

Министерство образования и науки Российской Федерации  
Южно-Российский государственный политехнический университет  
(НПИ) имени М.И. Платова

**С.П. Воробьёв**

# **КОМПЬЮТЕРНЫЕ СЕТИ**

*Учебно-методическое пособие  
для практических занятий*

Новочеркасск  
ЮРГПУ(НПИ)  
2017

УДК 004.023 (О76.5)

Рецензент – **Д.В. Гринченков**, канд. техн. наук, декан факультета информационных технологий и управления, заведующий кафедрой «Программное обеспечение вычислительной техники» ЮРГПУ(НПИ)

### **Воробьев С.П**

Компьютерные сети: учебно-методическое пособие для практических занятий / Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова.– Новочеркасск: ЮРГПУ(НПИ), 2017.– 84 с.

Изложены особенности построения современной архитектуры компьютерных сетей в рамках распределенных корпоративных систем. Представлены тематика, содержание практических занятий по разработке и исследованию архитектуры сетей.

Предназначено для студентов вузов, обучающихся по направлениям подготовки «Прикладная информатика» (бакалавриат), «Информационные системы и технологии» (бакалавриат).

УДК 004.023 (О76.5)

© Южно-Российский государственный  
политехнический университет (НПИ)  
имени М.И. Платова, 2017

## **ВВЕДЕНИЕ**

Современные компьютерные сети — это уникальный комплекс решений для бизнеса, предлагающий широкую функциональность, полную интеграцию, неограниченную масштабируемость и простое взаимодействие в рамках корпоративных инфраструктур ведения бизнеса.

Пособие включает девять тем для проведения практических занятий в течение трех семестров.

Материал пособия позволяет оценивать перспективность новых сетевых технологий и тенденции их развития с точки зрения твердого теоретического фундамента и полученных практических навыков. Пособие может быть использовано также и для самостоятельного изучения базовых принципов и способов реализации компьютерных сетей при повышении квалификации или профессиональной переподготовке.

# 1. ЭЛЕМЕНТЫ ТЕОРИИ ИНФОРМАЦИИ И СИГНАЛОВ

## 1.1. Информационные параметры источника сообщений

Эффективное кодирование информации (или кодирование источника сообщений) связано с преобразованием выходной информации дискретного источника в последовательность букв заданного кодового алфавита. Естественно, что правила кодирования следует выбирать таким образом, чтобы с высокой вероятностью последовательность на выходе источника могла быть восстановлена по закодированной последовательности, а также, чтобы число букв кода, требуемых на одну букву источника, было по возможности меньшим. В теории информации показывается, что минимальное число двоичных букв кода на одну букву источника, требуемых для представления выхода источника, задается энтропией источника. Энтропия  $H(A)$  источника сообщений  $A$  определяется как математическое ожидание количества информации:

$$H(A) = M \left\{ \log \frac{1}{P(a)} \right\},$$

где  $P(a)$  – вероятность того, что источник передает сообщение  $a$  из ансамбля  $A$ . Здесь математическое ожидание обозначает усреднение по всему ансамблю сообщений.

Чем больше энтропия источника, тем больше степень неожиданности передаваемых им сообщений в среднем, т.е. тем более неопределенным является ожидаемое сообщение. Поэтому энтропию часто называют мерой неопределенности сообщения. Можно характеризовать энтропию так же, как меру разнообразия выдаваемых источником сообщений. Если ансамбль содержит  $K$  различных сообщений, то

$$H(A) \leq \log K,$$

причем равенство имеет место только тогда, когда все сообщения передаются равновероятно и независимо. Число  $K$  называется объемом алфавита источника.

Для двоичного источника, когда  $K=2$ , энтропия максимальна при  $P(a_1)=P(a_2)=0,5$  и равна  $\log 2=1$  бит. Энтропия источника зависимых сообщений всегда меньше энтропии источника независимых сообщений при том же объеме алфавита и тех же

безусловных вероятностях сообщений. Для источника с объемом алфавита  $K=32$ , когда буквы выбираются равновероятно и независимо друг от друга, энтропия источника  $H(A)=\log K=5$  бит.

Величина

$$\varepsilon = \frac{H_{\max} - H(A)}{H(A)} = \frac{\log K - H(A)}{\log K}$$

называется избыточностью источника. Она показывает, какая доля максимально возможной при этом алфавите энтропии не используется источником.

В теории информации утверждается, что, передавая двоичные символы со скоростью  $V_k$  симв/с, можно закодировать сообщения так, чтобы передавать их со скоростью

$$V_c = V_k / H(A) - \varepsilon \text{ (сообщений в секунду)}, \quad (1.1)$$

где  $\varepsilon$  - сколь угодно малая величина;

$V_k = 1/T_k$  - число кодовых символов;

$V_c = 1/T_c$  - число передаваемых символов в секунду;

$T_k$  - длительность кодового символа;

$T_c$  - длительность элементарного сообщения.

Если источник передает сообщения независимо и равновероятно, то  $H(A)=\log K$ , и если к тому же  $K=2^n$ , то  $H(A)=n$ . С другой стороны, используя для передачи каждого сигнала последовательность из  $n$  двоичных символов, получим  $2^n = K$  различных последовательностей. То есть каждому сигналу можно сопоставить одну из кодовых последовательностей, так что (1.1) выполняется и при  $\varepsilon=0$ . Аналогичным образом можно закодировать сообщения любого источника с объемом алфавита  $K$ , затрачивая  $V_0 = \log K$  двоичных символов на элементарное сообщение.

Если же сообщения передаются не равновероятно и (или) не независимо, то  $H(A) < \log K$  и сформулированная теорема утверждает, что возможно более экономное кодирование с затратой  $\nu = H(A)$  символов на сообщение. Относительная экономия символов при этом равна  $(\nu_0 - \nu) / \nu_0 \approx \varepsilon$ , т.е. избыточность определяет достижимую степень «сжатия сообщения».

Для источника с равновероятными и независимыми элементарными сообщениями с  $K=32=2^5$  (русские буквы)  $H(A)=5$

бит. Каждую букву можно закодировать последовательностью из пяти двоичных символов, поскольку существует 32 такие последовательности. Подобным образом, т.е. равномерным кодом, можно закодировать и буквы в связном русском тексте. Но можно и уменьшить число символов на букву. Как отмечалось выше, для русского литературного текста  $H(A) \approx 1,5$  бита и, следовательно, возможен способ эффективного кодирования (кодирования со сжатием информации), при котором в среднем на букву русского текста будет затрачено немногим более 1,5 двоичных символа, т.е. на 70 % меньше, чем для равномерного кода.

Во многих практических кодах при кодировании источника короткие кодовые слова приписываются наиболее часто возникающим буквам или сообщениям, а длинные кодовые слова - более редким буквам или сообщениям. Такие коды, у которых различные кодовые слова содержат различное число кодовых символов, называются неравномерными кодами.

Отметим, что применение неравномерного кода позволяет снизить избыточность, вызванную неравной вероятностью букв или сообщений, тогда как увеличение алфавита источника (переход от кодирования отдельных букв к кодированию словаря слов) снижает избыточность, вызванную зависимостью между сообщениями. При этом следует помнить, что зависимость между рядом стоящими буквами в одном слове значительно больше, чем зависимость между соседними словами.

## 1.2. Неравномерные эффективные коды

Пусть источник имеет алфавит  $(a_1, a_2, \dots, a_k)$  с вероятностями  $P(a_1), P(a_2), \dots, P(a_k)$ . Каждая буква источника должна быть представлена кодовым словом, состоящим из последовательности букв, принадлежащих заданному кодовому алфавиту. Обозначим через  $D$  число различных символов в кодовом алфавите ( $D=\{0,1\}$  для двоичного алфавита), а через  $n_k$  - число символов в кодовом слове, соответствующем  $a_k$ . Обозначим величину  $\bar{n}$ , как среднее число кодовых символов на одну букву источника:

$$\bar{n} = \sum_{k=1}^K P(a_k) n_k.$$

Согласно закону больших чисел, если кодируется очень длинная последовательность букв источника с помощью описанной процедуры кодирования, то число кодовых символов на одну букву источника будет с большой вероятностью близко к  $\bar{n}$ .

Для того чтобы изучить вопрос о том, на сколько мало может быть  $\bar{n}$  при однозначном декодировании, рассмотрим ограничения на неравномерный код. Пусть алфавит источника содержит шесть сообщений  $(a_1, a_2, a_3, a_4, a_5, a_6)$ , которые передаются независимо друг от друга с вероятностями  $P(a_1)=0,4$ ,  $P(a_2)=0,3$ ,  $P(a_3)=0,1$ ,  $P(a_4)=0,08$ ,  $P(a_5)=0,07$ ,  $P(a_6)=0,05$ .

Сумма этих вероятностей равна 1. Энтропия этого источника

$$H = \sum_{i=1}^6 P(a_i) \log \frac{1}{P(a_i)} \approx 2,16.$$

Чтобы закодировать эти сообщения равномерным двоичным кодом, требуется на каждое сообщение три символа. В соответствии с теоремой кодирования для источника эти сообщения можно закодировать двоичными символами так, чтобы в среднем на каждое сообщение затрачивалось  $\bar{n} = 2,16 + \varepsilon$  двоичных символов ( $\varepsilon$  – сколь угодно малое положительное число). В табл. 1.1 представлен один из возможных вариантов кодирования K1, где наиболее вероятным сообщениям присвоены наиболее короткие кодовые слова. Таким образом для передачи сообщений  $P(a_1)$ ,  $P(a_2)$ , имеющих суммарную вероятность 0,7, используется один символ, а для передачи остальных четырех сообщений, имеющих суммарную вероятность 0,3, – два символа, так что среднее число символов на сообщение

$$\bar{n} = 0,7 * 1 + 0,3 * 2 = 1,3 \text{ символа.}$$

То есть сообщения закодированы еще более экономично, чем позволяет теорема кодирования. Но при этом не обеспечивается однозначность декодирования (следовательно, выбранный код не пригоден для передачи сообщения). Действительно, если принята последовательность символов  $L = (100110100011110\dots)$ , то ее в соответствии с кодом K1 табл. 1.1 можно декодировать, как  $(a_1, a_1, a_2, a_2, a_1, a_2, a_1, a_1, a_1, a_2)$  или  $(a_3, a_6, a_4, a_3, a_1, a_6, a_2, a_5 \dots)$  и другими различными способами.

Однозначность декодирования при коде K1 можно обеспечить, если после каждого сообщения передавать некоторый символ, разделяющий сообщения. В этом случае уже будет не двоичный код, а троичный. Это используется в коде Морзе, где кроме точки и тире, используется третий символ – "пробел". Очевидно, что введение разделительного символа снижает эффективность кодирования.

Таблица 1.1 - Произвольное кодирование сообщений

Сообщение	a1	a2	a3	a4	a5	a6
Код K1	0	1	00	01	10	11

Однозначность декодирования можно обеспечить, не вводя разделительного символа, если строить код так, чтобы он удовлетворял условию, известному под названием "свойство префикса". Оно заключается в том, что ни одно используемое кодовое слово не должно совпадать с началом ("префиксом") другого кодового слова. Коды, удовлетворяющие этому условию, называют префиксными кодами. Это свойство не выполнено у кода K1, так как, например, слово, соответствующее сообщению  $a_1$ , является началом слова, соответствующего сообщению  $a_3$ , и т.д.

Существует несколько алгоритмов построения префиксных кодов. Среди них коды Шеннона-Фано и Хаффмана ближе всего позволяют приблизиться к границе, определяемой энтропией.

### 1.3. Код Шеннона-Фано

Сообщения алфавита источника выписывают в порядке убывания вероятностей их появления. Далее разделяют их на две части так, чтобы суммарные вероятности сообщений в каждой из этих частей были по возможности почти одинаковыми. Сообщениям первой части приписывается в качестве первого символа 0, а сообщениям второй части – 1 (можно и наоборот). Затем каждая из этих частей (если она содержит более одного сообщения) делится на две по возможности равновероятные части и в качестве второго символа для первой из них берется 0, а для второй – 1. Этот процесс повторяется, пока в каждой из полученных частей не останется по одному сообщению. Для примера, приведенного в табл. 1.1, на первом этапе деления первой части окажется одно сообщение  $a_1$  с вероятностью  $P(a_1)=0,4$ , во второй части – остальные сообщения с суммарной вероятностью



$P_{\Sigma}(a_2-a_6)=0,6$ . Припишем сообщению  $a_1$  символ 0, а остальным сообщениям в качестве первого символа – 1.

На втором этапе разделим сообщения  $(a_2, a_3, a_4, a_5, a_6)$  на две равновероятные части, включив в первую часть сообщения  $a_2$ , а во вторую часть – сообщения  $(a_3, a_4, a_5, a_6)$ . Припишем сообщению  $a_2$  в качестве второго символа 0, а остальным сообщениям – 1 и т.д. В результате приходим к коду  $K_2$ , приведенному в табл. 1.2.

Таблица 1.2 - Кодирование сообщений кодом Шеннона-Фано

Сообщение	Вероятности	Код $K_2$	Степень разбиения
$a_1$	0,4	0	I
$a_2$	0,3	10	II
$a_3$	0,1	1100	III
$a_4$	0,08	1101	IV
$a_5$	0,07	1110	III
$a_6$	0,05	1111	IV

Код по своему построению удовлетворяет свойству префикса. Поэтому вышеприведенная после последовательность двоичных символов “ $L$ ” декодируется однозначно:  $(a_1, a_1, a_4, a_1, a_1, a_1, a_6, a_1)$ . Среднее число символов на одно сообщение с учетом их вероятностей  $\bar{n} = 0,4 \cdot 1 + 0,3 \cdot 2 + 0,3 \cdot 4 = 2,2$ , т.е. незначительно превышает энтропию источника сообщений.

#### 1.4. Средняя длина кодового слова

Процедура Шеннона-Фано не обязательно минимизирует  $\bar{n}$ , так как достижение большого значения средней собственной информации на одной кодовой букве может привести к обедненному выбору для последующих кодовых букв. Если это разбиение может быть вычислено так, что группы будут в точности равновероятны на каждом этапе разбиения, то вероятности букв источника и длины кодовых слов будут связаны равенством

$$P(a_k) = \bar{A}^{-n_k}. \quad (1.2)$$

Ограничения на длины кодовых слов префиксного кода задаются неравенством Крафта и теоремой кодирования для источника.

**Теорема 1.** Неравенство Крафта. Если целые числа  $(n_1, n_2, n_3, \dots, n_K)$  удовлетворяют неравенству

$$\sum_{k=1}^K 2^{-n_k} \leq 1, \quad (1.3)$$

то существует код, обладающий свойством префикса с алфавитом объемом  $D$ , длины кодовых слов в котором равны этим числам. Обратно, длины кодовых слов любого кода, обладающего свойством префикса, удовлетворяет неравенству (1.3). Теорема не утверждает, что любой код с длинами кодовых слов, удовлетворяющими (1.3), является префиксным. Так, например, множество двоичных кодовых слов  $(0; 00; 11)$  удовлетворяет (1.3), но не обладает свойством префикса. Теорема утверждает, что существует некоторый префиксный код с такими длинами, например код  $(0; 10; 11)$ . Не любой однозначно декодирующий код обладает свойством префикса, например, код КЗ (табл. 1.3). В нем каждое кодовое слово является префиксом каждого более длинного кодового слова. Вместе с тем однозначность декодирования является тривиальной, так как символ 0 всегда определяет начало нового кодового слова. Коды, обладающие свойством префикса, отличаются, от других однозначно декодируемых кодов тем, что конец кодового слова всегда может быть опознан, так что декодирование может быть выполнено без задержки наблюдаемой последовательности кодовых слов (код. К4, табл. 1.3). По этой причине префиксные коды иногда называют мгновенными кодами.

Таблица 1.3 - Однозначно декодируемые коды

Сообщение	Вероятности	Код КЗ	Код К4
a1	0.5	0	0
a2	0.25	01	10
a3	0.125	011	110
a4	0.125	0111	111

### 1.5. Коды Хаффмана

Методика Шеннона-Фано не всегда приводит к однозначному построению кода, поскольку при разбиении на части можно сделать больше по вероятности как верхнюю, так и нижнюю части. Кроме того, методика не обеспечивает отыскания оптимального множества кодовых слов для кодирования данного множества сообщений. (Под

оптимальностью подразумевается то, что никакое другое однозначно декодируемое множество кодовых слов не имеет меньшую среднюю длину кодового слова, чем заданное множество.) Предложенная Хаффманом конструктивная методика свободна от отмеченных недостатков. Методика Хаффмана основывается на нижеследующей лемме.

**Лемма.** Для любого заданного источника с  $K \geq 2$  буквами существует оптимальный двоичный код, в котором два наименее вероятных кодовых слова  $X_k$  и  $X_{k-1}$  имеют одну и ту же длину и отличаются лишь последним символом:  $X_k$  оканчивается на 1, а  $X_{k-1}$  на 0 ( $X_1, X_2, \dots, X_k$  – множество двоичных кодовых слов источника ( $a_1, a_2, \dots, a_k$ ) с вероятностями  $P(a_1), P(a_2), \dots, P(a_k)$ , и для простоты обозначений буквы упорядочены так, что  $P(a_1) \geq P(a_2) \geq \dots \geq P(a_k)$ ,  $(n_1, n_2, \dots, n_k)$  – длины кодовых слов).

С помощью этой леммы задача построения оптимального кода сводится к задаче построения  $X_1, X_2, \dots, X_{k-2}$  и отыскания первых  $n_{k-1}$  символов  $X_k$ . Определим редуцированный ансамбль  $A^*$  как ансамбль, состоящий из букв  $a_1, a_2, \dots, a_{k-1}$ , с вероятностями:

$$P_r(a'_k) = \begin{cases} P(a_k) & , k \leq K-2; \\ P(a_{k-1}) + P(a_k) & , k = K-1. \end{cases}$$

Таким образом, любой префиксный код для  $A^*$  можно превратить в соответствующий код для  $A$  добавлением конечного символа 0 к  $X'_{k-1}$  для получения  $X_{k-1}$ , и добавления конечного символа 1 к  $X'_{k-1}$  для получения  $X_k$ . Отсюда следует, что задача отыскания оптимального кода сведена теперь к задаче отыскания оптимального кода для редуцированного ансамбля, имеющего на одно сообщение меньше. Данный ансамбль может иметь свои два наименее вероятные сообщения, сгруппированные вместе, и может быть произведен следующий редуцированный ансамбль. Продолжая, таким образом, можно достичь того, что получится ансамбль, состоящий только из двух сообщений, и тогда оптимальный код получается приписыванием 1 одному сообщению и 0 другому.

Систематическая процедура для выполнения описанных операций может быть представлена следующим образом. Буквы алфавита сообщений выписывают в основной столбец таблицы кодирования в порядке убывания вероятностей. Две последние буквы объединяют в одну вспомогательную букву, которой приписывают

суммарную вероятность. Вероятности букв, не участвовавших в объединении, и полученная суммарная вероятность слова располагаются в порядке убывания вероятностей в дополнительном столбце, а две последние объединяют. Процесс продолжается до тех пор, пока не получим единственную вспомогательную букву с вероятностью, равной единица. Осуществим оптимальное кодирование ансамбля сообщений, приведенного в табл. 1.4, кодом Хаффмана.

Для нахождения кодовой комбинации, соответствующей  $i$ -му знаку, необходимо проследить путь перехода знака по строкам и столбцам таблицы. Это наиболее наглядно осуществимо по кодовому дереву. Из точки, соответствующей вероятности 1, направляется две ветви, причем ветви с большей вероятностью присваиваем символ 1, а с меньшей - 0. Такое последовательное ветвление продолжается до тех пор, пока не дойдем до вероятности каждой буквы.

Таблица 1.4 - Кодирование сообщений кодом Хаффмана

Сообщение	Вероятность	Вспомогательные столбцы						
		1	2	3	4	5	6	7
a1	0.22	0.22	0.22	0.26	0.32	0.42	0.58	1.0
a2	0.20	0.20	0.20	0.22	0.26	0.32	0.42	
a3	0.16	0.16	0.16	0.20	0.22	0.26		
a4	0.16	0.16	0.16	0.16	0.20			
a5	0.10	0.10	0.16	0.16				
a6	0.10	0.10	0.10					
a7	0.04	0.06						
a8	0.02							

Таблица 1.5 - Кодовые комбинации кода Хаффмана

Сообщение	1	2	3	4	5	6	7	8
Код K5	01	0	11	10	00	011	0101	0100

Кодовое дерево для алфавита сообщений табл. 1.4 приведено на рис. 1.1. Двигаясь по кодовому дереву сверху вниз, можно записать для каждого сообщения соответствующие ему кодовые комбинации (табл. 1.5).

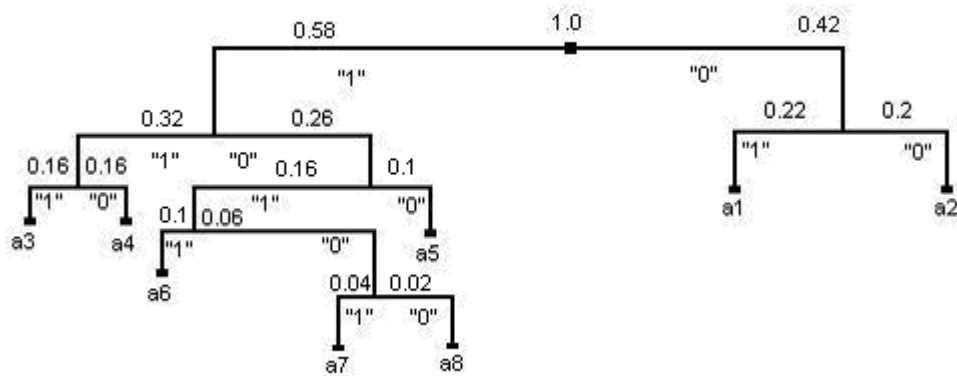


Рис. 1.1. Кодовое дерево кода Хаффмана

### 1.6. Задание практической работы

1. Закодировать сообщения табл. 1.4 кодом Шеннона-Фано при различных разбиениях. Вычислить энтропию сообщений и среднюю длину кодового слова. Сравнить средние длины кодовых слов для различных разбиений.
2. Вычислить среднюю длину кодового слова кодовых комбинаций табл. 1.5.
3. Закодировать сообщения табл. 1.2 кодом Хаффмана. Вычислить среднюю длину кодового слова.
4. Пусть  $A$  – ансамбль равновероятных десятичных цифр от 0 до 9. Энтропия этого ансамбля  $H(A) = \log_{10} 10 \approx 3,32$  бит.
5. Найдите длину равномерного двоичного кода, однозначно кодирующего ансамбль  $A$ .
6. Найдите среднюю длину двоичного кода Хаффмана, кодирующего этот же ансамбль.

