МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»
(СибГУТИ)
ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)
(ХИИК СибГУТИ)
СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ по дисциплине

ОП.13 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

для студентов специальности
09.02.07 «Информационные системы и программирование»

Хабаровск
2024

ББК 22
Р 189

Райлян М.Н., Дергунов Е.А. Методические указания по выполнению лабораторных работ по дисциплине ОП.13 «Информационная безопасность» для студентов специальности 09.02.07 «Информационные системы и программирование» среднего профессионального образования. – г. Хабаровск, ХИИК СибГУТИ; 2024год

Методические указания содержат задания для выполнения лабораторных работ по дисциплине Информационная безопасность, порядок выполнения работ, основные требования к знаниям и умениям студентов. Для студентов специальности 09.02.07 «Информационные системы и программирование» СПО.

Рецензенты:
Кузнецова М.В. – преподаватель высшей категории ХИИК СибГУТИ;
Щербаков А.Г. – преподаватель специальных дисциплин ХИИК СибГУТИ.

Рассмотрено на заседании ПЦК ИСП протокол № 8 от 06.03.2024 г.
Председатель ПЦК ИСП _____О.В. Диденко

Утверждено на заседании Методического совета
Протокол № 8 от 17 апреля 2024 года

г. Хабаровск, 2024 год

СОДЕРЖАНИЕ

Лабораторная работа №1 Программы шифрования и дешифрования по «шифру замены»	4
Лабораторная работа №2 Шифр перестановки и дешифрование шифра перестановки. Шифр замены по ГОСТ 28147-89	11
Лабораторная работа №3 Шифрование методом гаммирования	16
Лабораторная работа №4 Шифрование с имитовставкой	19
Лабораторная работа №5 Ассиметричное шифрование методом RSA	22
Лабораторная работа №6 Алгоритмы поведения вирусных и других вредоносных программ	27
Лабораторная работа №7 Алгоритмы предупреждения и обнаружения вирусных угроз.	34
Лабораторная работа № 8 Работа протокола STP на коммутаторах	43
Лабораторная работа № 9 Распределение сетей на сетевом уровне	55
Лабораторная работа № 10 Организация VPN-туннеля в сети	81

Лабораторная работа № 1

Тема: Программы шифрования и дешифрования по «шифру замены».

1. Цель: Закрепить теоретические знания и получить практические навыки по программированию алгоритма шифрования и дешифрования по методу «шифра замены».

Знать:

- понятие симметричного шифра, шифра замены;
- алгоритм «шифра замены»;
- алгоритм дешифрования «шифра замены».

Уметь:

- программировать на алгоритмических языках.

2. Подготовка к работе:

- повторить теоретический материал согласно лекциям.

3. Теоретический материал:

Метод замены - это один из классических методов шифрования, основанный на замене символов в открытом тексте другими символами согласно определенному правилу или ключу. Он включает в себя два основных шага: шифрование и дешифрование.

Существует несколько методов шифрования методом замены, включая:

1. Шифр Цезаря: Это один из самых простых методов шифрования, где каждая буква в открытом тексте заменяется на букву, находящуюся на определенном числе позиций в алфавите. Например, сдвиг на три позиции может превратить "А" в "Д", "Б" в "Е" и так далее.

Пример:

Открытый текст:	СЛУЧАЙ
Ключ:	5
Шифрованный текст:	ЦРШЬЕО

2. Шифр Виженера: Данный полиалфавитный шифр использует ключевое слово или фразу для создания нескольких шифрованных алфавитов. Затем каждая буква открытого текста шифруется с использованием

соответствующего алфавита согласно ключу. Этот метод делает анализ частоты букв более сложным, чем в случае с шифром Цезаря.

