получил 31–50 баллов, т. е. FX, то студенту разрешается добор необходимого (до 51) количества баллов путём повторного одноразового выполнения предусмотренных контрольных мероприятий, при этом по усмотрению преподавателя аннулируются соответствующие предыдущие результаты. Ликвидация задолженностей проводится в период с 07.02 по 28.02 (с 07.09 по 28.09) по согласованию с деканатом.

Сведения об авторах

Королькова Анна Владиславовна — кандидат физико-математических наук, доцент-исследователь кафедры прикладной информатики и теории вероятностей РУДН.

Кулябов Дмитрий Сергеевич — кандидат физико-математических наук, доцент, доцент-исследователь кафедры прикладной информатики и теории вероятностей РУДН.

Учебное издание

**Анна Владиславовна Королькова
Дмитрий Сергеевич Кулябов**

Сетевые технологии

Лабораторные работы

Тематический план изданий учебной и научной литературы
2014 г., №17

Редактор *И.Л. Панкратова*
Технический редактор *Н. А. Ясько*
Компьютерная вёрстка *А. В. Королькова, Д. С. Кулябов*
Дизайн обложки *Н. А. Ясько*

Подписано в печать 20.03.2014 г. Формат 60×84/16. Печать офсетная.
Усл. печ. л. 6,75. Тираж 100 экз. Заказ № 250.

Российский университет дружбы народов
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3

Типография РУДН
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3, тел. 952-04-41