### Контрольные вопросы

1. Чему равна энтропия источника дискретных сообщений при равновероятных и независимых сообщениях? Как она изменяется при увеличении объема алфавита?
2. Что называется избыточностью алфавита источника?
3. Объясните сущность побуквенного кодирования и кодирования словаря?
4. За счет чего при эффективном кодировании уменьшается средняя длина кодового слова?

5. До какого предела может быть уменьшена средняя длина кодового слова при эффективном кодировании?
6. В чем преимущество методики Хаффмана по сравнению с методикой Шеннона-Фано?
7. Какому основному условию должны удовлетворять эффективные коды?

## 2. СЕТИ И СЕТЕВЫЕ ТЕХНОЛОГИИ НИЖНИХ УРОВНЕЙ

### Задача № 1

По IP-адресу и маске сети построить:

- а) адрес сети;
- б) широковещательный адрес.

Исходные данные:

- 1) IP-адрес – 198.87.137.221; маска – 255.255.128.0;
- 2) 198.87.137.221/14;
- 3) 88.37.246.135/2;
- 4) 88.37.246.135/9;
- 5) 88.37.246.135/18;
- 6) 88.37.246.135/25.

### Задача № 2

Для заданной пары хостов с известными IP и MAC-адресами построить заголовки IP и Ethernet. Обратить внимание на взаимодействие уровней эталонной модели при инкапсуляции с помощью номера типа сетевого (транспортного) уровня:

- 1) X: IP=194.125.16.38, MAC=08:00:09:a1:cc:b6, TTL=67, Protocol=TCP, Data\_Length=356  
Y: IP=224.88.137.15, MAC=02:60:8c:cd:a8:1b
- 2) X: IP=25.125.225.75, MAC=00:00:0c:aa:bb:cc, TTL=10, Protocol=UDP, Data\_Length=100;  
Y: IP=111.222.133.155, MAC=00:00:0f:1a:2b:3c
- 3) X: IP=94.194.34.244, MAC=00:00:10:a5:c6:b7, Precedence=5, TTL=100, Protocol=TCP,  
Data\_Length=500; Y: IP=33.131.237.215, MAC=00:00:0d:dd:ee:3a

4) X: IP=143.19.74.138, MAC=00:00:c0:88:a6:be, Precedence=3, TTL=200, Protocol=UDP, Data\_Length=300; Y: IP=244.166.242.185, MAC=00:aa:00:da:ba:a7  
 5) X: IP=14.65.243.138, MAC=08:00:20:10:11:c1, TTL=210, Protocol=TCP, Data\_Length=1500; Y: IP=47.87.237.219, MAC=08:00:5a:c6:ae:8b

### Задача № 3

Построить последовательность IP-пакетов при фрагментации дейтаграмм:

- 1) X: IP=237.163.83.179, длина исходного пакета – 600, TTL=220, Protocol=TCP  
 Y: IP=92.157.165.18, MRU=256
- 2) X: IP=25.125.225.75, TTL=10, Protocol=UDP, Data\_Length=1000;  
 Y: IP=111.222.133.155, MRU=400.
- 3) X: IP=94.194.34.244, Precedence=5, TTL=100, Protocol=TCP, Data\_Length=500;  
 Y: IP=33.131.237.215, MRU=300.
- 4) X: IP=143.19.74.138, Precedence=3, TTL=200, Protocol=UDP, Data\_Length=300;  
 Y: IP=244.166.242.185, MRU=100.
- 5) X: IP=14.65.243.138, TTL=210, Protocol=TCP, Data\_Length=1500;  
 Y: IP=47.87.237.219, MRU=500.

### Задача № 4

Для заданной локальной сети построить кадры ARP-запроса и ответа.

- 1) X: IP=237.163.83.179, MAC=08:00:09:a1:cc:b6,  
 Y: IP=92.157.165.18, MAC=02:60:8c:cd:a8:1b
- 2) X: IP=25.125.225.75, MAC=00:00:77:22:aa:33;  
 Y: IP=111.222.133.155, MAC=08:00:10:ab:cd:ef.
- 3) X: IP=94.194.34.244, MAC=08:00:69:cc:aa:2b;  
 Y: IP=33.131.237.215, MAC=00:00:1d:ae:ed:e5.
- 4) X: IP=143.19.74.138, MAC=00:00:6b:9d:a7:c3;  
 Y: IP=244.166.242.185, MAC=08:00:11:ac:77:ed.
- 5) X: IP=14.65.243.138, MAC=00:00:0f:99:aa:bb;  
 Y: IP=47.87.237.219, MAC=08:00:38:a6:c7:e5.

**Задача № 5**

Построить кадр PPP с инкапсуляцией IP-дейтаграммы.

**Задача № 6**

Построить последовательность LCP-пакетов конфигурирования линии:

- параметры соединения согласованы (Configure–Ack);
- параметры соединения частично согласованы (Configure–Nak);
- параметры соединения не согласованы (Configure–Reject).

1) Составьте запрос с параметром соединения Authentication-Protocol (протокол аутентификации) PAP.

2) Составьте запросы согласования параметра соединения Maximum-Receive-Unit (MRU).

3) Составьте запрос с параметром соединения Quality-Protocol.

**Задача № 7**

Построить последовательность LCP-пакетов завершения связи.

**Задача № 8**

Построить последовательность пакетов аутентификации PAP.

1) Составьте запрос Authenticate-Request с параметрами:

имя\_пользователя=car, пароль=bmw, используя протокол аутентификации PAP.

2) Составьте ответ Authenticate-Nak, не принятый параметр пароль\_пользователя.

3) Составьте ответ Authenticate-Reject.

**Задача № 9**

Построить последовательность IPCP-пакетов для получения IP-адреса:

1) Составьте ответ Configure-Nak, в котором передаются опции с кодами 0x03, 0x81, 0x83.

2) Составьте ответ Configure-Reject, в котором не согласованы опции с кодами 0x81, 0x83.



**Задача № 10**

Выполнить анализ заданного дампа последовательности PPP-пакетов.

**Контрольные вопросы**

1. Перечислить основные поля заголовка Ethernet.
2. Перечислить основные поля заголовка IP.
3. Перечислить основные поля ARP запроса/ответа.
4. Каким образом ПО стека протоколов определяет, какой пакет инкапсулирован в Ethernet кадр?
5. Каким образом ПО стека протоколов определяет, какой протокол транспортного уровня следует использовать при интерпретации IP-дейтаграммы?
6. Для чего необходима фрагментация пакетов?
7. Каким образом задаётся последовательность фрагментов?
8. Как определяется неизвестный MAC-адрес с помощью протокола ARP?
9. Перечислить основные поля пакета PPP.
10. Перечислить основные фазы PPP-соединения.
11. Перечислить основные запросы для установления соединения.
12. Перечислить основные опции, определяемые для установления соединения.
13. Какие протоколы аутентификации используются в PPP-соединении?
14. Каким образом происходит присвоение IP адреса устройству?
15. Как осуществляется передача информации (данных) в PPP-соединении?
16. Какова процедура завершения в PPP-соединения?

### 3. ИНТЕГРИРОВАННЫЕ И ДИФФЕРЕНЦИРОВАННЫЕ УСЛУГИ КОМПЬЮТЕРНЫХ СИСТЕМ

#### Управление полосой пропускания

Современные коммутаторы позволяют регулировать интенсивность трафика на своих портах с целью обеспечения функций качества обслуживания.

Для управления полосой пропускания входящего и исходящего трафика на портах *Ethernet* коммутаторы D-Link поддерживают функцию *Bandwidth Control*, которая использует для ограничения скорости механизм *Traffic Policing*. Администратор может вручную устанавливать требуемую скорость соединения на порте в диапазоне от 64 Кбит/с до максимально поддерживаемой скорости интерфейса с шагом 64 Кбит/с.

**Задание:** Настроить ограничение полосы пропускания на коммутаторе D-Link на стенде сети, структура которого показана на рис. 3.1.

#### Оборудование:

DES-3200-28	1 шт.
Рабочая станция	4 шт.
Кабель Ethernet	5 шт.
Консольный кабель	1 шт.

#### Порядок выполнения задания по настройке DES-3200-28

**Примечание.** Значение скорости в командах указывается в килобитах.

*Перед выполнением задания необходимо сбросить настройки коммутатора к заводским настройкам по умолчанию командой*

```
reset config
```

*Настройте полосу пропускания на портах 1-4, равной 5 Мбит/с для входящего и исходящего трафика*

```
config bandwidth_control 1-4 rx_rate 5270 tx_rate 5270
```

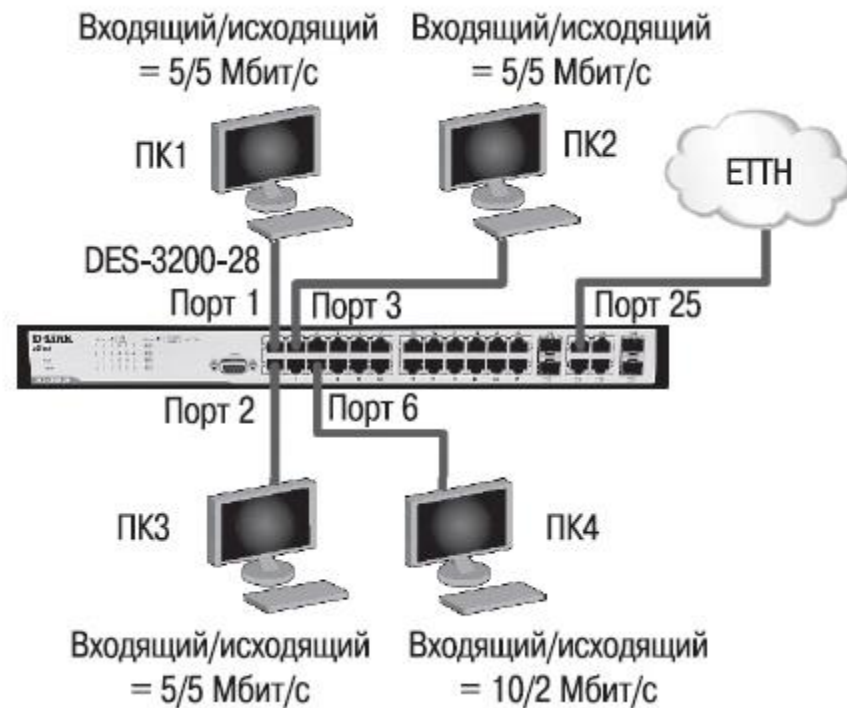


Рис. 3.1. Схема вычислительной сети для изучения управления полосой пропускания

Настройте полосу пропускания на порте 6, равной 10 Мбит/с для входящего и 2 Мбит/с для исходящего трафика

```
config bandwidth_control 6 rx_rate 10240 tx_rate 2048
```

*Проверьте выполненные настройки*

```
show bandwidth_control 1-10
```

### Упражнения

Подключите станции ПК1 и ПК2 к портам 8 и 10 и попробуйте скачать файл размером 50 Мб со станции ПК1 на станцию ПК2 и обратно.

*Запишите время передачи файла (в секундах):*

Подключите станцию ПК1 к порту 1, повторите скачивание.

*Запишите время передачи файла (в секундах):*

Подключите станцию ПК1 к порту 6, повторите скачивание.

- *Запишите время передачи файла (в секундах):*

## Настройка QoS. Приоритизация трафика

Сети с коммутацией пакетов на основе протокола *IP* не обеспечивают гарантированной пропускной способности, поскольку не обеспечивают гарантированной доставки.

Для приложений, где не важен порядок и *интервал* прихода пакетов, время задержек между отдельными пакетами не имеет решающего значения. Для приложений, чувствительных к задержкам, в сети должны быть реализованы *механизмы*, обеспечивающие функции качества обслуживания (*Quality of Service*, QoS).

Функции качества обслуживания в современных сетях заключаются в обеспечении гарантированного и дифференцированного уровня обслуживания сетевого трафика, запрашиваемого теми или иными приложениями на основе различных механизмов распределения ресурсов, ограничения интенсивности трафика, обработки очередей и приоритизации.

Для обеспечения QoS на канальном уровне модели *OSI* коммутаторы поддерживают стандарт *IEEE 802.1p*. Стандарт *IEEE 802.1p* позволяет задать до 8 уровней приоритетов (от 0 до 7, где 7 — наивысший), определяющих способ обработки кадра, используя 3 бита поля приоритета тега *IEEE 802.1Q*.

При выполнении практической работы подразумевается следующий пример: на компьютерах В и Д запущены приложения *VoIP*, и им необходимо обеспечивать высокий приоритет обработки по сравнению с приложениями других станций.

**Задание:** Изучить настройку приоритизации трафика на коммутаторах D-Link на стенде сети, структура которого показана на рис. 3.2.

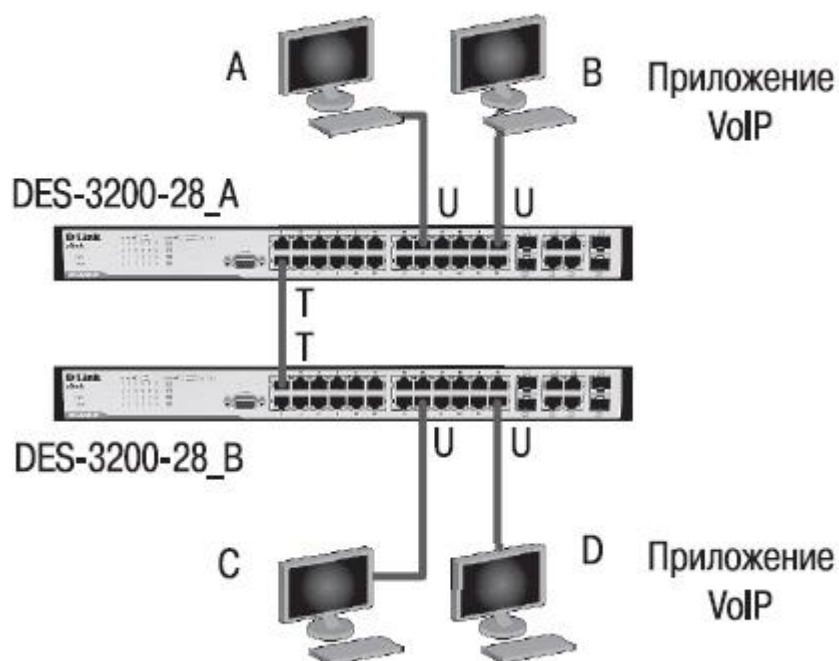
### Оборудование:

DES-3200-28	2 шт.
Рабочая станция	4 шт.
Кабель Ethernet	5 шт.
Консольный кабель	2 шт.

## Порядок выполнения задания

*Перед выполнением задания необходимо сбросить настройки коммутаторов к заводским настройкам по умолчанию командой*

```
reset config
```



**Рис. 3.2. Схема сети для изучения механизмов приоритизации трафика**  
Настройка DES-3200-28\_A

*Переведите порт 1 на коммутаторе в состояние передачи маркированных кадров (для обеспечения возможности передачи информации о приоритете 802.1p)*

```
config vlan default delete 1
config vlan default add tagged 1
```

*Поменяйте приоритет по умолчанию порта 23, к которому подключена станция B*

```
config 802.1p default_priority 23 7
```

**Примечание.** Пользовательский приоритет и метод обработки остаются по умолчанию.

## Настройка DES-3200-28\_B

*Переведите порт 1 на коммутаторе в состояние передачи маркированных кадров (для обеспечения возможности передачи информации о приоритете 802.1p)*

```
config vlan default delete 1  
config vlan default add tagged 1
```

*Поменяйте приоритет по умолчанию порта 23, к которому подключена станция D*

```
config 802.1p default_priority 24 7
```

**Примечание.** Благодаря изменению значения приоритета портов, к которым подключены компьютеры с *VoIP*-приложениями на 7, все кадры, передаваемые ими, получают наивысший приоритет по сравнению с кадрами, поступающими от других компьютеров на остальные порты обоих коммутаторов.

### **Упражнение 1**

*Посмотрите текущие настройки приоритета по умолчанию на всех портах коммутаторов A и B*

```
show 802.1p default_priority
```

*Посмотрите карту привязки пользовательских приоритетов 802.1p к очередям класса обслуживания*

```
show 802.1p user_priority
```

### **Упражнение 2 (дополнительно)**

Изучите механизм обслуживания очередей приоритетов *Weighted Round Robin* (WRR, взвешенный алгоритм кругового обслуживания).

**Примечание.** Для обработки очередей приоритетов могут использоваться различные механизмы обслуживания. В коммутаторах D-Link используются две схемы обслуживания очередей: очереди

приоритетов со строгим режимом (*Strict Priority Queue*) и взвешенный алгоритм кругового обслуживания (*Weighted Round Robin*). В первом случае пакеты, находящиеся в очереди с высшим приоритетом, начинают передаваться первыми. При этом пока очередь с более высоким приоритетом не опустеет, пакеты из очередей с низшим приоритетом передаваться не будут. Вторым алгоритм WRR устраняет это ограничение, а также исключает нехватку полосы пропускания для очередей с низким приоритетом. Этот механизм обеспечивает обработку очередей в соответствии с назначенным им весом и предоставляет полосу пропускания для пакетов из низкоприоритетных очередей.

*Поменяйте механизм обработки очередей на WRR (Strict используется по умолчанию)*

```
config scheduling_mechanism weight_fair
```

*Проверьте механизм обработки очередей*

```
show scheduling_mechanism
```

*Назначьте вес обработки*

```
config scheduling 0 weight 10
config scheduling 1 weight 15
config scheduling 2 weight 25
config scheduling 3 weight 55
```

*Выполните команду просмотра очередности обслуживания*

```
show scheduling
```

## Контрольные вопросы

1. Какой командой можно сбросить настройки коммутатора?
2. Какой командой настраивается полоса пропускания?
3. В чём заключаются функции качества обслуживания?
4. Сколько уровней приоритетов позволяет задать стандарт *IEEE 802.1p*?
5. Какой командой поменять приоритет по умолчанию для порта?

## 4. АРХИТЕКТУРА ETHERNET

Спецификация сети Ethernet была предложена фирмами DEC, Intel и Xerox (DIX) в 1980 году, 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 (T, 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 обнаруживаем 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; при 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.

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); «100BASEFX» и «100BASE-SX» (оптоволокно).

**Первая модель сети Fast Ethernet.** Модель представляет собой по сути набор правил построения сети:

- 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.

Значения для предельно допустимого диаметра домена коллизий в Fast Ethernet при различных комбинациях сегментов приведены в табл. 4.1.

Таблица 4.1 - 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	-	-
Один повторитель класса I	200	272	231	260,8
Один повторитель класса II	200	320	-	308,8
Два повторителя класса II	205	228	-	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. Время двойного оборота рассчитывается для наихудшего (в смысле распространения сигнала) пути между двумя узлами домена коллизий (табл. 4.2.). Расчёт выполняется путём суммирования временных задержек в сегментах, повторителях и терминалах. 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 непредвиденных 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.

Таблица 4.2 - 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

На рис. 4.1 приведён пример одной из предельно допустимых конфигураций сети Fast Ethernet.

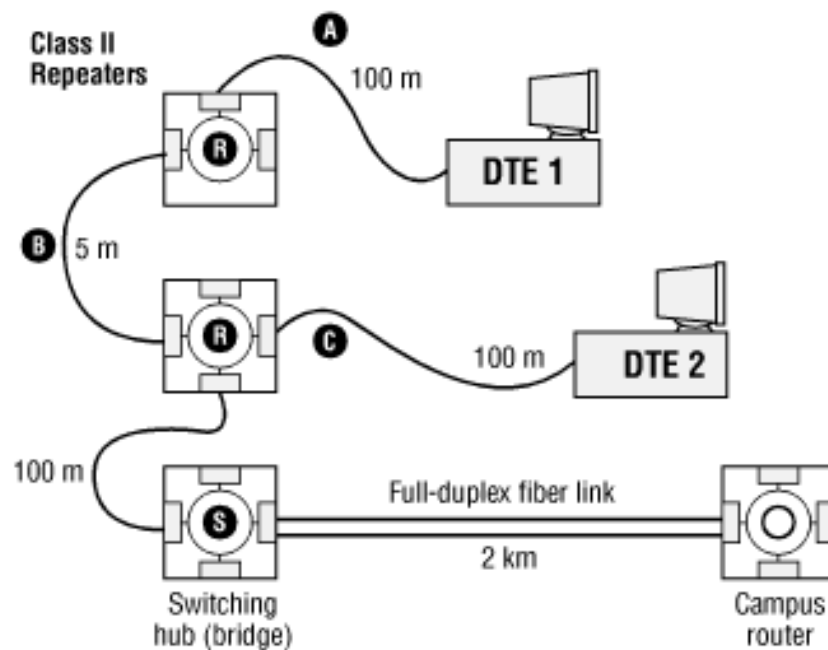


Рис. 4.1. Пример допустимой конфигурации сети Fast Ethernet

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. 4.1, считается конечным 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. 4.3 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 4.3 - Время двойного оборота сети (рис. 4.1)

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

### Задание для выполнения

Требуется оценить работоспособность 100-мегабитной сети Fast Ethernet в соответствии с первой и второй моделями. Конфигурации сети приведены в табл. 4.4. Топология сети представлена на рис. 4.2–4.3.