Таблица шифрования выглядит следующим образом:

	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
А	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
Б	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А
В	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б
Г	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В
Д	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г
Е	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д
Ж	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е
З	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж
И	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З
Й	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И
К	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й
Л	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К
М	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л
Н	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М
О	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н
П	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О
Р	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	С	Н	О	П
С	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р
Т	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
У	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
Ф	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Х	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ъ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Ы	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
Ь	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
Э	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
Ю	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
Я	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю

Рисунок 1.1 – Таблица Виженера

Пример шифрования:

Открытый текст:	ПЛАНЕТА
Ключ:	ШИФР
Шифротекст:	ЗУФЭЭЪФН

3. Аффинный шифр: Это комбинация шифра Цезаря с умножением на число и прибавлением другого числа. Каждая буква в открытом тексте сначала умножается на определенное число, затем прибавляется другое число, и наконец применяется модуль алфавита для получения шифротекста.

Предположим, у нас есть сообщение "Пример" для шифрования с ключами a=3 и b=7. Для русского алфавита m=33 (32 буквы и пробел).

Найдем числовое представление каждой буквы в сообщении (используя нумерацию от 0 до 32 для русских букв и пробела):

$П \rightarrow 15, р \rightarrow 15;$

$р \rightarrow 17, r \rightarrow 17;$

и так далее.

Применим формулу шифрования $E(x) = (3x + 7)\ mod\ 33$ для каждой буквы:

Для Р: $E(15) = (3 \cdot 15 + 7)\ mod\ 33 = 52\ mod\ 33 = 19$ Зашифрованная буква: С.

Для р: $E(15) = (3 \cdot 15 + 7)\ mod\ 33 = 52\ mod\ 33 = 19$ Зашифрованная буква: С.

Для р: $E(17) = (3 \cdot 17 + 7)\ mod\ 33 = 58\ mod\ 33 = 25$ Зашифрованная буква: Я.

Для и: $E(14) = (3 \cdot 14 + 7)\ mod\ 33 = 49\ mod\ 33 = 16$ Зашифрованная буква: П.

И так далее.

Таким образом, зашифрованное сообщение для "Пример" будет "СЯПЖОА".

4. Шифровка методом простой замены - это метод шифрования, при котором каждая буква или символ в открытом тексте заменяется на другую букву или символ согласно определенному правилу или ключу. Это один из наиболее простых исторических методов шифрования, который легко понять и реализовать

Все эти методы обеспечивают базовый уровень безопасности, однако современные методы шифрования обычно используют более сложные алгоритмы для защиты данных.

Шифровка методом простой замены выполняется следующим образом: Каждая буква открытого текста заменяется на один и тот же символ шифротекста по одному заданному правилу.

Алгоритмы для шифрования и дешифровки:

Шифрование:

1. Задать ключ замены, который определяет соответствие между символами открытого текста и символами шифртекста.

2. Разбить открытый текст на отдельные символы или блоки символов.

3. Для каждого символа в открытом тексте:

3.1 Найти его соответствие в ключе замены.

3.2 Заменить данный символ, на соответствующий символ из ключа замены.

4 Объединить полученные шифртексты в одну строку или последовательность символов.

Дешифрование:

1. Использовать тот же ключ замены, который использовался для шифрования.

2. Разбить шифртекст на отдельные символы или блоки символов.

3. Для каждого символа в шифртексте:

3.1 Найти его соответствие в обратном ключе замены.

3.2 Заменить данный символ на соответствующий символ из обратного ключа замены.

4. Объединить полученные символы в одну строку или последовательность символов, что представляет собой расшифрованный текст.

Пример простой замены:

Открытый алфавит:	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
	К	Л	М	Н	О	П	Р	С	Т	У	Ф
	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
Шифрованный алфавит:	В	Ы	А	У	Л	И	К	Й	О	Т	Н
	Е	М	Х	Р	Ч	П	Я	С	Ю	Д	Ь
	Ш	Б	Ж	З	Г	Ф	Ц	Э	Ё	Щ	Ъ
Открытый текст:	СЛУЧАЙ										
Шифротекст:	СМДЖВН										

Таким образом, каждая буква алфавита сдвигается на три позиции влево и используемое сообщение полностью шифруется по данному правилу. При дешифровке происходит обратный процесс: каждая буква шифрованного текста

заменяется на соответствующую букву в открытом тексте согласно тому же правилу.