Таблица 4.4 - Варианты заданий

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

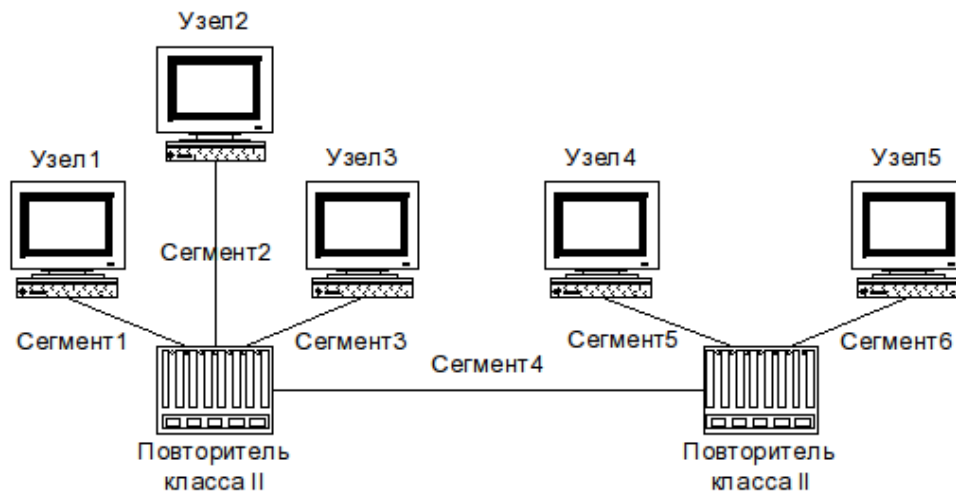


Рис. 4.2. Топология сети 1

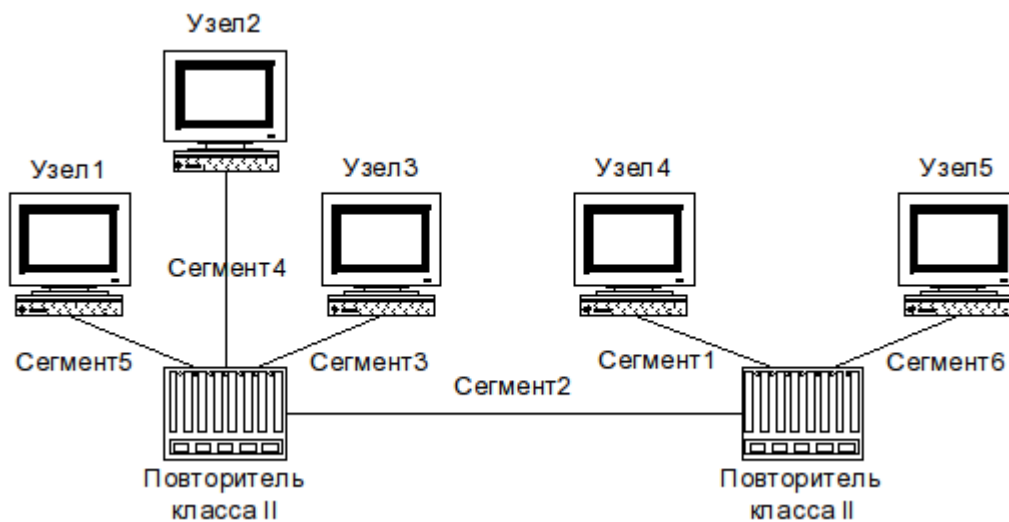


Рис. 4.3. Топология сети 2

### Контрольные вопросы

1. В чем заключается метод доступа CSMA/CD?

2. Что представляет собой первая модель сети Fast Ethernet?
3. Что представляет собой вторая модель сети Fast Ethernet?
4. Каковы значения предельно допустимого диаметра домена коллизий сети Fast Ethernet для различных комбинаций сегментов?
5. Как вычислить время двойного оборота в сети Fast Ethernet?

## **5. КОНФИГУРИРОВАНИЕ ЛВС**

Зачастую, настройка локальной сети в операционных системах Windows Vista, Windows 7, Windows Server 2008/2008 R2 начинается с такой области конфигурирования сетевых свойств, как компонент «Центр управления сетями и общим доступом». При помощи данного средства конфигурирования сетей можно выбирать сетевое размещение, просматривать карту сети, настраивать сетевое обнаружение, общий доступ к файлам и принтерам, а также настраивать и просматривать состояние текущих сетевых подключений.

### **Открытие компонента «Центр управления сетями и общим доступом»**

Чтобы открыть окно «Центр управления сетями и общим доступом» (рис. 5.1), необходимо выполнить одно из следующих действий:

- В области уведомлений нажать правой кнопкой мыши на значок «Сеть» и из контекстного меню выбрать команду «Центр управления сетями и общим доступом»;
- Нажать на кнопку «Пуск» для открытия меню, выделить элемент «Сеть» и нажать на нем правой кнопкой мыши. Из контекстного меню выбрать команду «Свойства»;
- Нажать на кнопку «Пуск» для открытия меню, открыть «Панель управления», из списка компонентов панели управления выбрать категорию «Сеть и Интернет», а затем перейти по ссылке «Центр управления сетями и общим доступом»;

- Нажать на кнопку «Пуск» для открытия меню, в поле поиска ввести Центр управления и в найденных результатах открыть приложение «Центр управления сетями и общим доступом»;
- Воспользоваться комбинацией клавиш **Win+R** для открытия диалога «Выполнить». В диалоговом окне «Выполнить», в поле «Открыть» ввести  
`%windir%\system32\control.exe /name Microsoft.NetworkAndSharingCenter` и нажать на кнопку «ОК».

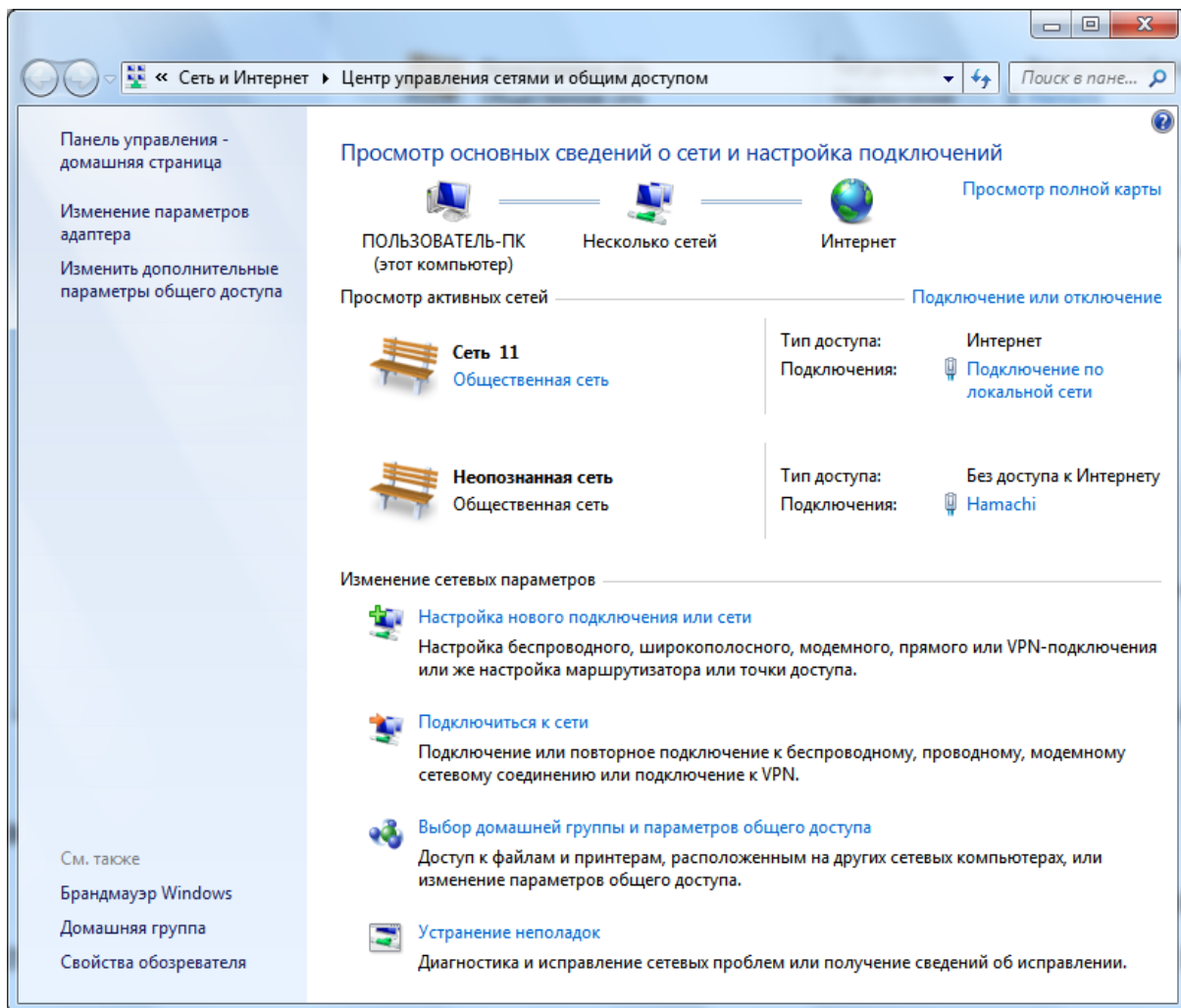


Рис. 5.1. Центр управления сетями и общим доступом

## Понятие сетевого расположения

Параметр «сетевое расположение» задается для компьютеров при первом подключении к сети и во время подключения автоматически настраивается брандмауэр и параметры безопасности для того типа сети, к которому производится подключение. В отличие от

операционной системы Windows Vista, где для всех сетевых подключений используется самый строгий профиль брандмауэра для сетевого размещения, операционная система Windows 7 поддерживает несколько активных профилей, что позволяет наиболее безопасно использовать несколько сетевых адаптеров, подключенных к различным сетям. Существует четыре типа сетевого расположения (рис.5.2).

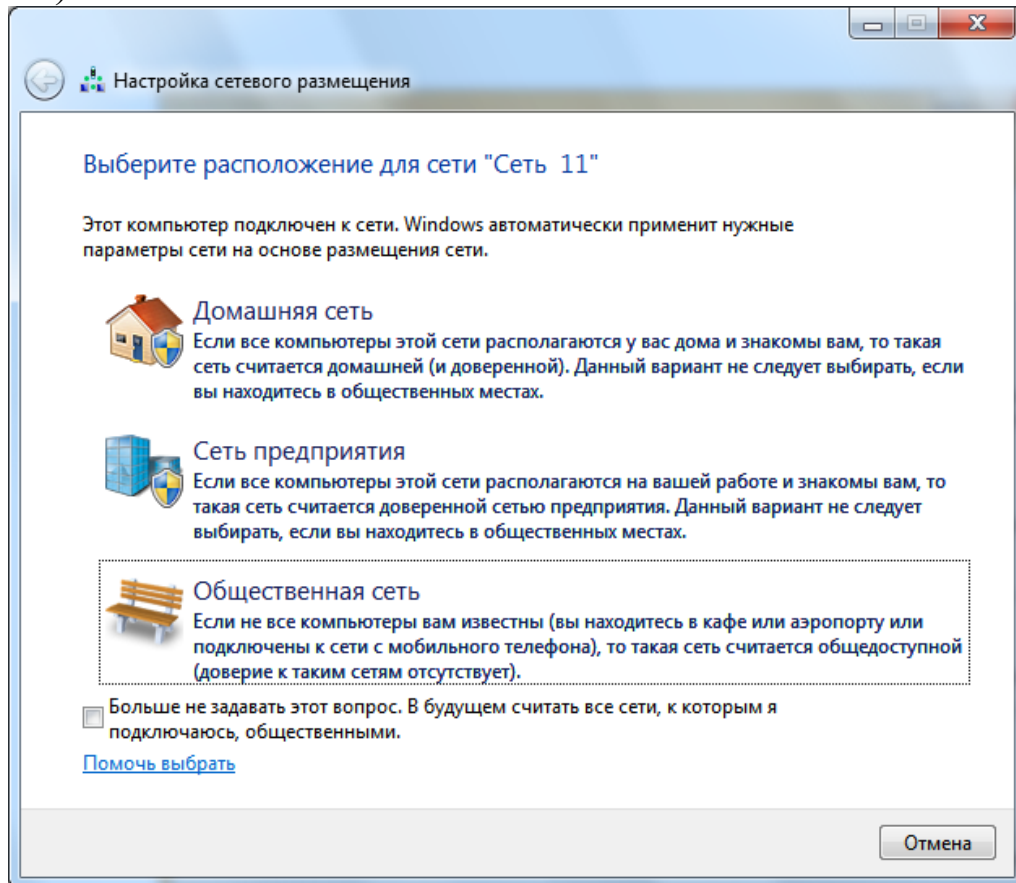


Рис. 5.2. Выбор сетевого расположения

**Домашняя сеть.** Данное сетевое расположение предназначено для использования компьютера в домашних условиях или в таких сетях, где пользователи очень хорошо знают друг друга. Такие компьютеры могут создавать и присоединяться к домашним группам. Для домашних сетей автоматически включается обнаружение сети.

**Сеть предприятия.** Такое сетевое расположение используется в сети малого офиса (SOHO). Для этого сетевого расположения также включено обнаружение сети, но вы не можете ни создавать, ни присоединять компьютер к домашней группе.

**Общественная сеть.** Это сетевое расположение предназначено для использования компьютера в таких общественных местах, как



кафе или аэропорты. Это наиболее строгое размещение, у которого по умолчанию отключены возможности присоединения к домашней группе и сетевое обнаружение.

**Доменная сеть.** Если компьютер присоединён к домену Active Directory, то существующей сети будет автоматически назначен тип сетевого размещения «Домен». Доменный тип сетевого расположения аналогичен рабочей сети, за исключением того, что в домене конфигурация брандмауэра Windows, сетевого обнаружения, а также сетевой карты определяется групповой политикой.

Каким образом связаны компьютеры в сети, можно просматривать с помощью карты сети. Однако этот компонент доступен не для всех типов сетевого расположения.

### **Карта сети**

Карта сети – это графическое представление расположения компьютеров и устройств, которое позволяет увидеть все устройства вашей локальной сети, а также схему их подключения друг к другу. В окне «Центр управления сетями и общим доступом» отображается только локальная часть сетевой карты, компоновка которой зависит от имеющихся сетевых подключений. Компьютер, на котором выполняется создание карты, отображается в левом верхнем углу. Другие компьютеры подсети отображаются слева. Такие устройства инфраструктуры, как коммутаторы, концентраторы и шлюзы в другие сети отображаются справа. Сетевое сопоставление работает в проводных и беспроводных сетях, однако только в частных и доменных сетях. Просмотреть карту публичной сети невозможно. Протокол LLTD обеспечивает сопоставление только компьютеров в одной подсети, которая является обычной установкой в домашних или малых офисах.

Можно заметить, что некоторые компьютеры и устройства отображаются отдельно в нижней части окна «Карта сети» либо могут вообще отсутствовать. Например, если сервер печати беспроводной сети поддерживает технологию UPnP, а не LLTD, то он будет располагаться в нижней части окна «Карта сети». Подобная ситуация возникает, поскольку не все операционные системы и устройства предполагают поддержку протокола LLTD или вследствие возможной неправильной настройки устройств. Пример карты сети представлен на рис. 5.3.



Рис. 5.3. Пример карты сети

За работу карты сети в операционных системах отвечают два компонента:

- Обнаружение топологии связи Link Layer (Link Layer Topology Discover Mapper – LLTD Mapper) – компонент, который запрашивает в сети устройства для включения их в карту;
- Отвечающее устройство LLTD (Link Layer Topology Discover Responder – LLTD Responder) – компонент, который отвечает за запросы компонента LLTD Mapper.

По умолчанию, карту сети можно просматривать только для расположений «Домашняя сеть» или «Сеть предприятия». При попытке просмотра сетевой карты для расположений «Доменная сеть» или «Общественная сеть» появляется сообщение о невозможности отображения карты (рис. 5.4).

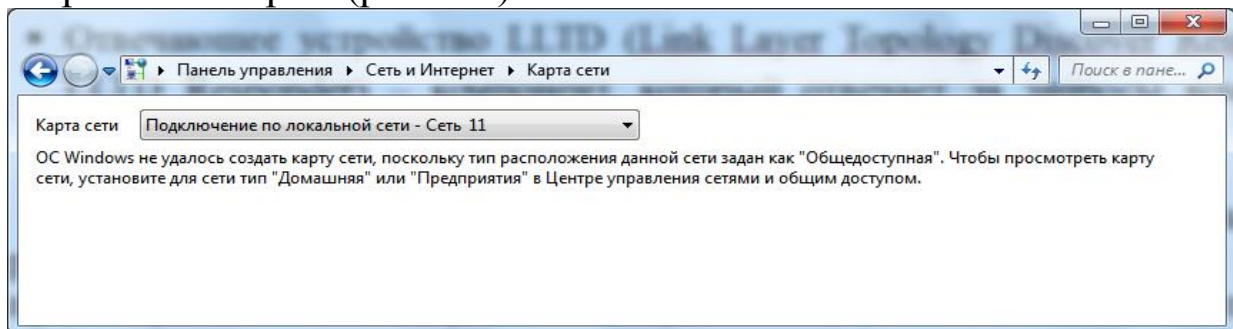


Рис. 5.4. Попытка просмотра карты сети

Для того чтобы включить сетевое сопоставление в доменной сети, нужно на контроллере домена выполнить следующие действия:

1. Открыть оснастку «Управление групповой политики»;

2. Выбрать объект групповой политики (например, Default Domain Policy, область действия – весь домен), который будет распространяться на компьютер, расположенный в доменной сети, нажать на нем правой кнопкой мыши и из контекстного меню выбрать команду «Изменить»;
3. В оснастке «Редактор управления групповыми политиками» развернуть узел Конфигурация компьютера/Политики/Административные шаблоны/Сеть/Обнаружение топологии связи (Link Layer) и выбрать политику «Включает драйвер отображения ввода/вывода (LLTDIO)»;
4. В свойствах параметра политики установить переключатель на опцию «Включить» и установить флажок «Разрешить операцию для домена»;
5. Повторь аналогичные действия для параметра политики «Включить драйвер «Ответчика» (RSPNDR)»;
6. Обновить параметры политики на клиентской машине, используя команду `gpupdate /force /boot`;
7. Обновите карту сети.

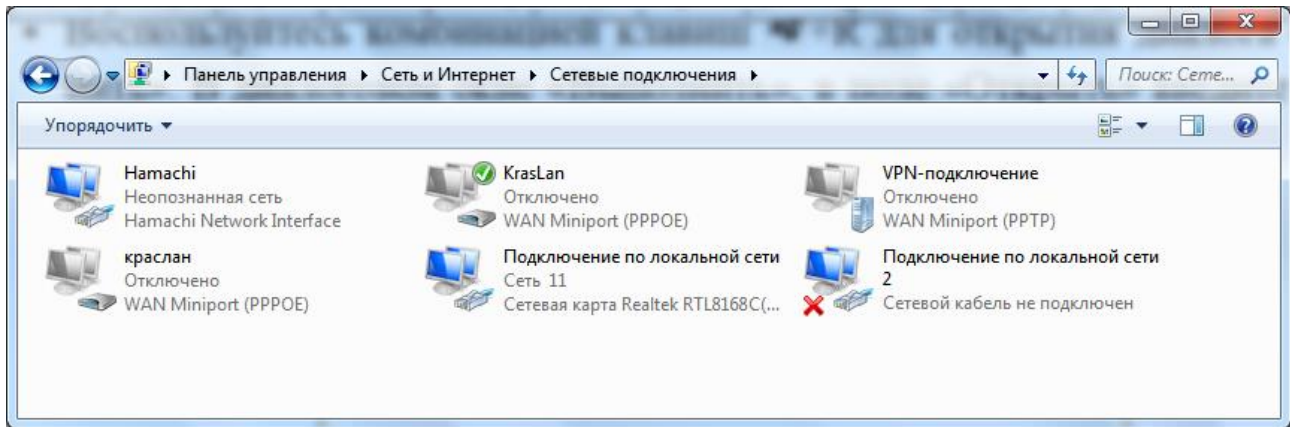
## **Сетевые подключения**

После установки драйвера для каждого сетевого адаптера, операционная система Windows пытается автоматически сконфигурировать сетевые подключения на локальном компьютере. Все доступные сетевые подключения отображаются в окне «Сетевые подключения» (рис. 5.5). Сетевое подключение представляет собой набор данных, необходимых для подключения компьютера к Интернету, локальной сети или любому другому компьютеру.

Открыть окно «Сетевые подключения» можно любым из следующих способов:

- Открыть окно «Центр управления сетями и общим доступом» и перейти по ссылке «Изменение параметров адаптера»;
- Нажать на кнопку «Пуск» для открытия меню, в поле поиска ввести «Просмотр сетевых» и в найденных результатах открыть приложение «Просмотр сетевых подключений»;

- Воспользоваться комбинацией клавиш **Win+R** для открытия диалога «Выполнить». В диалоговом окне «Выполнить», в поле «Открыть» ввести `ncpa.cpl` или `control netconnection` и нажать на кнопку «ОК».



**Рис. 5.5. Окно «Сетевые подключения»**

При выборе любого сетевого подключения можно выполнить с ним следующие действия:

- **Переименование подключения.** Операционная система по умолчанию назначает всем сетевым подключениям имена «Подключение по локальной сети» или «Подключение к беспроводной сети» и номер подключения в том случае, если существует более одного сетевого подключения. При желании, можно переименовать любое сетевое подключение одним из трех следующих способов:
  - Нажать на клавишу F2, ввести новое имя сетевого подключения, после чего нажать на клавишу Enter;
  - Нажать правой кнопкой мыши на переименовываемом сетевом подключении и из контекстного меню выбрать команду «Переименовать». Ввести новое имя сетевого подключения, после чего нажать на клавишу Enter;
  - Выбрать сетевое подключение и нажать на кнопку «Переименование подключения», которая расположена на панели инструментов, после чего ввести новое имя сетевого подключения и нажать на клавишу Enter.
- **Состояние сети.** Используя данное окно (рис. 5.6), можно просмотреть любые данные о состоянии сетевого подключения и такие детали, как IP-адрес, MAC-адрес и прочее. Чтобы открыть диалоговое

окно сведений о сетевом подключении, необходимо выполнить следующие действия:

1. Открыть диалоговое окно «Состояние» одним из следующих способов:
  - нажать правой кнопкой мыши на сетевом подключении и из контекстного меню выбрать команду «Состояние»;
  - выбрать сетевое подключение и нажать на кнопку «Просмотр состояния подключения», которая расположена на панели инструментов;
  - выбрать сетевое подключение и нажать на клавишу Enter.

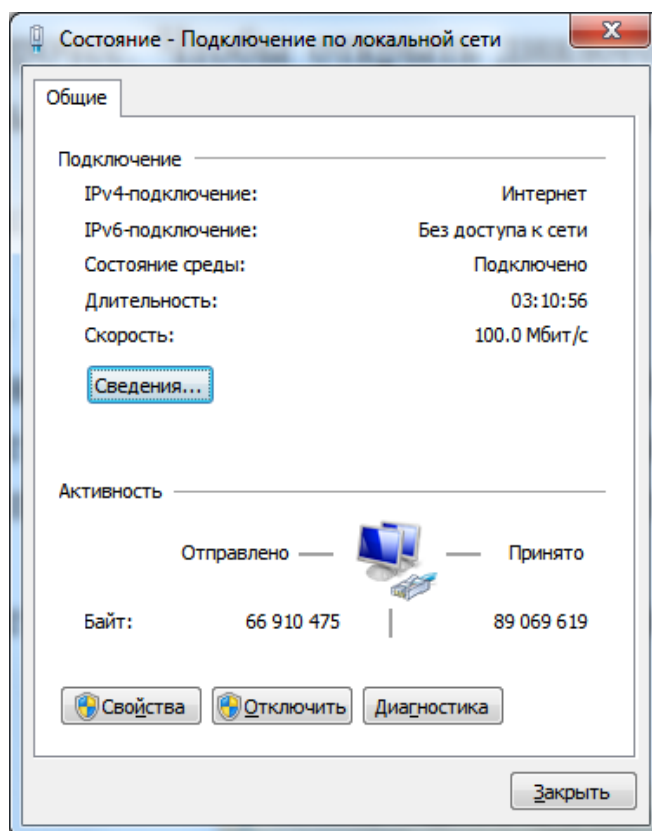


Рис. 5.6. Диалоговое окно состояния подключения по локальной сети

2. В окне «Состояние – подключение по локальной сети» нажать на кнопку «Сведения». В диалоговом окне «Сведения о сетевом подключении», отображенном на рис. 5.7, можно просмотреть подробные сведения о текущем сетевом подключении.

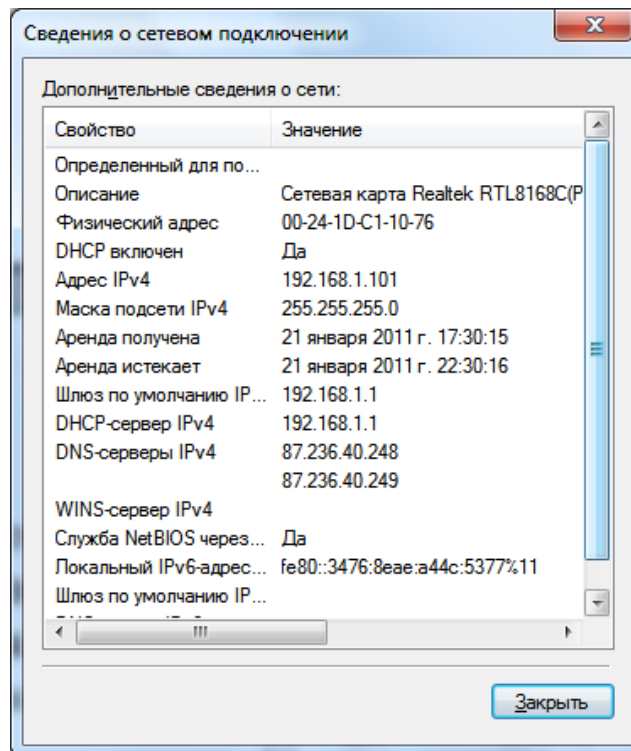


Рис. 5.7. Сведения о сетевом подключении

- **Диагностика подключения.** В случае обнаружения проблем в работе вашего сетевого подключения, окно «Сетевые подключения» предлагает средство диагностики «Устранение неполадок», которое содержит возможность решения при помощи анализа подключения. Для того чтобы воспользоваться данным средством, необходимо выполнить любое из следующих действий:

- нажать правой кнопкой мыши на сетевом подключении и из контекстного меню выбрать команду «Диагностика» (рис. 5.8).
- выбрать сетевое подключение и нажать на кнопку «Диагностика подключения», которая расположена на панели инструментов.

В открывшемся диалоговом окне «Диагностика сетей Windows» для устранения неполадок необходимо следовать действиям мастера.

- **Отключение сетевого устройства.** Иногда проблемы с сетевыми подключениями решаются посредством отключения сетевого адаптера компьютера от сети. Для того чтобы отключить сетевой адаптер необходимо выполнить одно из следующих действий:

- нажать правой кнопкой мыши на сетевом подключении и из контекстного меню выбрать команду «Отключить»;
- выбрать сетевое подключение и нажать на кнопку «Отключение сетевого устройства», которая расположена на панели инструментов.

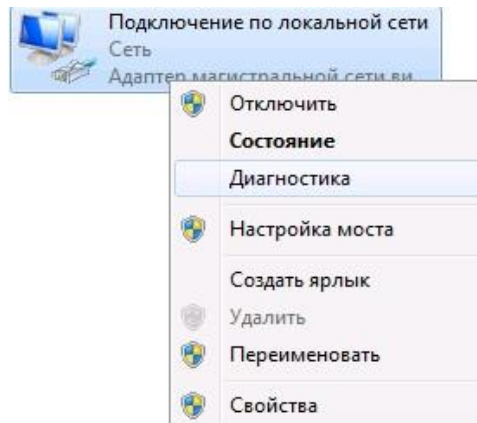


Рис. 5.8. Открытие мастера устранения неполадок подключения по локальной сети

- **Настройка параметров подключения.** Как таковые, сетевые подключения не позволяют осуществлять коммуникации. Осуществление коммуникаций обеспечивают сетевые клиенты, службы и протоколы, которые привязаны к созданным сетевым подключениям (рис. 5.9).

Для того чтобы изменить настройки сетевого подключения, можно воспользоваться средствами настройки параметров подключения. Для изменения компонентов и настроек сетевого подключения необходимо выполнить следующие действия:

- нажать правой кнопкой мыши на сетевом подключении и из контекстного меню выбрать команду «Свойства»;
- выбрать сетевое подключение и нажать на кнопку «Настройка параметров подключения», которая расположена на панели инструментов;
- выбрать сетевое подключение и воспользоваться комбинацией клавиш Alt + Enter.

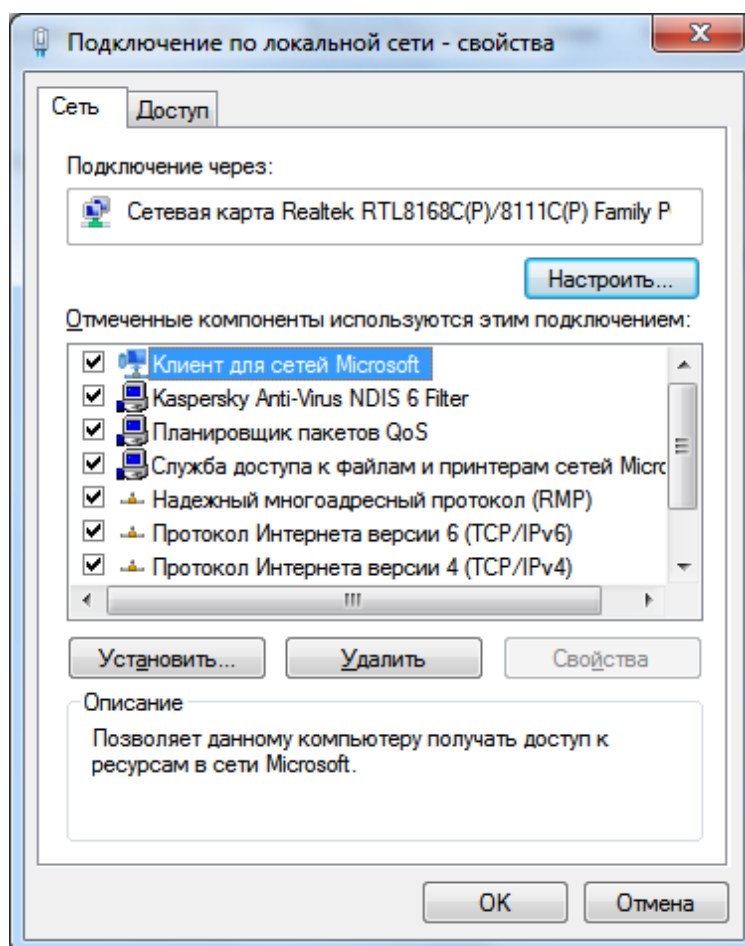


Рис. 5.9. Диалоговое окно свойств сетевого подключения

Установленные возле компонентов флажки указывают, что эти компоненты привязаны к подключению.

**Задание:** Выполнить конфигурирование сети рабочей станции.

1. Просмотреть карту сети.
2. Включить сетевое сопоставление в доменной сети.
3. Переименовать сетевое подключение.
4. Просмотреть состояние сети.
5. Выполнить диагностику подключения.
6. Настроить параметры подключения.

### Контрольные вопросы

1. Какое назначение у компонента «Центр управления сетями и общим доступом»?

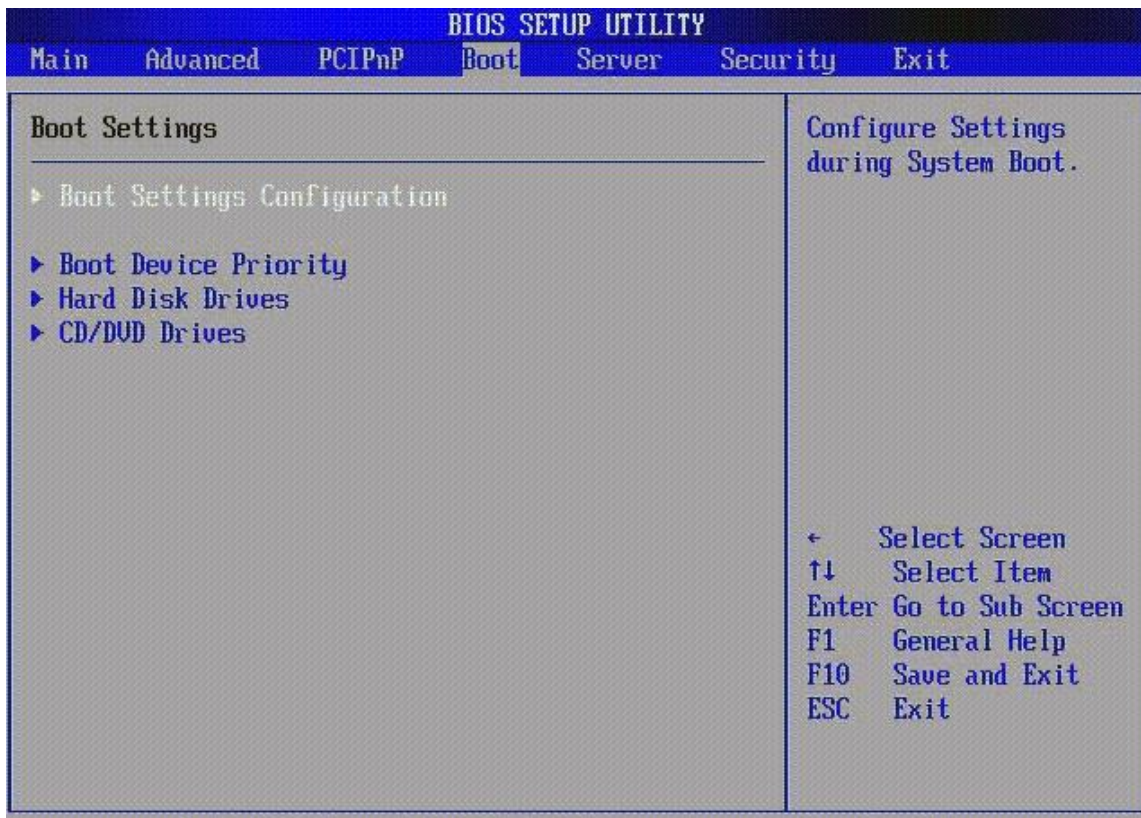


2. Продемонстрировать, какие существуют способы открытия компонента «Центр управления сетями и общим доступом»?
3. Охарактеризовать типы сетевого расположения.
4. Что представляет собой карта сети?
5. Какие протоколы отвечают за построение карты сети?
6. В каких ситуациях просмотр карты сети будет не возможен?
7. С каким аппаратным сетевым компонентом связываются свойства доступных сетевых подключений?
8. Какие существуют способы открытия окна «Сетевые подключения»?
9. Какие действия пользователь обычно может выполнить по отношению к любому сетевому подключению?
10. Охарактеризовать свойства сведений о сетевом подключении для компьютера. Объяснить значения данных свойств.
11. Какими способами можно вызвать средства диагностики подключения?

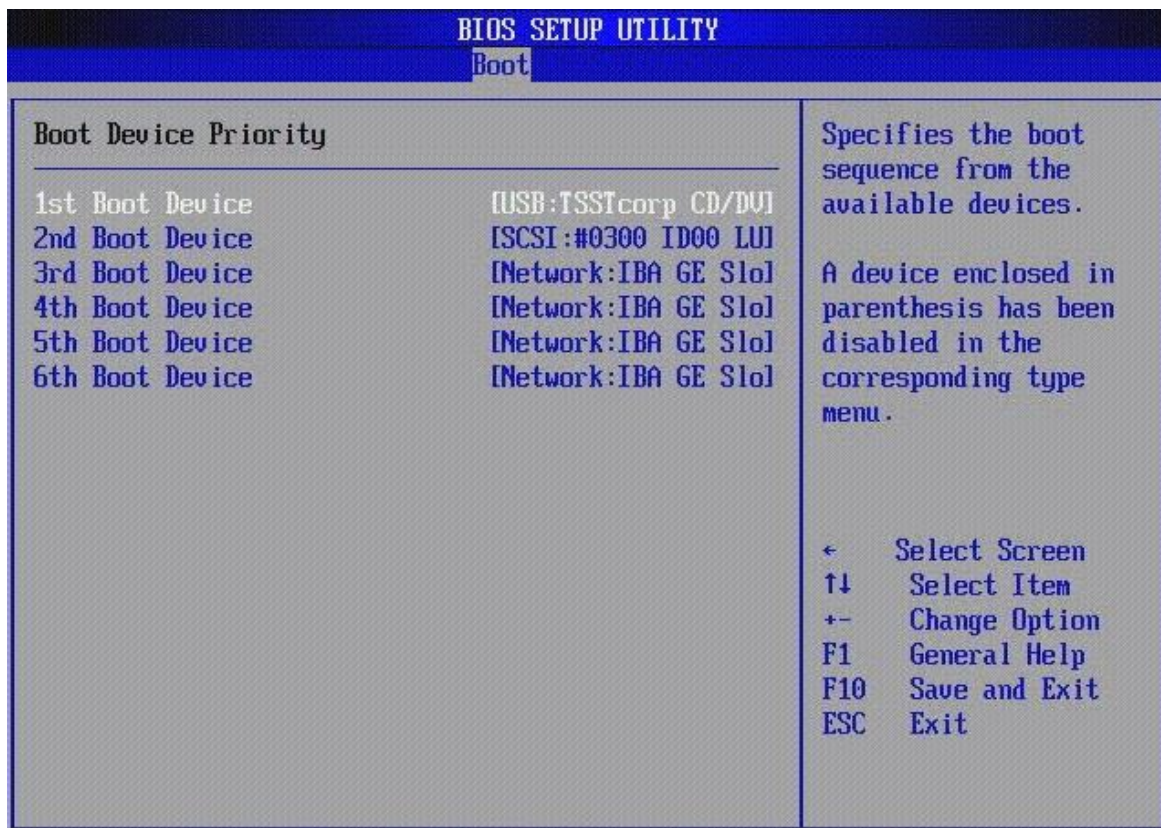
## **6. УСТАНОВКА И НАСТРОЙКА СЕТЕВОЙ ОС**

### **ПОРЯДОК ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ ПО УСТАНОВКЕ WINDOWS SERVER 2016**

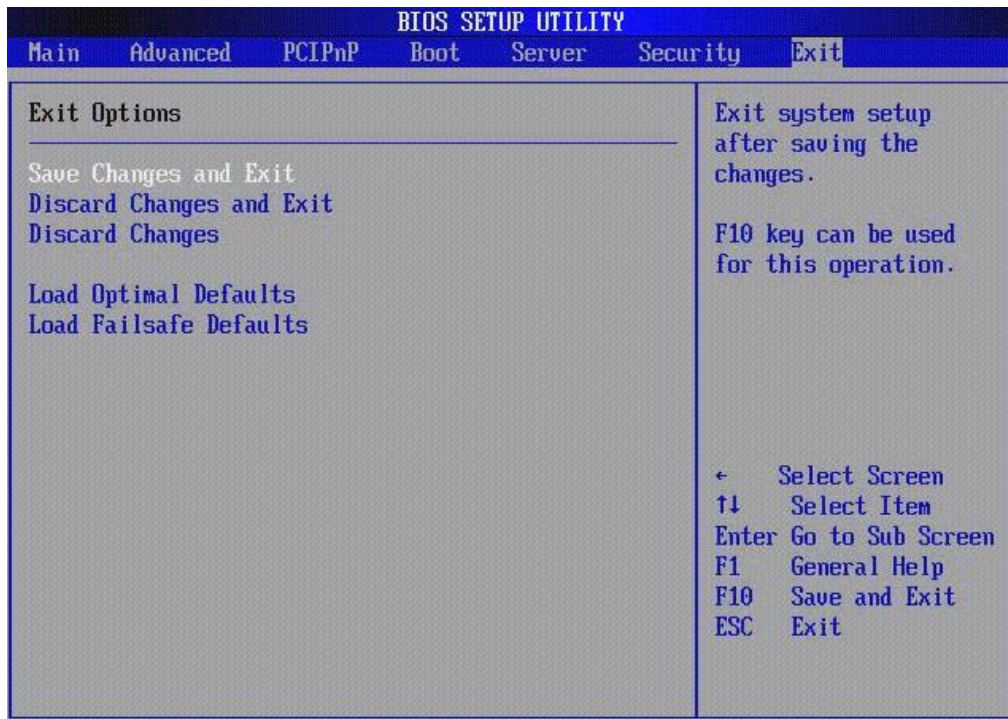
Для того чтобы начать установку Windows Server 2016, нужно загрузиться с установочного DVD диска Windows Server 2016. Для этого необходимо в BIOS сервера выбрать первым в списке загрузок CD/DVD привод или воспользоваться возможностями Boot Menu. На каждом сервере кнопки вызова BIOS и Boot Menu отличаются, но обычно при загрузке сервера на экране пишется, какие клавиши нужно нажать, для того чтобы попасть в BIOS или в Boot Menu. Внешний вид BIOS на разных серверах может отличаться, но идеология остается прежней. Нужно перейти на вкладку “Boot” и выбрать пункт “Boot Device Priority”.



Далее необходимо переместить “CD/DVD привод” на самый верх.



Сохранить сделанные изменения. Для этого нужно перейти на вкладку “Exit” и выбрать пункт “Save Changes and Exit”.



Далее необходимо подтвердить сохранение новой конфигурации.

Нажать “OK”.

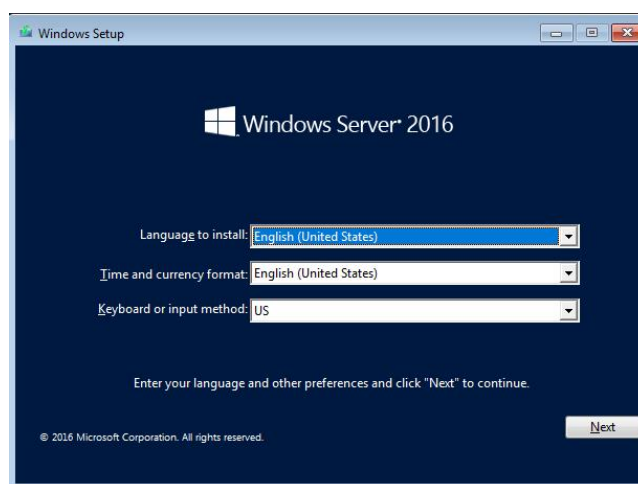




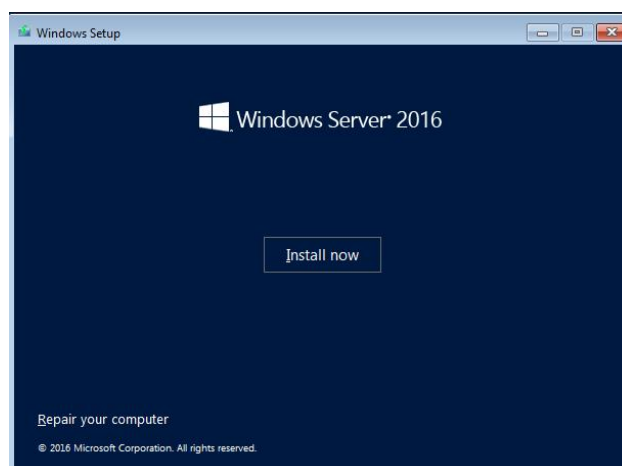
Теперь при последующей перезагрузке компьютера, можно загрузиться с установочного DVD диска Windows Server 2016. Для этого потребуется нажать любую клавишу на клавиатуре, когда появится сообщение “Press any key to boot from CD or DVD”.

A black terminal window with white text that reads "Press any key to boot from CD or DVD.\_".

После успешной загрузки с установочного DVD диска Windows Server 2016, необходимо выбрать региональные параметры. В примере используется англоязычная версия Windows Server 2016, поэтому можно использовать параметры по умолчанию. Нажать на кнопку “Next”.



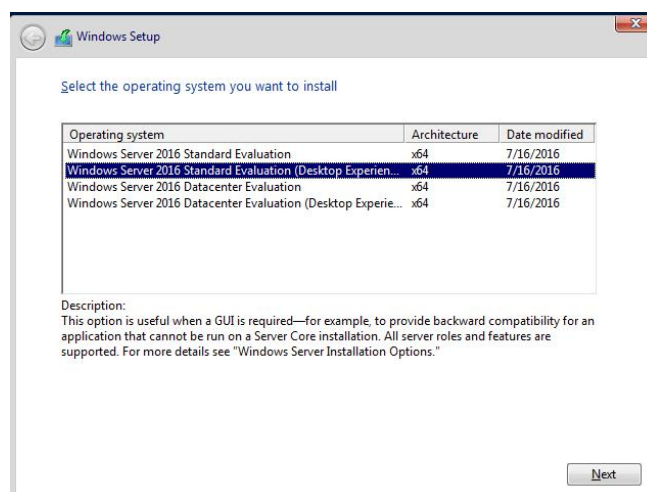
Нажать на кнопку “Install now”.