4. Задание:

1. Разработать алгоритм и написать программу шифрования и дешифрования методом шифра Цезаря для русского текста в соответствии с вашим вариантом:

Вариант	**Ключ**	**Сообщение**
1	8	Следующая встреча в парке у фонтана
2	3	Под ключом - код к сейфу
3	12	Срочно: план эвакуации на случай X
4	9	Секрет: подарок у подушки
5	5	Перехватите пакет вечером у киоска
6	4	Статус: красный. Начинайте операцию
7	10	Пятый этаж, третья дверь - вход
8	11	Объект под наблюдением. Будьте осторожны
9	17	Код 12345: взломайте дверь
10	21	Тайна сокровищ в красной коробке

2. Используя систему Виженера, зашифруйте сообщение в соответствии с вашим вариантом приведенным в таблице. Задание выполнить в тетради.

Вариант	Сообщение	Ключевое слово
1	За пару секунд компьютер успевает сделать ошибку таких размеров, что сотни людей трудятся над ней месяцами	РАДОСТЬ
2	Первые криптографические системы были изобретены в глубокой древности, но не перестали развиваться в наши дни	УСПЕХ
3	Первые каналы связи были очень простыми, их организовывали с помощью надежных курьеров	ЛЕТО
4	Процесс преобразования открытого текста с целью сделать непонятным его смысл для посторонних называется шифрованием	УДАЧА
5	Знания бывают двоякого рода: либо мы что-нибудь знаем, либо мы знаем, где найти сведения об этом	ПРАЗДНИК
6	Параметр, с помощью которого выбирается отдельное	МЫШКА

	используемое преобразование, называется криптографическим ключом	
7	Расшифрование – процесс извлечения открытого текста из криптограммы при условии знания ключа	КСЕРОКС
8	Каждое криптографическое преобразование однозначно определяется ключом и описывается некоторым криптографическим алгоритмом	КАНИКУЛЫ
9	С развитием компьютерных технологий и вычислительной техники усложнились и методы шифрования	РАБОТА
10	Ключ передается отправителю и получателю сообщения таким образом, что его невозможно перехватить	РУЧКА

5. Порядок выполнения.

1. Определить исходные данные для задания в соответствии с вариантом.
2. Разработать и начертить блок-схему алгоритма задачи в задании 1.
3. Написать исходный текст программы.
4. Отладить программу.
5. Протестировать программу на минимальных, средних и максимальных значениях.
6. Выполнить программу на данных заданных преподавателем.
7. Защитить работу с предоставлением блок-схемы и исходного текста программы.

6. Контрольные вопросы:

1. Криптографическое шифрование.
2. Криптографическое расшифрование.
3. Целостность и неоспоримость шифра.
4. Понятие криптографических шифров.
5. Понятие криптографического ключа.
6. Криптографический алгоритм.
7. Симметричный алгоритм шифрования.
8. Алгоритм дешифрования «шифра замены».
9. Симметричный ключ «шифра замены».

10. Алфавитные «шифры замены».
11. Блочные «шифры замены».
12. Виды шифров замены.
13. Суть шифра замены.
14. Полиалфавитная замена.
15. Цепочная структура алфавитного «шифра замены».

Лабораторная работа № 2

Тема: Шифр перестановки и дешифрование шифра перестановки. Шифр замены по ГОСТ 28147-89.

1. Цель: Получить практические навыки по программированию алгоритма шифра перестановки и дешифрования “шифра перестановки”, а также получение практических навыков по составлению монограммного алгоритма и программирования шифра замены по ГОСТу.