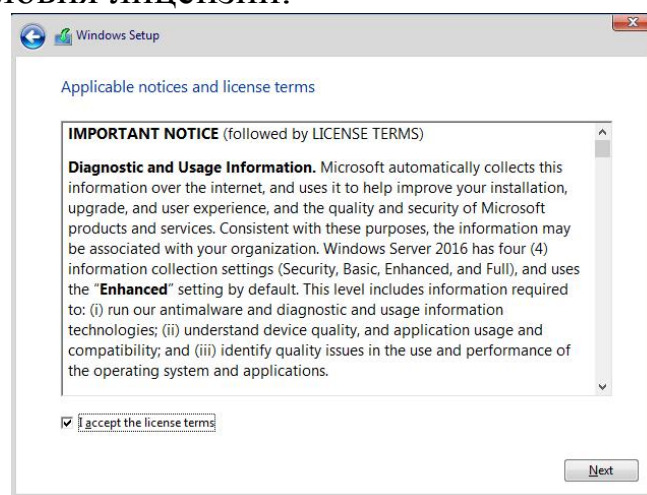
Если необходимо установить Windows Server 2016 без графического интерфейса, то нужно выбрать “Windows Server 2016 Standard Evaluation”.

В данном примере рассматривается установка Windows Server 2016 с графическим интерфейсом.

Выбрать “Windows Server 2016 Standard Evaluation (Desktop Experience)” и нажать “Next”.



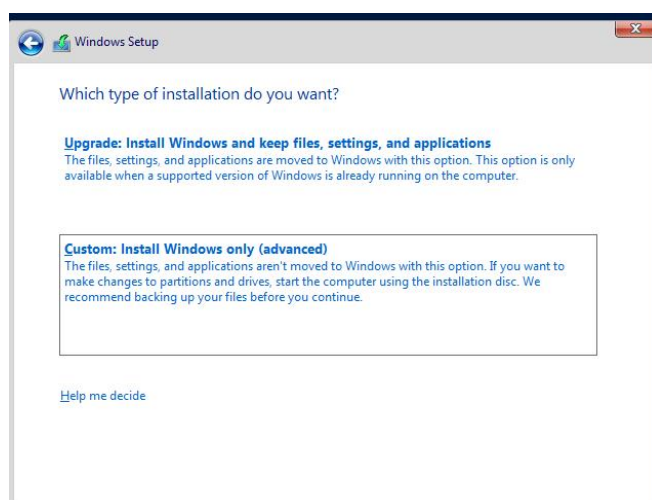
Принять условия лицензии.



На этом шаге предлагается два варианта установки:

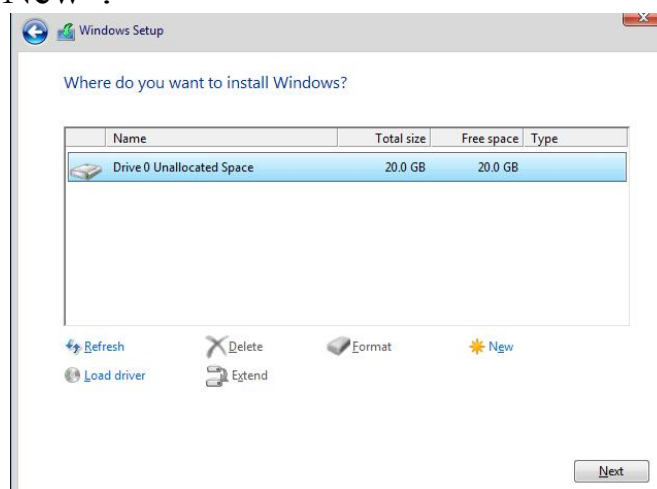
1. “Upgrade”. Как показывает практика, многочисленные программы могут быть не совместимы с новой операционной системой, и кроме того есть вероятность сохранить проблемы старой операционной системы, таким, образом потеряв всякую стабильность.

2. “Custom”. Он позволяет начать работу с системой “с чистого листа”, таким образом, после установки получив максимальное быстродействие и стабильность. Останется только установить драйвера и привычное для работы программное обеспечение.



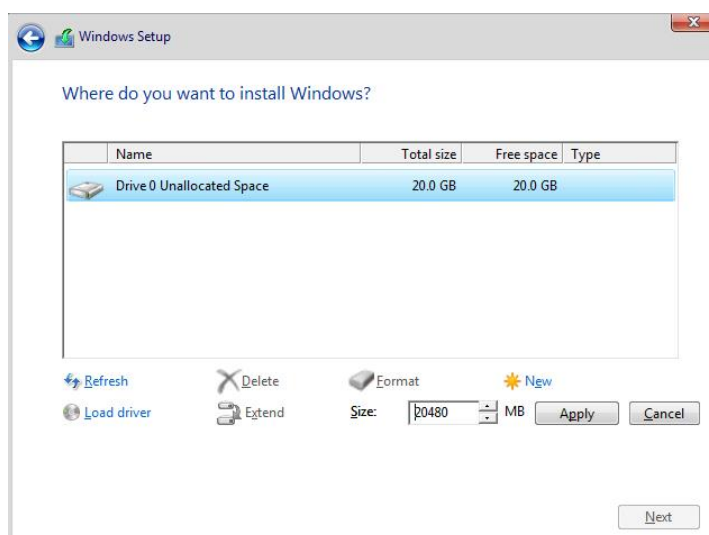
Теперь необходимо выбрать, на какой диск будет установлена новая операционная система, и выделить место для установки. В случае если установлено более одного диска или на диске уже имеется несколько разделов, все это будет отображаться на данном этапе. Необходимо соблюдать осторожность и заранее понимать, на какой раздел следует установить операционную систему. В данном примере установлен один диск объемом 60Gb.

Выбрать пункт “New”.



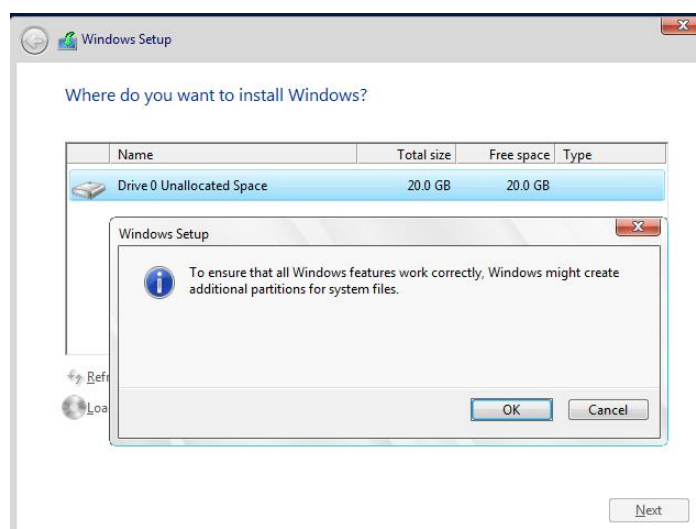
В данном случае под систему будет выделено все свободное место на диске, поэтому оставить значение в разделе “Size” по умолчанию.

Нажать на кнопку “Apply”.



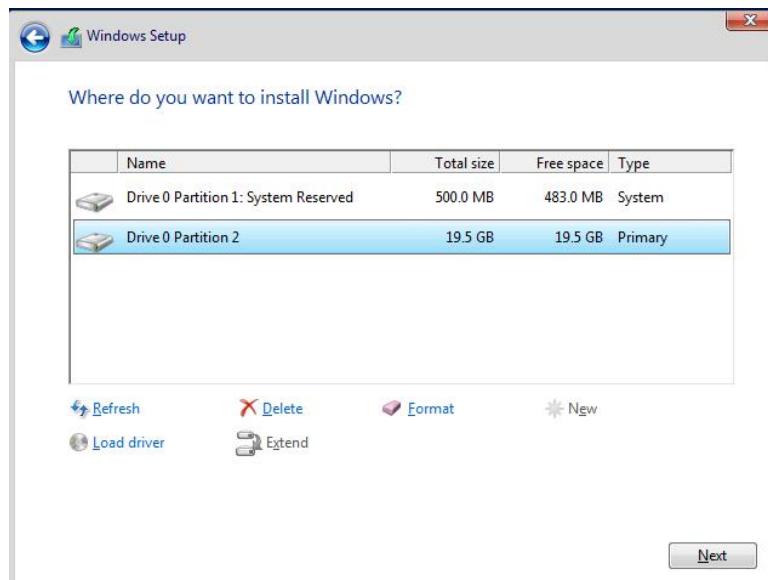
Операционная система уведомляет о том, что ей может понадобиться создать дополнительные разделы на диске для хранения системных файлов.

Нажать на кнопку “OK”.



Таким образом, под операционную систему было выделено все свободное место на диске, но в то же время система зарезервировала для себя небольшой раздел.

Теперь необходимо выбрать раздел, на который предполагается установить операционную систему, и нажать на кнопку “Next”.



Начался процесс установки операционной системы.



Компьютер автоматически перезагрузится несколько раз.

После завершения установки, операционная система начнет подготавливать сервер к работе.

Теперь нужно указать надежный пароль для учетной записи “Administrator”.

Нажать на кнопку “Finish”.





Customize settings

Type a password for the built-in administrator account that you can use to sign in to this computer.

User name

Password

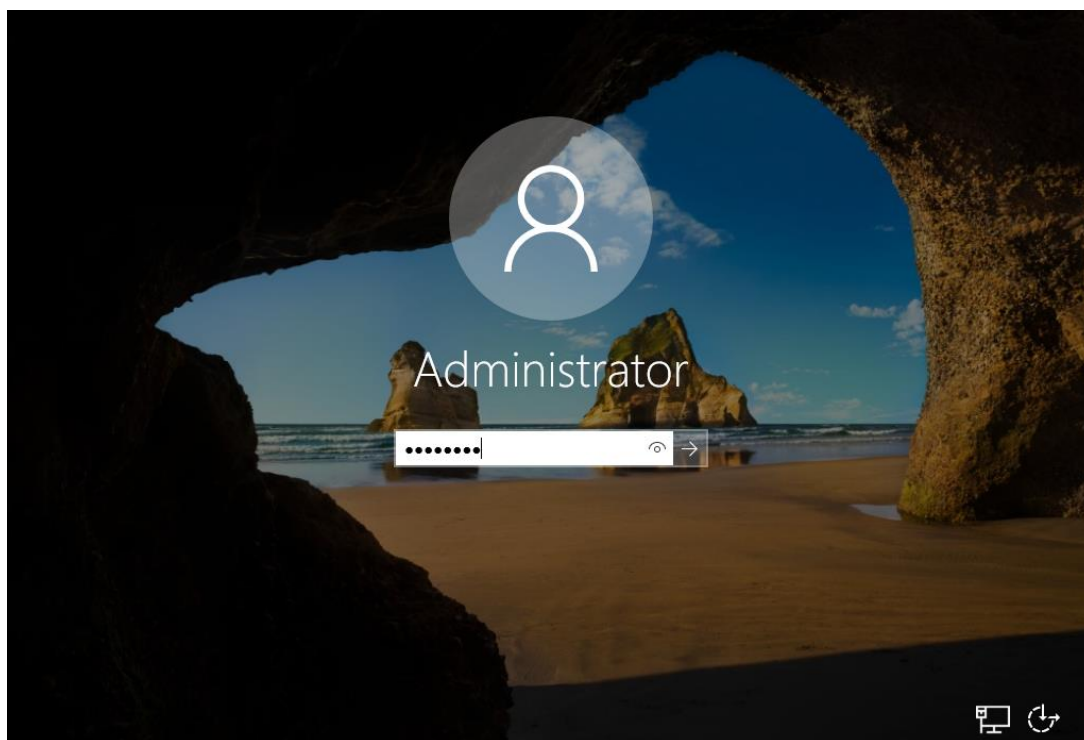
Reenter password  

 Finish

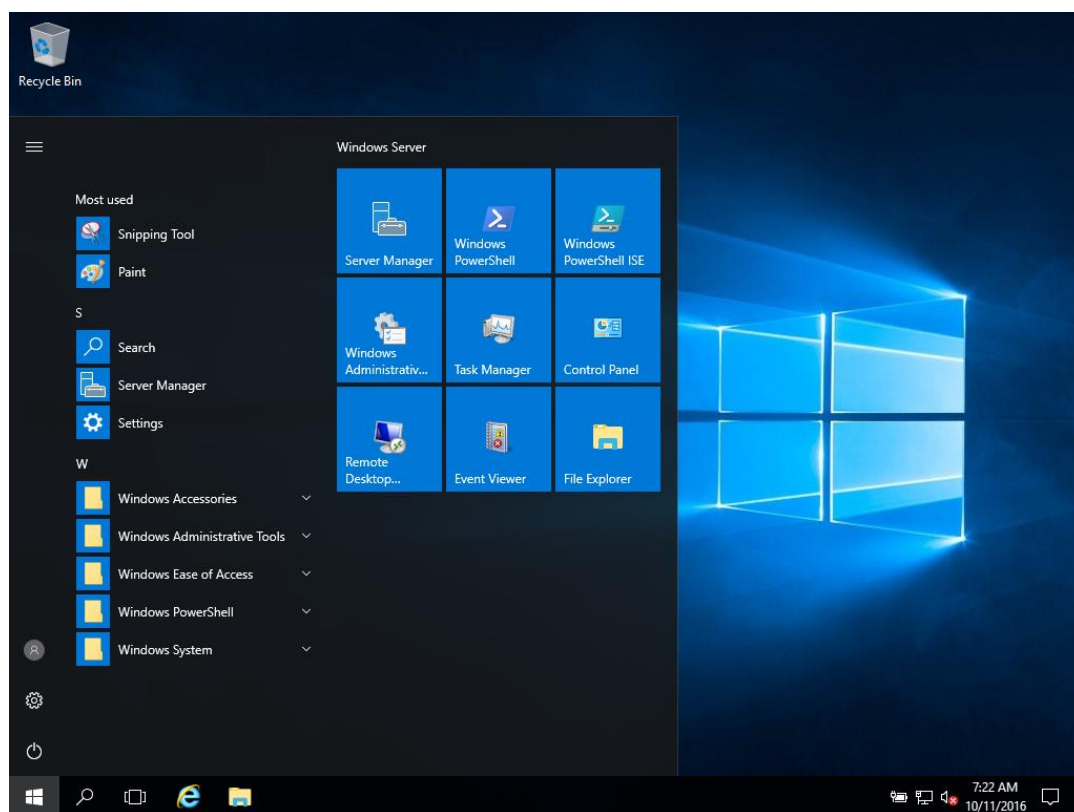
После завершения настроек, появится экран блокировки Windows Server 2016. Нажать “Ctrl+Alt+Delete”.



Далее необходимо авторизоваться под учетной записью “Administrator”.



Установка Windows Server 2016 завершена.



Теперь нужно установить драйвера.

Ссылки на раздел техподдержка популярных производителей:

IBM – <http://www.ibm.com/support>

Dell – <http://support.dell.com>

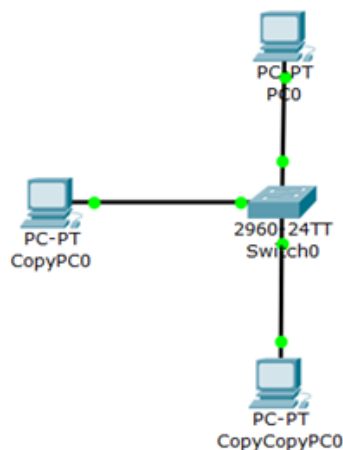
HP – <http://www8.hp.com/ru/ru/support-drivers.html>

### Контрольные вопросы

1. Каким образом можно загрузиться с установочного DVD диска Windows Server 2016?
2. Какие существуют два варианта установки Windows Server 2016?
3. Как зарезервировать место для установки Windows Server 2016?

## 7. МОДЕЛИРОВАНИЕ И АНАЛИЗ ХАРАКТЕРИСТИК ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Рассмотрим сеть на базе коммутатора (рис. 7.1).



**Рис. 7.1. Звезда на базе коммутатора модели 2960**

На вкладке Physical можно посмотреть вид коммутатора, имеющего 24 порта Fast Ethernet и 2 порта Gigabit Ethernet (рис. 7.2).

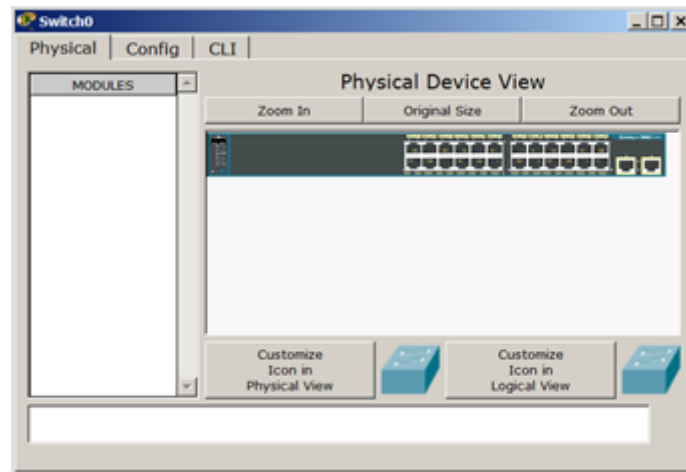



Рис. 7.2. Физический внешний вид коммутатора модели 2960

IPv4	IPv6	Misc
<input type="checkbox"/> ARP	<input type="checkbox"/> BGP	<input type="checkbox"/> DHCP
<input type="checkbox"/> DNS	<input type="checkbox"/> EIGRP	<input type="checkbox"/> HSRP
<input checked="" type="checkbox"/> ICMP	<input type="checkbox"/> OSPF	<input type="checkbox"/> RIP

В режиме *Simulation* настроить фильтры и с помощью функции  можно просмотреть прохождение пакета между двумя ПК через коммутатор. Маршруты пакета в концентраторе и коммутаторе будут разными: как в прямом, так и в обратном направлении хаб отправляет всем, а коммутатор – только одному.

### Задание 7.1. Создание и тестирование простой сети

Произведите проектирование локальной сети из хаба, коммутатора и 4х ПК. Сеть, которую необходимо спроектировать, представлена на рис. 7.3.

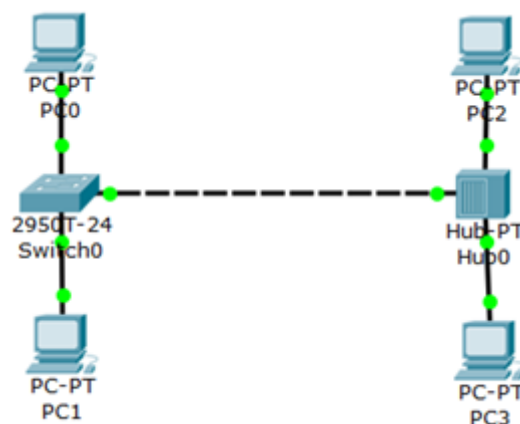


Рис. 7.3. Проектируемая сеть

Произвести настройку и диагностику этой сети двумя способами: утилитой *ping* и в окне списка *PDU*. Убедиться в успешности работы сети в режиме симуляции.

### Примечание

Перед выполнением симуляции необходимо задать фильтрацию пакетов. Для этого нужно нажать на кнопку "Изменить фильтры", откроется окно, в котором нужно оставить только протоколы "ICMP" и "ARP". Кнопка "Авто захват/Воспроизведение" подразумевает моделирование всего *ping*-процесса в едином процессе, тогда как "Захват/Вперед" позволяет отображать его пошагово.

### Задание 7.2. Исследование качества передачи трафика по сети

При исследовании пропускной способности *ЛВС* (качества передачи трафика по сети) желательно увеличить размер пакета и отправлять запросы с коротким интервалом времени, не ожидая ответа от удаленного узла, для того, чтобы создать серьезную нагрузку на сеть. Однако утилита *ping* не позволяет отправлять эхо-запрос без получения эхо-ответа на предыдущий запрос и до истечения времени ожидания. Поэтому для организации существенного трафика необходимо воспользоваться программой *Traffic Generator*. Для работы создать и настроить следующую сеть (рис. 7.4).

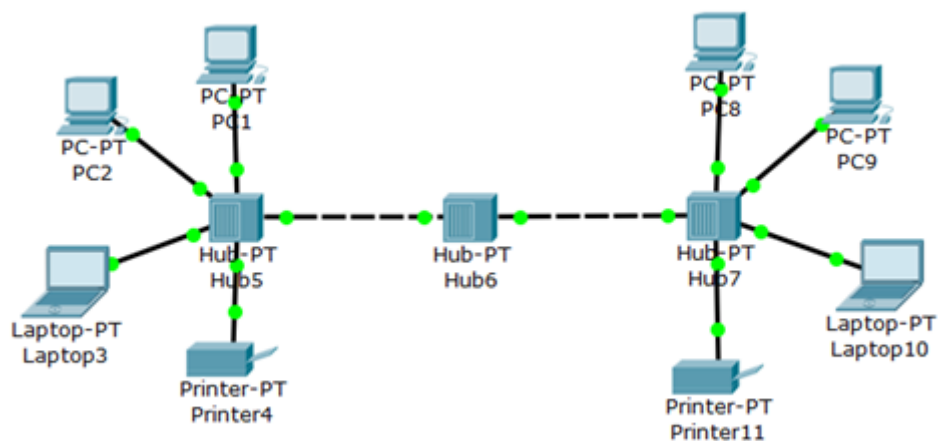


Рис. 7.4. Топология сети для исследования качества передачи

В окне управления PC1 во вкладке Desktop выбрать приложение *Traffic Generator* и задать настройки, как на рис. 7.5 для передачи

трафика от PC1 на PC8. Для ясности рядом с английской версией окна показан тот же текст в русской версии программы CPT.

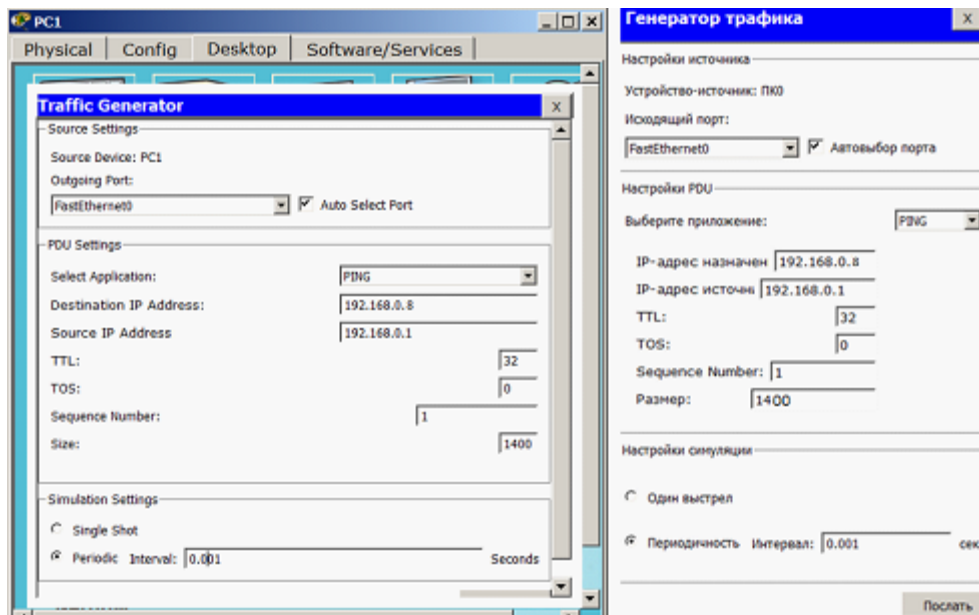


Рис. 7.5. Настройка генератора трафика (Вариант трафика от PC1 до PC8)

Итак, при помощи протокола ICMP формируется трафик между компьютерами PC1 с адресом 192.168.0.1 и PC8 с адресом 192.168.0.8. При этом в разделе **Source Settings** (Настройки источника) необходимо установить флажок **Auto Select Port** (Автовыбор порта), а в разделе **PDU Settings** (настройки IP-пакета) задать следующие значения параметров этого поля:

**Select application:** *PING*

**Destination: IPAddress:** 192.168.0.8 (адрес получателя);

**Source IP Address:** 192.168.0.1 (*адрес* отправителя);

**TTL:** 32 (время жизни пакета);

**TOS:** 0 (тип обслуживания, "0" - обычный, без приоритета);

**Sequence Number:** 1 (начальное *значение* счетчика пакетов);

**Size:** 1400 (размер поля данных пакета в байтах);

**Simulations Settings** - здесь необходимо активировать *переключатель*;

**Periodic Interval:** 0.3 Seconds (период повторения пакетов)

После нажатия на кнопку **Send** (Послать) между PC1 и PC8 начнется активный обмен данными.

## Исследование качества работы сети

Для оценки качества работы сети необходимо передать поток пакетов между PC1 и PC8 при помощи команды *ping -n 200 192.168.0.8* и оценить качество работы сети по числу потерянных пакетов. Параметр "-n" позволяет задать количество передаваемых эхо-запросов (200 запросов) – рис. 7.6.

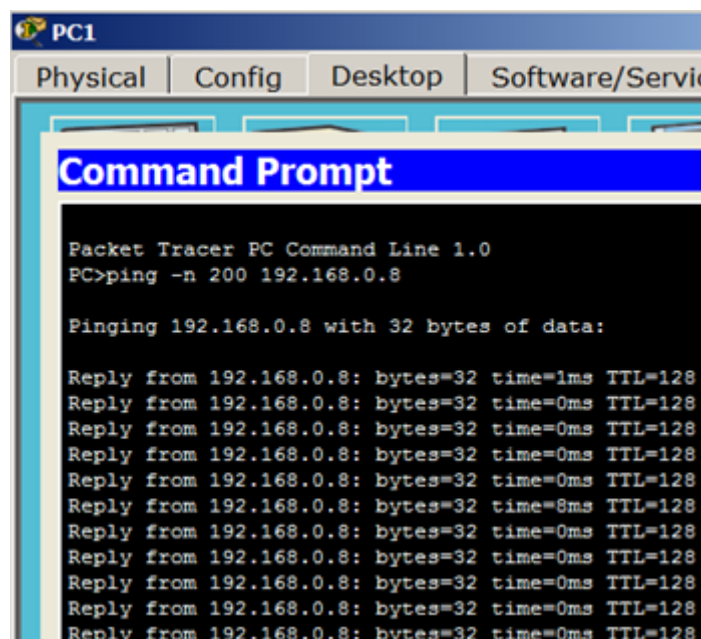


Рис. 7.6. Передача 200 пакетов на PC8

Одновременно с пингом, следует нагрузить сеть, включив генератор трафика на компьютере PC2 (узел назначения – PC8, размер поля данных–2500 байт, период повторения передачи - 0,1 с, – рис. 7.7).



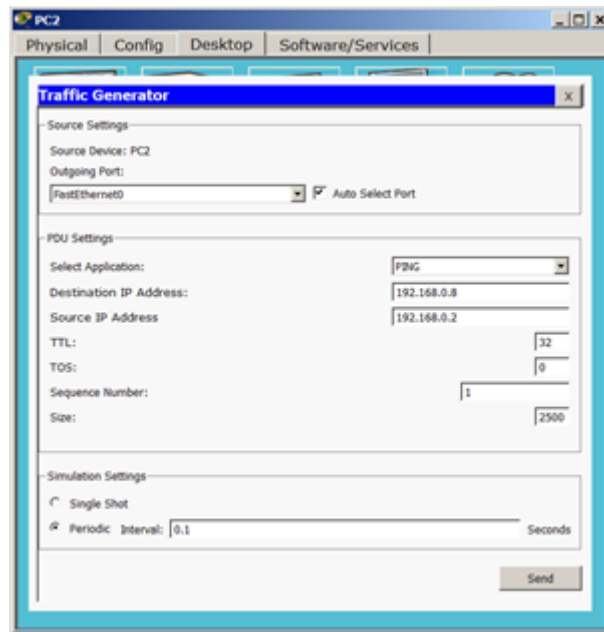


Рис. 7.7. Увеличение нагрузки на сеть

Для оценки качества работы сети следует зафиксировать число потерянных пакетов (рис. 7.8).

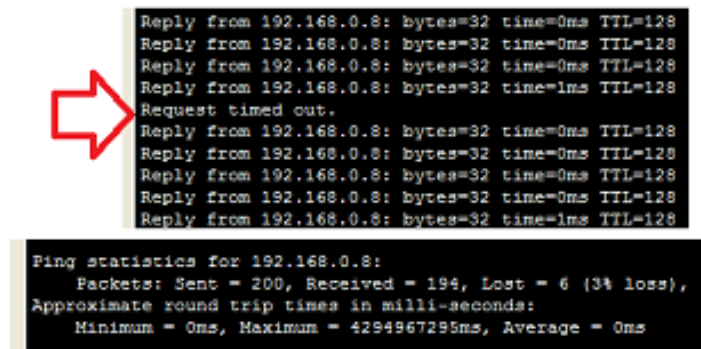


Рис. 7.8. Потеряно 6 пакетов

Как вариант, можно было бы загрузить сеть путем организации еще одного потока трафика между какими-либо узлами сети, например, включив генератор трафика еще на ноутбуке PC3.

В заключение этой части работы следует остановить *Traffic Generator* на всех узлах, нажав кнопку **Stop**.



## Повышение пропускной способности локальной вычислительной сети

Следует проверить тот факт, что установка коммутаторов вместо хабов устраняет возможность возникновения коллизий между пакетами пользователей сети. Заменить центральный концентратор на коммутатор (рис. 7.9). Убедиться, что сеть находится в рабочем состоянии - все маркеры портов не красные, а зеленые.

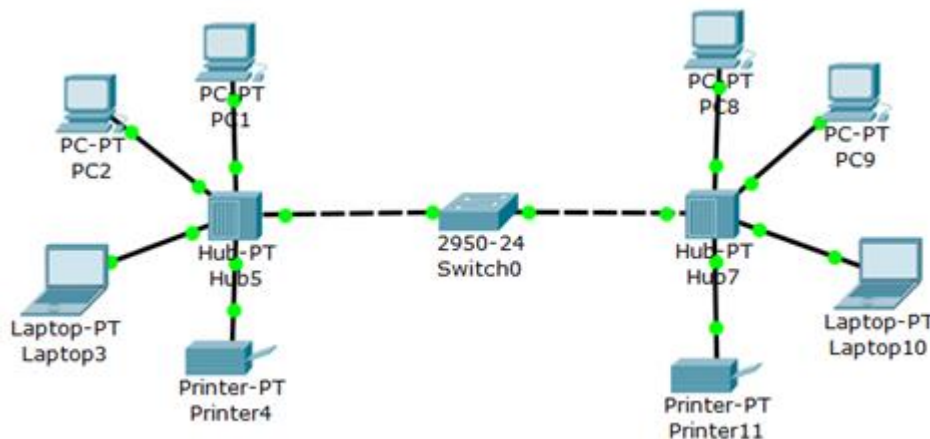


Рис. 7.9. Топология сети при замене центрального концентратора на коммутатор

Задать поток пакетов между PC1 и PC8 при помощи команды `ping -n 200 192.168.0.8` и включить *Traffic Generator* на PC2. Проследить работу нового варианта сети. Убедиться, что за счет снижения паразитного трафика качество работы сети стало выше (рис. 7.10).

```

Ping statistics for 192.168.0.8:
    Packets: Sent = 200, Received = 199, Lost = 1 (1% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 4294967295ms, Average = 0ms
  
```

```

Ping statistics for 192.168.0.8:
    Packets: Sent = 200, Received = 199, Lost = 1 (1% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 4294967295ms, Average = 0ms
  
```

Рис. 7.10. Потерян 1 пакет

### Задание 7.3. Улучшение качества передачи

Проверить самостоятельно, что замена не одного, а всех хабов коммутаторами существенно улучшит качество передачи трафика в сети.

#### Контрольные вопросы

1. Чем отличается концентратор от коммутатора?
2. Каким образом можно задать фильтрацию пакетов?
3. Какие настройки можно задать в приложении Traffic Generator?
4. Для чего предназначен параметр TTL?
5. Как можно улучшить качество передачи трафика в сети?

## 8. ПРОТОКОЛЫ МАРШРУТИЗАЦИИ

### Практическая работа 8-1. Настройка протокола RIP версии 2

**Задание:** настроить маршрутизацию на схеме, представленной на рис. 8.1.

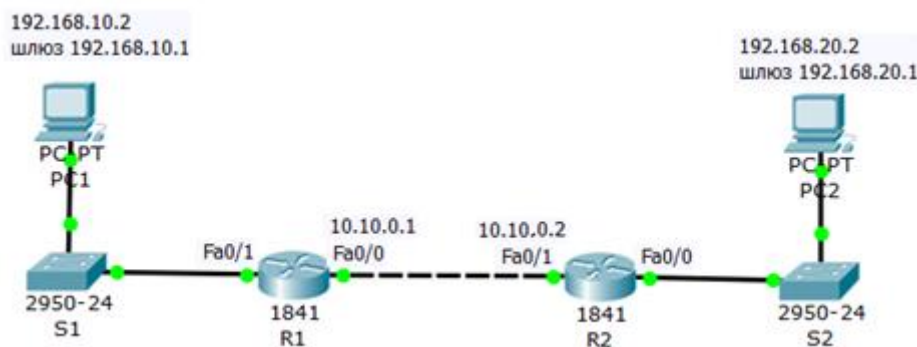


Рис. 8.1. Схема сети

#### *Настройка протокола RIP на маршрутизаторе R1*

Войти в конфигурации в консоль маршрутизатора и выполнить следующие настройки (рис. 8.2).

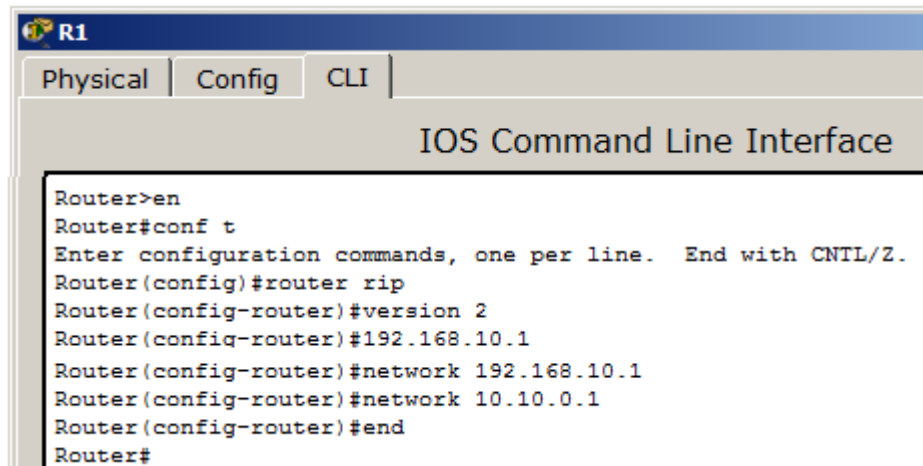


Рис. 8.2. Настройка протокола RIPv2 на маршрутизаторе Router1

**Router(config)#router rip** (Вход в режим конфигурирования протокола RIP).

**Router(config-router)#network 192.168.10.1** (Подключение клиентской сети к роутеру со стороны коммутатора S1).

**Router(config-router)#network 192.168.20.1** (Подключение второй сети, то есть сети между роутерами).

**Router(config-router)#version 2** (Задание использования второй версии протокол RIP).

### *Настройка протокола RIP на маршрутизаторе R2*

Войти в конфигурации роутера 2 и выполните следующие настройки (рис. 8.3).

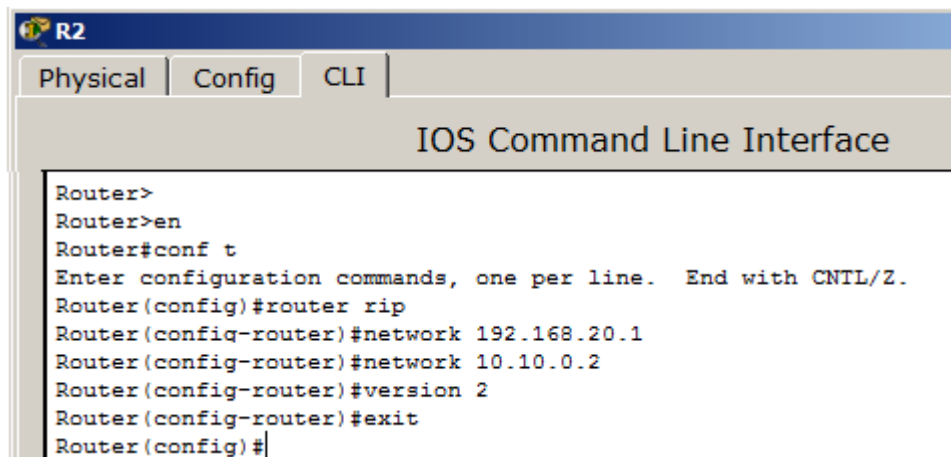


Рис. 8.3. Настройка протокола RIPv2 на маршрутизаторе R2

## Проверка настройки коммутаторов и протокола RIP

Посмотреть настройки протокола RIPv2 на маршрутизаторах R1 и R2 (рис. 8.4).

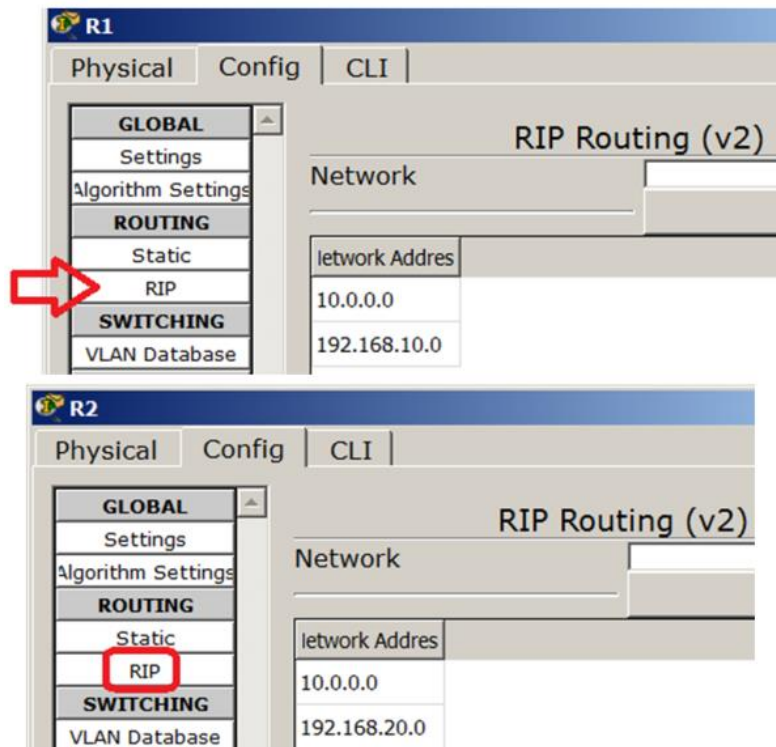


Рис. 8.4. Настройки маршрутизаторов R1 и R2

Чтобы убедиться в том, что маршрутизаторы действительно правильно сконфигурированы и работают корректно, посмотреть таблицу RIP роутеров, используя команду: **Router#show ip route rip** (рис. 8.5 и рис. 8.6).



Рис. 8.5. Таблица маршрутизации R1

Данная таблица показывает, что к сети 192.168.10.0 есть только один маршрут: через R1(сеть 10.10.0.1).

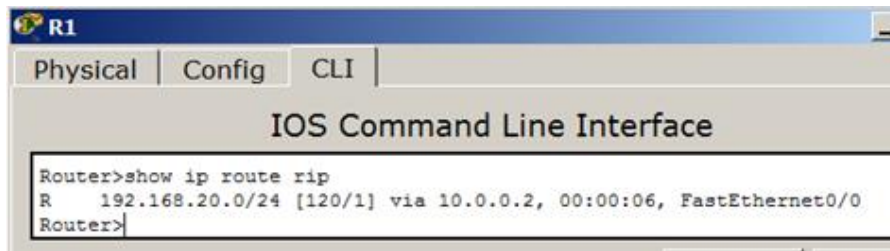


Рис. 8.6. Таблицы маршрутизации R2

Данная таблица показывает, что к сети 192.168.20.0 есть только один маршрут: через R2 (сеть 10.10.0.2).

### *Проверка связи между PC1 и PC2*

Проверить, что маршрутизация производится верно (рис. 8.7).

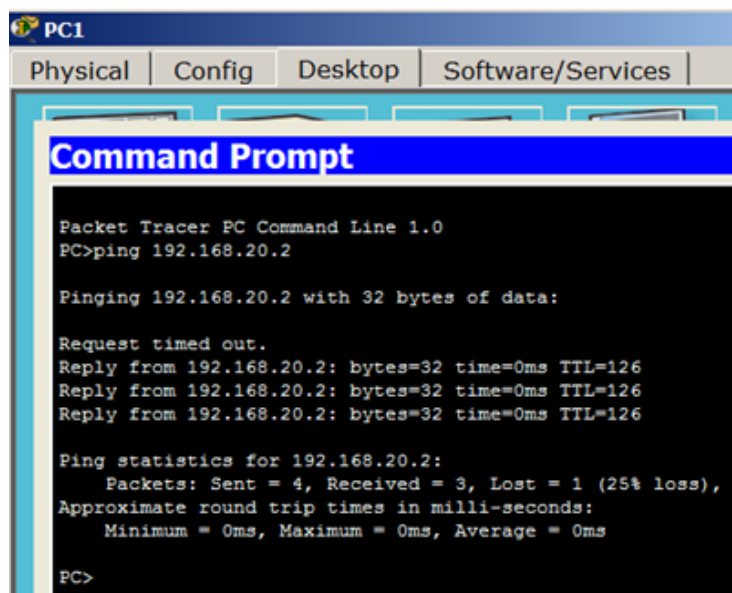


Рис. 8.7. Пинг с PC1 на PC2

## Практическая работа 8-2. Конфигурирование протокола RIP версии 2

На рис. 8.8 представлена сеть, на примере которой необходимо сконфигурировать протокол маршрутизации RIP v2.

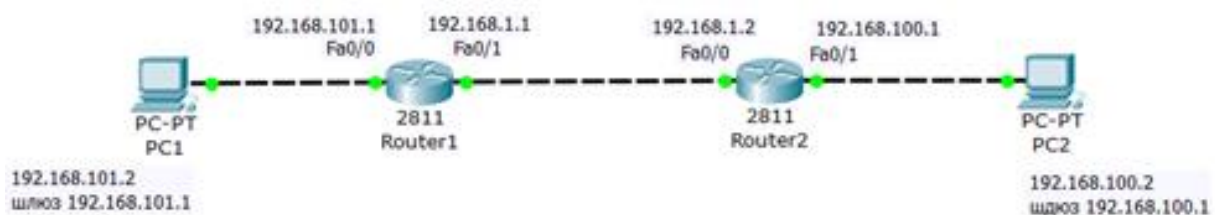


Рис. 8.8. Сеть для конфигурации протоколов маршрутизации

Выполнить конфигурацию R1 (рис. 8.9).

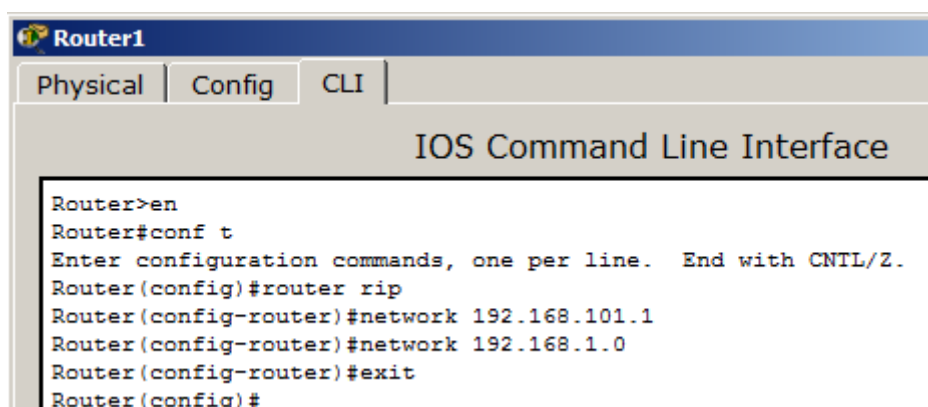


Рис. 8.9. Настройка RIP на R1

Просмотр результата на вкладке **Config** (рис. 8.10).

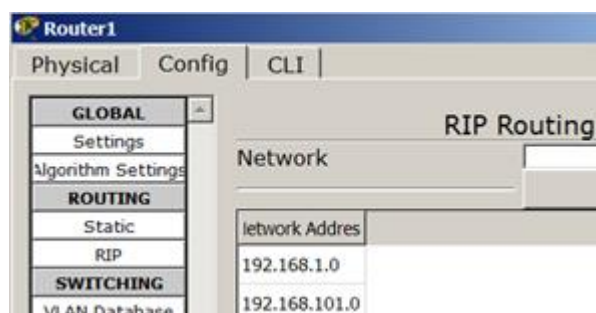


Рис. 8.10. Окно R1, вкладка Config

Конфигурация R2 (рис. 8.11).

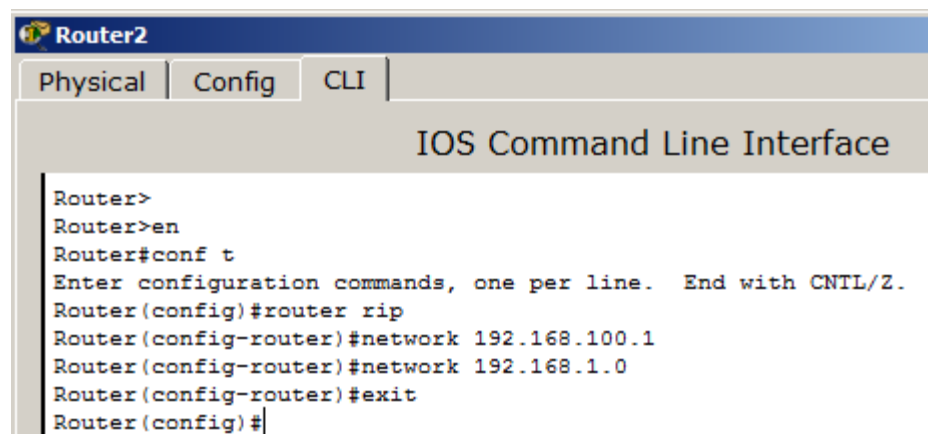


Рис. 8.11. Настройка RIP на R2

Просмотр результата (рис. 8.12).

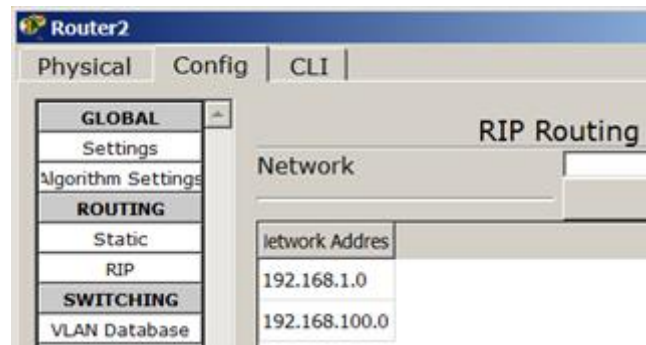


Рис. 8.12. Окно R2, вкладка Config

Проверка доступности ПК из разных сетей (рис. 8.13).

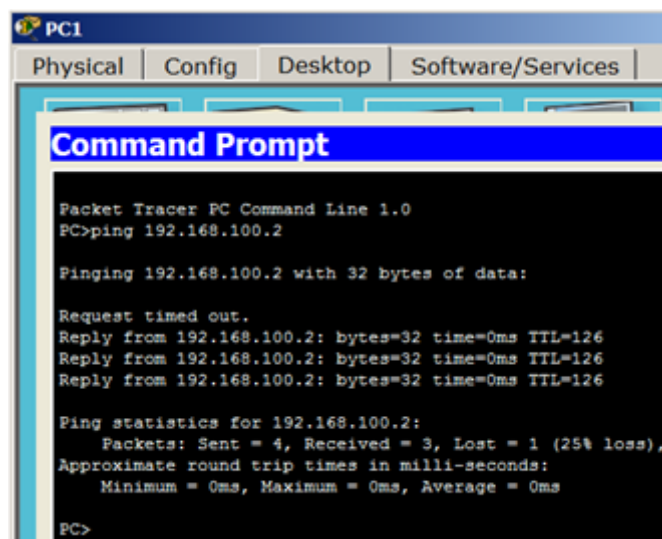


Рис. 8.13. Результат маршрутизации по протоколу RIP

## Протокол маршрутизации EIGRP

Протокол EIGRP более прост в реализации и менее требователен к вычислительным ресурсам маршрутизатора, чем протокол OSPF. Также EIGRP имеет более продвинутый алгоритм вычисления метрики. В формуле вычисления метрики есть возможность учитывать загруженность и надежность интерфейсов на пути пакета. Недостатком протокола EIGRP является его ограниченность в его использовании только на оборудовании компании Cisco.

## Практическая работа 8-3. Конфигурирование протокола EIGRP

Схема сети изображена на рис. 8.14.



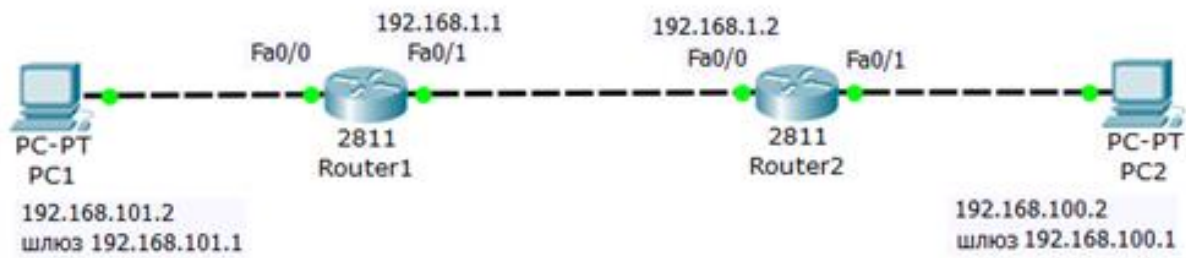


Рис. 8.14. Схема для конфигурации протокола EIGRP

### *Программирование R1*

Конфигурация R1 (рис. 8.15).

```

Router1
Physical Config CLI
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router eigrp 10
Router(config-router)#network 192.168.101.1
Router(config-router)#exit
Router(config)#
  
```

Рис. 8.15. Конфигурирование R1

### *Программирование R2*

Конфигурация R2 (рис. 8.16).

```

Router2
Physical Config CLI
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router eigrp 10
Router(config-router)#network 192.168.100.1
Router(config-router)#network 192.168.1.0
Router(config-router)#exit
Router(config)#
  
```

Рис. 8.16. Конфигурирование R2

### *Проверка работы сети*

Проверка работы маршрутизаторов (рис. 8.17).



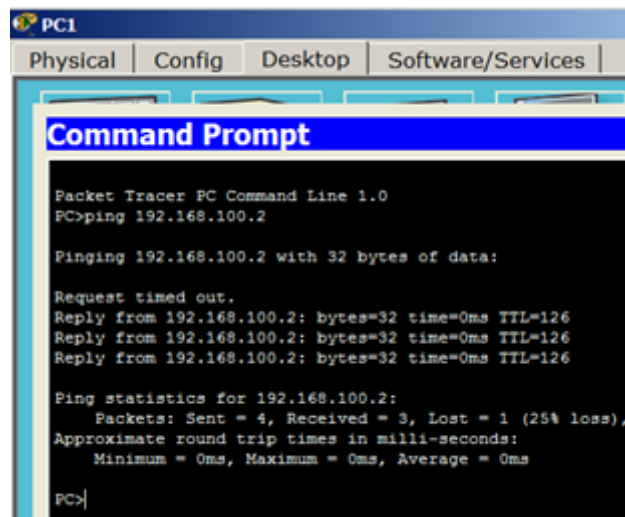


Рис. 8.17. Результат проверки работоспособности сети

## Практическая работа 8-4. Пример конфигурирования протокола OSPF

Собрать схему, изображенную на рис. 8.18.

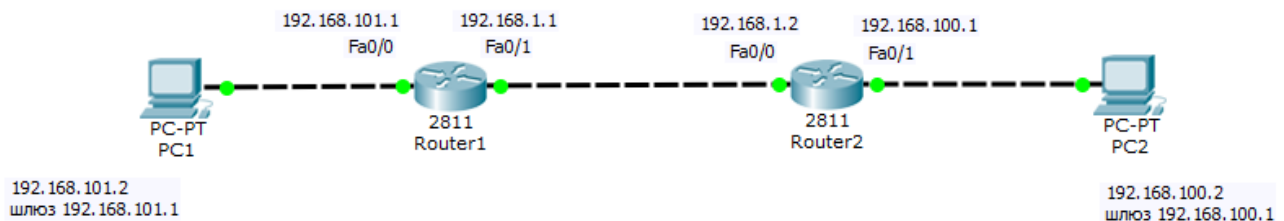


Рис. 8.18. Схема для конфигурации протокола OSPF

### Настройка роутеров

Выполнить конфигурирование R1 (рис. 8.19).

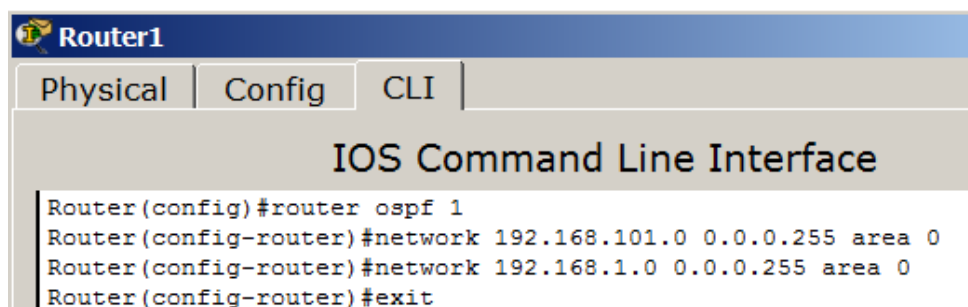


Рис. 8.19. Настройка R1

Выполнить настройки R2 (рис. 8.20).

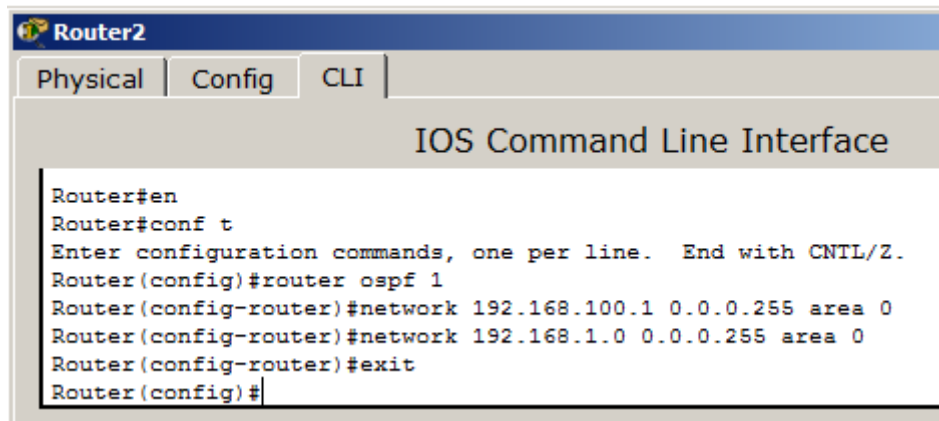


Рис. 8.20. Настройка R2

### Проверка результата

Для проверки маршрутизации пропинговать ПК из разных сетей (рис. 8.21).

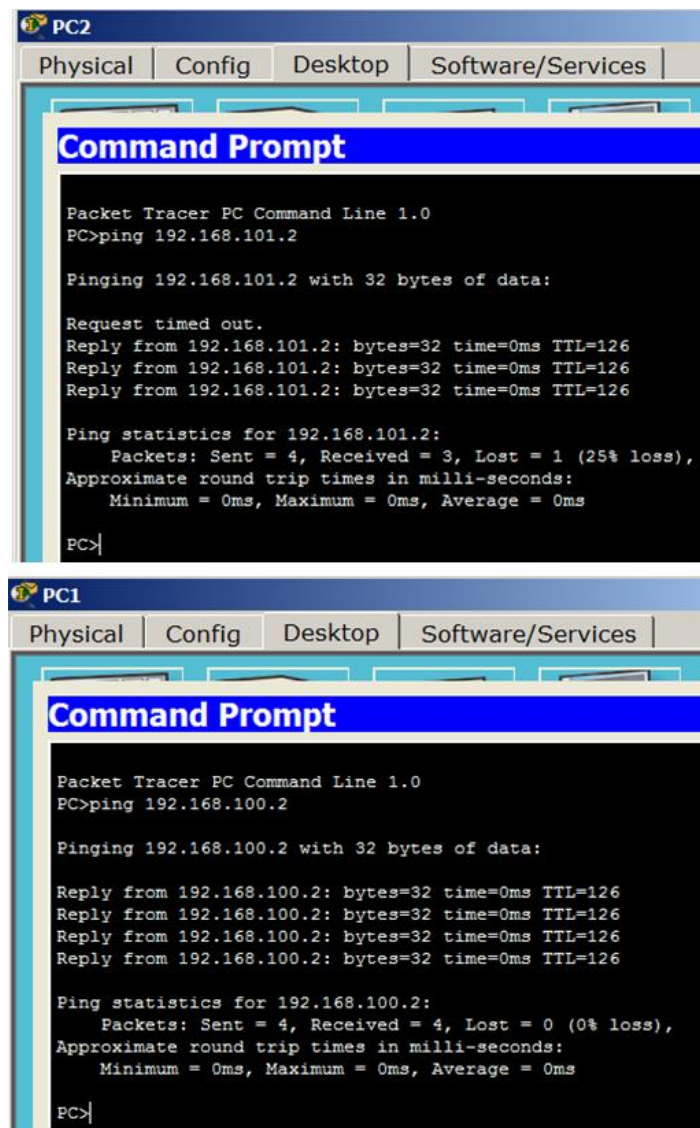


Рис. 8.21. Результат проверки работоспособности OSPF

## Практическая работа 8-5. Настройка маршрутизации по протоколу OSPF

Построить следующую схему (рис. 8.22).

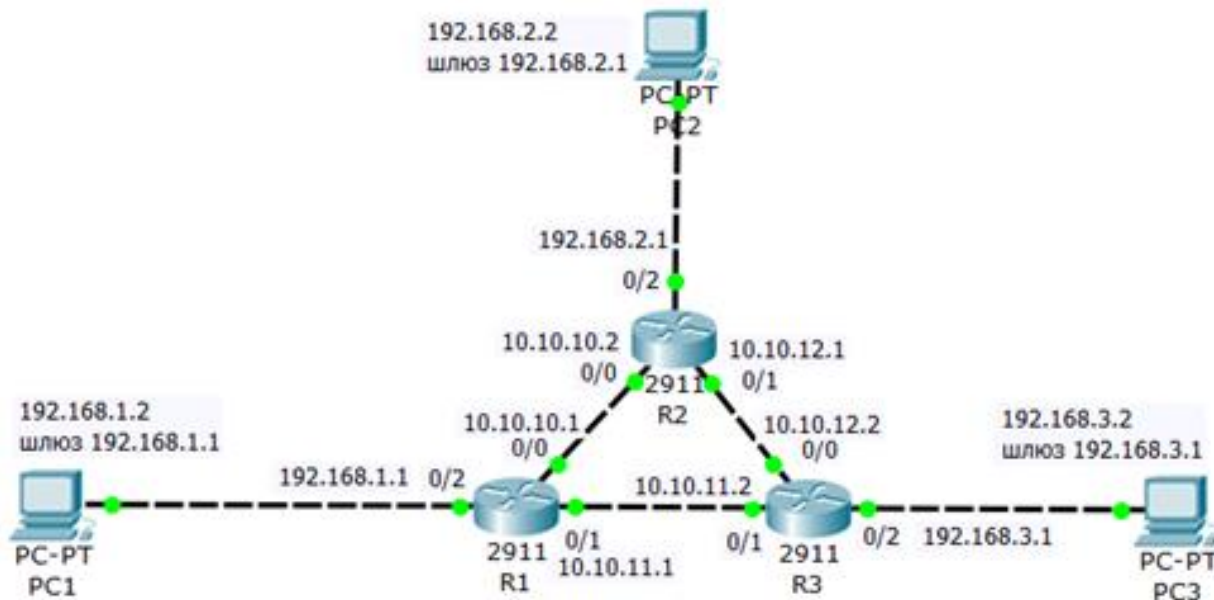


Рис. 8.22. Начальная схема сети для нашей работы

### *Настроить loopback интерфейс на R1*

На R1 настроить программный **loopback** интерфейс — алгоритм, который направляет полученный сигнал (или данные) обратно отправителю (рис. 8.23).

IPv4-адрес, назначенный loopback-интерфейсу, может быть необходим для процессов маршрутизатора, в которых используется IPv4-адрес интерфейса в целях идентификации. Один из таких процессов — алгоритм кратчайшего пути (OSPF). При включении интерфейса loopback для идентификации маршрутизатор будет использовать всегда доступный адрес интерфейса loopback, а не IP-адрес, назначенный физическому порту, работа которого может быть нарушена. На маршрутизаторе можно активировать несколько интерфейсов loopback. IPv4-адрес для каждого интерфейса loopback должен быть уникальным и не должен быть задействован другим интерфейсом.



Рис. 8.23. Настройка интерфейса loopback на R1

### *Настроить протокол OSPF на R1*

Включить OSPF на R1, все маршрутизаторы должны быть в одной зоне **area 0** (рис. 8.24).



Рис. 8.24. Включаем протокол OSPF на R1

Проверка результата настроек (рис. 8.25).

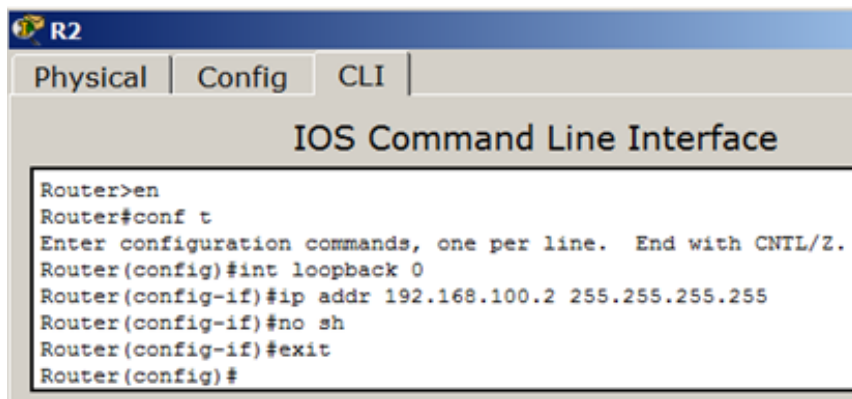
Port	Link	VLAN	IP Address
GigabitEthernet0/0	Up	--	10.10.10.1/30
GigabitEthernet0/1	Up	--	10.10.11.1/30
GigabitEthernet0/2	Up	--	192.168.1.1/24
Loopback0	Up	--	192.168.100.1/32

Рис. 8.25. Маршрутизатор R1 настроен

Следует обратить внимание, что физически порта 192.168.100.1 нет, он существует только логически (программно).

### *Настроить loopback интерфейс на R2*

На R2 настроить программный loopback интерфейс по аналогии с R1 (рис. 8.26).



```

R2
Physical | Config | CLI |
IOS Command Line Interface

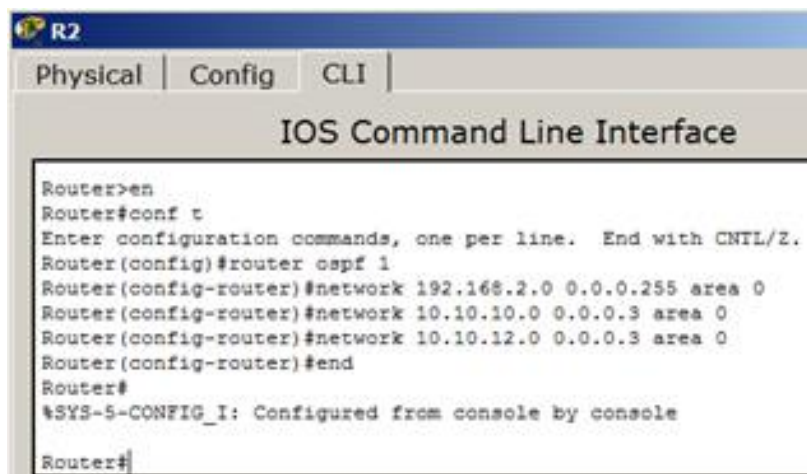
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int loopback 0
Router(config-if)#ip addr 192.168.100.2 255.255.255.255
Router(config-if)#no sh
Router(config-if)#exit
Router(config)#

```

Рис. 8.26. Настройка логического интерфейса loopback на R2

### *Настроить OSPF на R2*

Включить протокол OSPF на R2, все маршрутизаторы должны быть в одной зоне area 0 (рис. 8.27).



```

R2
Physical | Config | CLI |
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.2.0 0.0.0.255 area 0
Router(config-router)#network 10.10.10.0 0.0.0.3 area 0
Router(config-router)#network 10.10.12.0 0.0.0.3 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#

```

Рис. 8.27. Включение протокола OSPF на R2

Проверить результат настроек (рис. 8.28).

Port	Link	VLAN	IP Address
GigabitEthernet0/0	Up	--	10.10.10.2/30
GigabitEthernet0/1	Up	--	10.10.12.1/30
GigabitEthernet0/2	Up	--	192.168.2.1/24
Loopback0	Up	--	192.168.100.2/32

Рис. 8.28. Маршрутизатор R2 настроен

### *Настроить loopback интерфейс на R3*

Выполнить все аналогично предыдущим действиям (рис. 8.29 – 8.31).



Рис. 8.29. Настройка логического интерфейса loopback на R3

### *Настроить протокол OSPF на R3*

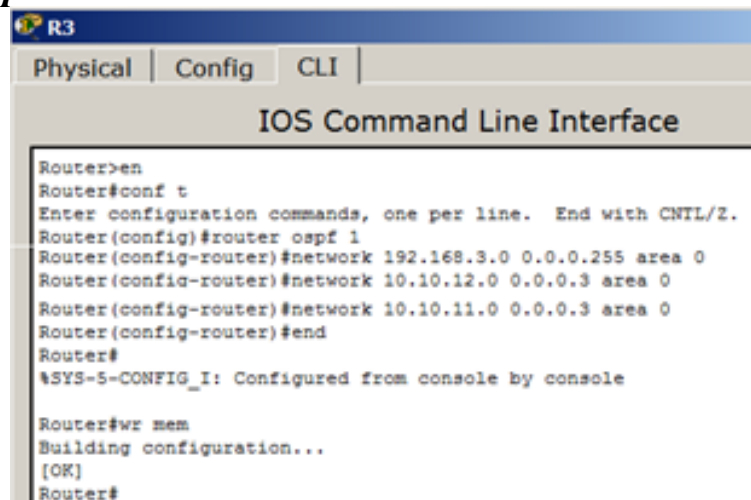


Рис. 8.30. Включение протокола OSPF на R2

Port	Link	VLAN	IP Address
GigabitEthernet0/0	Up	--	10.10.12.2/30
GigabitEthernet0/1	Up	--	10.10.11.2/30
GigabitEthernet0/2	Up	--	192.168.3.1/24
Loopback0	Up	--	192.168.100.3/32

Рис. 8.31. Маршрутизатор R3 настроен

### *Проверить работу сети*

Убедиться, что роутер R3 видит R2 и R1 (рис. 8.32).





Рис. 8.32. Роутер R3 видит своих соседей

Просмотреть таблицу маршрутизации для R3 (рис. 8.33).

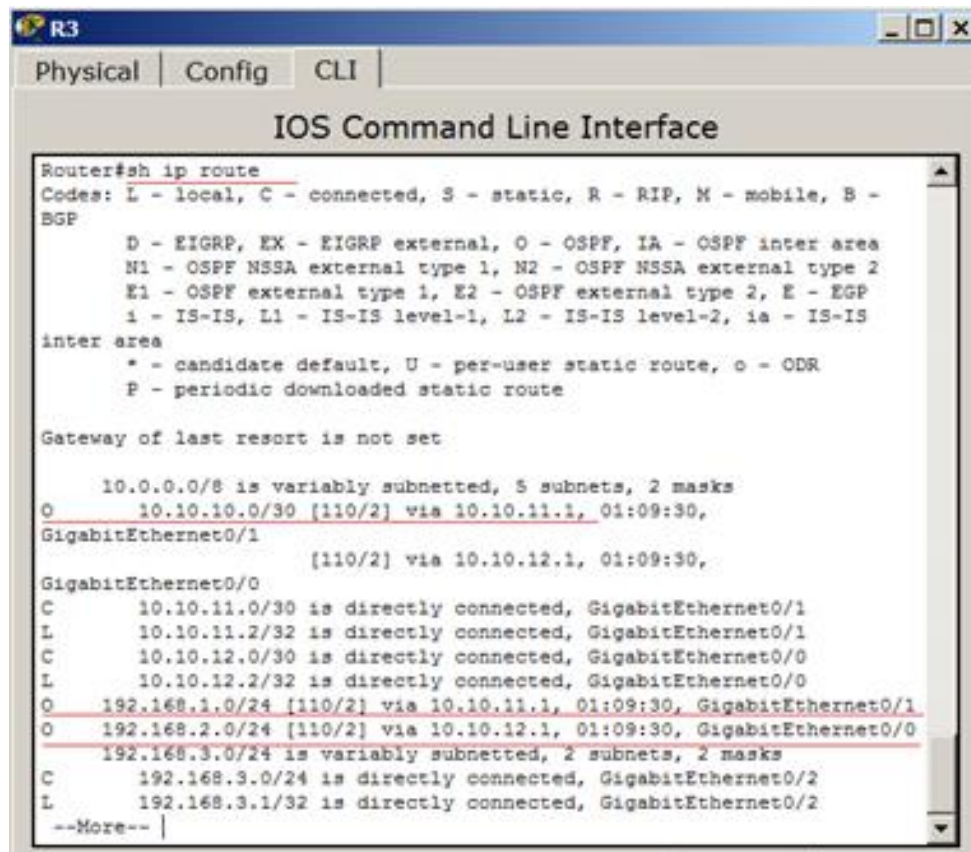
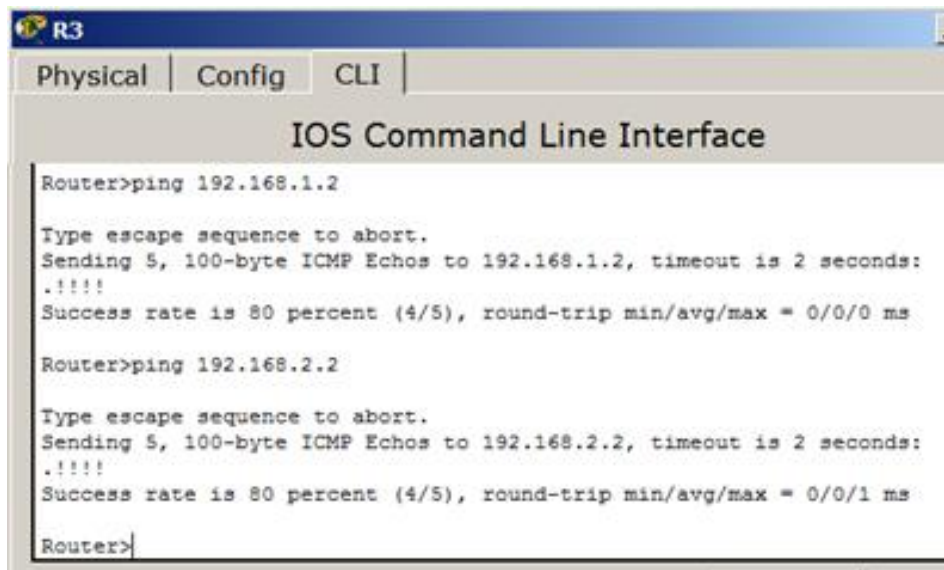


Рис. 8.33. Таблица маршрутизации для R3

В этой таблице запись с буквой "O" говорит о том, что данный маршрут прописан протоколом OSPF. Сеть 192.168.1.0 доступна для R3 через адрес 10.10.11.1 (это порт gig0/1 маршрутизатора R1). Аналогично, сеть 192.168.2.0 доступна для R3 через адрес 10.10.12.1 (это порт gig0/1 маршрутизатора R2).

Проверить доступность разных сетей (рис. 8.34).



```

R3
Physical | Config | CLI |
IOS Command Line Interface

Router>ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/0 ms

Router>ping 192.168.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/1 ms

Router>
  
```

Рис. 8.34. Сети 192.168.1.0 и 192.168.2.0 доступны

### Контрольные вопросы

1. Как выполнить настройку протокола RIP?
2. Какая команда служит для просмотра таблицы маршрутизации?
3. Дать краткую характеристику протокола EIGRP.
4. Каким образом выполнить конфигурирование протокола EIGRP?
5. Каким образом выполнить конфигурирование протокола OSPF?
6. Для чего предназначен интерфейс loopback?

## 9. КОНФИГУРИРОВАНИЕ QoS

### 9.1. FIFO

Элементарная очередь с последовательным прохождением пакетов, работающая по принципу первый пришел – первый ушел (First In First Out - FIFO). По сути, здесь нет никакой приоритезации. Включается по умолчанию на интерфейсах со скоростью больше 2 Мбит/с.



## 9.2. PQ. Очереди приоритетов

Priority Queuing (PQ) обеспечивает безусловный приоритет одних пакетов над другими. Всего 4 очереди: high, medium, normal и low. Обработка ведется последовательно (от high до low), начинается с высокоприоритетной очереди и до ее полной очистки не переходит к менее приоритетным очередям. Таким образом, возможна монополизация канала высокоприоритетными очередями. Трафик, приоритет которого явно не указан, попадет в очередь по умолчанию (default).

### Параметры команды.

распределение протоколов по очередям:

```
priority-list LIST_NUMBER protocol PROTOCOL {high | medium | normal | low}
list ACCESS_LIST_NUMBER
```

определение очереди по умолчанию:

```
priority-list LIST_NUMBER default {high | medium | normal | low}
```

определение размеров очередей (в пакетах):

```
priority-list LIST_NUMBER queue-limit HIGH_QUEUE_SIZE
MEDIUM_QUEUE_SIZE NORMAL_QUEUE_SIZE LOW_QUEUE_SIZE
```

обозначения:

LIST\_NUMBER - номер обработчика PQ (листа)

PROTOCOL - протокол

ACCESS\_LIST\_NUMBER - номер аксесс листа

HIGH\_QUEUE\_SIZE - размер очереди HIGH

MEDIUM\_QUEUE\_SIZE - размер очереди MEDIUM

NORMAL\_QUEUE\_SIZE - размер очереди NORMAL

LOW\_QUEUE\_SIZE - размер очереди LOW

### Алгоритм настройки.

1. Определить 4 очереди

```
access-list 110 permit ip any any precedence network
```

```
access-list 120 permit ip any any precedence critical
```

```
access-list 130 permit ip any any precedence internet
```

```
access-list 140 permit ip any any precedence routine
```

```
priority-list 1 protocol ip high list 110
```

```
priority-list 1 protocol ip medium list 120
```

```
priority-list 1 protocol ip normal list 130
```

```
priority-list 1 protocol ip low list 140
```

```
priority-list 1 default low
```

Дополнительно можно установить размеры очередей в пакетах  
 priority-list 1 queue-limit 30 60 90 120

## 2. Привязать к интерфейсу

!

```
interface FastEthernet0/0
ip address 192.168.0.2 255.255.255.0
speed 100
full-duplex
priority-group 1
no cdp enable
!
```

## 3. Просмотр результата

# sh queueing priority

Current priority queue configuration:

List	Queue Args	-	-
1	low	default	-
1	high	protocol ip	list 110
1	medium	protocol ip	list 120
1	normal	protocol ip	list 130
1	low	protocol ip	list 140

#sh interfaces fastEthernet 0/0

...

Queueing strategy: priority-list 1

...

#sh queueing interface fastEthernet 0/0

Interface FastEthernet0/0 queueing strategy: priority

Output queue utilization (queue/count)

high/19 medium/0 normal/363 low/0

## 9.3. CQ. Произвольные очереди

Custom Queuing (CQ) обеспечивает настраиваемые очереди. Предусматривается управление долей полосы пропускания канала для каждой очереди. Поддерживается 17 очередей. Системная 0 очередь зарезервирована для управляющих высокоприоритетных пакетов (маршрутизация и т.п.) и пользователю недоступна.

Очереди обходятся последовательно, начиная с первой. Каждая очередь содержит счетчик байт, который в начале обхода содержит заданное значение и уменьшается на размер пакета, пропущенного из этой очереди. Если счетчик не ноль, то пропускается следующий пакет целиком, а не его фрагмент, равный остатку счетчика.

### **Параметры команды.**

определение полосы пропускания очередей:

```
queue-list LIST-NUMBER queue QUEUE_NUMBER byte-count
BYTE_COUT
```

определение размеров очередей:

```
queue-list LIST-NUMBER queue QUEUE_NUMBER limit QUEUE_SIZE
```

обозначения:

LIST-NUMBER - номер обработчика

QUEUE\_NUMBER - номер очереди

BYTE\_COUT - размер очереди в пакетах

### **Алгоритм настройки.**

1. Определить очереди

```
access-list 110 permit ip host 192.168.0.100 any
```

```
access-list 120 permit ip host 192.168.0.200 any
```

```
queue-list 1 protocol ip 1 list 110
```

```
queue-list 1 protocol ip 2 list 120
```

```
queue-list 1 default 3
```

```
queue-list 1 queue 1 byte-count 3000
```

```
queue-list 1 queue 2 byte-count 1500
```

```
queue-list 1 queue 3 byte-count 1000
```

Дополнительно можно установить размеры очередей в пакетах

```
queue-list 1 queue 1 limit 50
```

```
queue-list 1 queue 2 limit 50
```

```
queue-list 1 queue 3 limit 50
```

2. Привязать к интерфейсу

!

```
interface FastEthernet0/0
```

```
ip address 192.168.0.2 255.255.255.0
```

```
speed 100
```

```
full-duplex
```

```
custom-queue-list 1
```

no cdp enable  
!

### 3. Просмотр результата

#sh queueing custom

Current custom queue configuration:

List	Queue	Args	-
1	3	default	-
1	1	protocol ip	list 110
1	2	protocol ip	list 120
1	1	byte-count 1000	-
1	2	byte-count 1000	-
1	3	byte-count 2000	-

#sh interface FastEthernet0/0

...

Queueing strategy: custom-list 1

...

#sh queueing interface fastEthernet 0/0

Interface FastEthernet0/0 queueing strategy: custom

Output queue utilization (queue/count)

0/90 1/0 2/364 3/0 4/0 5/0 6/0 7/0 8/0

9/0 10/0 11/0 12/0 13/0 14/0 15/0 16/0

## 9.4. WFQ. Взвешенные справедливые очереди

Weighted Fair Queuing (WFQ) автоматически разбивает трафик на потоки (flows). По умолчанию их число равно 256, но может быть изменено (параметр dynamic-queues в команде fair-queue). Если потоков больше, чем очередей, то в одну очередь помещается несколько потоков. Принадлежность пакета к потоку (классификация) определяется на основе TOS, протокола, IP адреса источника, IP адреса назначения, порта источника и порта назначения. Каждый поток использует отдельную очередь.

Обработчик WFQ (scheduler) обеспечивает равномерное (fair - честное) разделение полосы между существующими потоками. Для этого доступная полоса делится на число потоков и каждый получает равную часть. Кроме того, каждый поток получает свой вес (weight), с

некоторым коэффициентом, обратно пропорциональный IP приоритету (TOS). Вес потока также учитывается обработчиком.

В итоге WFQ автоматически справедливо распределяет доступную пропускную способность, дополнительно учитывая TOS. Потоки с одинаковыми IP приоритетами TOS получают равные доли полосы пропускания; потоки с большим IP приоритетом – большую долю полосы. В случае перегрузок ненагруженные высокоприоритетные потоки функционируют без изменений, а низкоприоритетные высоконагруженные – ограничиваются.

Вместе с WFQ работает RSVP. По умолчанию WFQ включается на низкоскоростных интерфейсах.

### **Алгоритм настройки.**

1. Пометить трафик каким-либо способом (установить IP приоритет - TOS) или получить его помеченным

2. Включить WFQ на интерфейсе  
`interface FastEthernet0/0`  
`fair-queue`

`interface FastEthernet0/0`  
`fair-queue CONGESTIVE_DISCARD_THRESHOLD DYNAMIC_QUEUES`

### **Параметры:**

`CONGESTIVE_DISCARD_THRESHOLD` - число пакетов в каждой очереди, при превышении которого пакеты игнорируются (по умолчанию - 64)

`DYNAMIC_QUEUES` - число подочереди, по которым классифицируется трафик (по умолчанию - 256)

3. Просмотр результата

`# sh queueing fair`

`# sh queueing interface FastEthernet0/0`

## **9.5. CBWFQ**

Class Based Weighted Fair Queuing (CBWFQ) соответствует механизму обслуживания очередей на основе классов. Весь трафик разбивается на 64 класса на основании следующих параметров: входной интерфейс, доступный лист (access list), протокол, значение DSCP, метка MPLS QoS.

Общая пропускная способность выходного интерфейса распределяется по классам. Выделяемую каждому классу полосу пропускания можно определять как в абсолютное значение (bandwidth в kbit/s) или в процентах (bandwidth percent) относительно установленного значения на интерфейсе.

Пакеты, не попадающие в сконфигурированные классы, попадают в класс по умолчанию, который можно дополнительно настроить и который получает оставшуюся свободной полосу пропускания канала. При переполнении очереди любого класса пакеты данного класса игнорируются. Алгоритм отклонения пакетов внутри каждого класса можно выбирать: включенное по умолчанию обычное отбрасывание (tail-drop, параметр queue-limit) или WRED (параметр random-detect). Только для класса по умолчанию можно включить равномерное ( честное ) деление полосы (параметр fair-queue).

CBWFQ поддерживает взаимодействие с RSVP.

### **Параметры команды.**

критерии отбора пакетов классом:

```
class-map match-all CLASS
match access-group
match input-interface
match protocol
match ip dscp
match ip rtp
match mpls experimental
```

определение класса:

```
class CLASS
bandwidth BANDWIDTH
bandwidth percent BANDWIDTH_PERCENT
queue-limit QUEUE-LIMIT
random-detect
```

определение класса по умолчанию (default):

```
class class-default
bandwidth BANDWIDTH
bandwidth percent BANDWIDTH_PERCENT
queue-limit QUEUE-LIMIT
```

random-detect  
fair-queue

### **обозначения:**

CLASS - название класса.

BANDWIDTH - минимальная полоса kbit/s, значение независимо от bandwidth на интерфейсе.

BANDWIDTH\_PERCENT - процентное соотношение от bandwidth на интерфейсе.

QUEUE-LIMIT - максимальное количество пакетов в очереди.

random-detect - использование WRED.

fair-queue - равномерное деление полосы, только для класса по умолчанию

По умолчанию абсолютное значение Bandwidth в классе CBWFQ не может превышать 75% значение Bandwidth на интерфейсе. Это можно изменить командой max-reserved-bandwidth на интерфейсе.

### **Алгоритм настройки.**

1. Распределение пакетов по классам - class-map  
access-list 101 permit ip any any precedence critical

```
class-map match-all Class1
match access-group 101
```

2. Описание правил для каждого класса - policy-map

```
policy-map Policy1
class Class1
bandwidth 100
queue-limit 20
class class-default
bandwidth 50
random-detect
```

3. Запуск заданной политики на интерфейсе - service-policy

```
interface FastEthernet0/0
bandwidth 256
service-policy output Policy1
```

4. Просмотр результата

```
#sh class Class1
#sh policy Policy1
#sh policy interface FastEthernet0/0
```

**Пример 1.**

Деление общей полосы по классам в процентном соотношении (40, 30, 20).

```
access-list 101 permit ip host 192.168.0.10 any
access-list 102 permit ip host 192.168.0.20 any
access-list 103 permit ip host 192.168.0.30 any
```

```
class-map match-all Platinum
match access-group 101
class-map match-all Gold
match access-group 102
class-map match-all Silver
match access-group 103
```

```
policy-map lsp
class Platinum
bandwidth percent 40
class Gold
bandwidth percent 30
class Silver
bandwidth percent 20
```

```
interface FastEthernet0/0
bandwidth 256
service-policy output lsp
```

**9.6. LLQ**

Low Latency Queuing (LLQ) – очередность с низкой задержкой. LLQ можно рассматривать как механизм CBWFQ с приоритетной очередью PQ (LLQ = PQ + CBWFQ).

PQ в LLQ позволяет обеспечить обслуживание чувствительного к задержке трафика. LLQ рекомендуется в случае наличия голосового (VoIP) трафика. Кроме того, он хорошо работает с видеоконференциями.

**Алгоритм настройки.**

1. Распределение пакетов по классам - Class-map

```
access-list 101 permit ip any any precedence critical
```

```
class-map match-all Voice
match ip precedence 6
```



```
class-map match-all Class1
match access-group 101
```

## 2. Описание правил для каждого класса - Policy-map

Аналогично CBWFQ, только для приоритетного класса (он один) указывается параметр priority.

```
policy-map Policy1
class Voice
priority 1000
class Class1
bandwidth 100
queue-limit 20
class class-default
bandwidth 50
random-detect
```

## 3. Запуск заданной политики на интерфейсе - Service-policy

```
interface FastEthernet0/0
bandwidth 256
service-policy output Policy1
```

### Пример 2.

Отнести класс Voice к PQ, а все остальное к CQWFQ.

```
!
class-map match-any Voice
match ip precedence 5
!
policy-map Voice
class Voice
priority 1000
class VPN
bandwidth percent 50
class class-default
fair-queue 16
!
interface X
Service-policy output Voice
!
```

### Пример 3.

Дополнительно ограничить общую скорость для PQ в LLQ, чтобы он не монополизировал весь канал в случае неправильной работы.

```
!
class-map match-any Voice
match ip precedence 5
!
policy-map Voice
class Voice
priority 1000
police 1024000 32000 32000 conform-action transmit exceed-action drop
class Vpn
bandwidth percent 50
class class-default
fair-queue 16
!
interface FastEthernet0/0
service-policy output Voice
!
```

### Варианты заданий для практической работы:

1. Настроить дисциплину обработки очередей Priority Queuing.
2. Настроить дисциплину обработки очередей Custom Queuing.
3. Настроить дисциплину обработки очередей Weighted Fair Queuing.
4. Настроить дисциплину обработки очередей CBWFQ.
5. Настроить дисциплину обработки очередей LLQ.

### Контрольные вопросы

1. Сколько очередей используется при дисциплине PQ?
2. Какой недостаток имеет дисциплина PQ?
3. Как выполняется привязка к интерфейсу?
4. Для чего используется системная очередь 0 в дисциплине CQ?
5. В чем преимущество дисциплины WFQ?
6. Какие особенности имеет дисциплина CBWFQ?
7. Когда рекомендуется использовать дисциплину LLQ?

### Библиографический список

1. Берлин А.Н. Основные протоколы Интернет/Интернет-Университет Информационных Технологий • 2008 год • 504 с/. Режим доступа: <http://www.knigafund.ru>
2. Ермаков А.Е. Основы конфигурирования корпоративных сетей Cisco: учеб. пособие/изд-во УМЦ ЖДТ (Маршрут) • 2013 год • 248 с. Режим доступа: <http://www.knigafund.ru>
3. Пятибратов А.П., Гудыно Л.П., Кириченко А.А. Вычислительные системы, сети и телекоммуникации/ Финансы и статистика • 2013 год • 736 с. Режим доступа: <http://www.knigafund.ru>
4. Лапониная О.Р. Протоколы безопасного сетевого взаимодействия/ Национальный Открытый Университет «ИНТУИТ» 2016.- 462 с. Режим доступа: <http://www.knigafund.ru>
5. Семенов Ю.А. Алгоритмы телекоммуникационных сетей/ Интернет-Университет Информационных Технологий • 2007 год • 512 с. Режим доступа: <http://www.knigafund.ru>
6. Воробьев С.П. Локальные вычислительные сети : учеб. пособие - Новочеркасск: Изд-во НВВКУС, 2006. - 165 с

*Учебно-методическое издание*

**Воробьёв Сергей Петрович**

## **КОМПЬЮТЕРНЫЕ СЕТИ**

*Учебно-методическое пособие для практических занятий*

Редактор *Я.В. Максименко*

Подписано в печать 11.04.2017.

Формат 60×84  $\frac{1}{16}$ . Бумага офсетная. Печать цифровая.  
Усл. печ.л. 4,88. Уч.-изд.л.5,0. Тираж 100 экз. Заказ 46-0693.

Южно-Российский государственный политехнический университет  
(НПИ) имени М.И. Платова  
Редакционно-издательский отдел ЮРГПУ(НПИ)  
346428, г. Новочеркасск, ул. Просвещения, 132

Отпечатано в ИД «Политехник»  
346428, г. Новочеркасск, ул. Первомайская, 166  
[mdp-npi@mail.ru](mailto:mdp-npi@mail.ru)