Знать:

- метод алгоритма шифра перестановки;
- метод шифра замены по ГОСТ 28147-89 с ключом и таблицей замены.

Уметь:

- программировать на алгоритмических языках

2. Подготовка к работе:

- повторить теоретический материал согласно лекциям;

3. Теоретический материал:

В работе будут рассмотрены два алгоритма шифрования. Рассмотрим шифры перестановки:

Шифр обычной табличной перестановки (или перестановочный шифр) является одним из простых методов шифрования, основанным на перестановке символов открытого текста согласно заданной таблице перестановки. В отличие от шифра замены, где символы заменяются другими символами, в перестановочном шифре символы переставляются внутри текста.

Алгоритм шифрования:

В данном алгоритме ключом выступает размерность таблицы (количество строк и столбцов). Сообщение, которое будет кодироваться, записывается в таблицу поочередно по столбцам.

Например, сообщению «Рукописи не горят в огне» с ключом 5 будет соответствовать таблица, в которой открытый текст записывается последовательно по столбцам, зашифрованное сообщение получаем записывая построчно. Таблица шифрования приведена ниже:

Р	И	Г	В
У	С	О	О
К	И	Р	Г
О	Н	Я	Н
П	Е	Т	Е

Зашифрованным текстом будет: «РИГВ УСОО КИРГ ОНЯН ПЕТЕ».

Для расшифровки текста будет использоваться обратный ключ (длина строки) и будет строиться аналогичная таблица, в которой можно получить исходный текст.

Шифрующие таблицы с **двойной перестановкой** по ключу используют для повышения скрытности шифра. В данном методе используются два ключевых слова. Первое слово определяет перестановку столбцов, второе – перестановку строк таблицы. Перестановки производятся согласно порядку следования в алфавите символов ключевых слов.

На первом этапе исходный текст (или его фрагмент) построчно записывается в таблицу. Далее перестанавливаются столбцы исходной таблицы по первому ключевому слову. Затем переставляются строки полученной таблицы по второму ключевому слову. На последнем этапе из итоговой таблицы считывается шифртекст по столбцам.

Пример:

Зашифруем фразу с помощью таблицы размером 4х6 и ключевых слов «СКАНЕР» и «4123».

После заполнения исходной таблицы по строкам (см. таблицу 2.1) переставляем столбцы по порядку следования в алфавите букв слова «СКАНЕР» (см. таблицу 2.2). Затем переставляем строки. Порядковый номер строки определяет цифра второго ключевого слова «4123» (см. таблицу 2.3). На этом перестановки в таблице заканчиваются. Шифртекст считываем по столбцам и получаем: «ЙЛЕСП_ЕЕЫОМИ_ЬНТАИНМНРЗС»

Таблица 2.1 Исходная таблица

	С	К	А	Н	Е	Р
	6	3	1	4	2	5
4	С	И	С	Т	Е	М
1	Н	Ы	Й		П	А
2	Р	О	Л	Ь		И
3	З	М	Е	Н	Е	Н

Таблица 2.2 Таблица после перестановки столбцов

	А	Е	К	Н	Р	С
	1	2	3	4	5	6
4	С	Е	И	Т	М	С
1	Й	П	Ы		А	Н
2	Л		О	Ь	И	Р
3	Е	Е	М	Н	Н	З

Таблица 2.3 Таблица после перестановки строк

	А	Е	К	Н	Р	С
	1	2	3	4	5	6
1	Й	П	Ы		А	Н
2	Л		О	Ь	И	Р
3	Е	Е	М	Н	Н	З
4	С	Е	И	Т	М	С

Шифр замены по ГОСТу:

Перестановочный шифр с таблицей 4x8 или 8x4 для шифрования текста данным способом необходим словесный ключ, который будет использоваться для построения таблицы. Размерность таблицы всегда либо 8x4, либо 4x8. Сначала в таблице идет слово, выбранное ключом, а после него идут все буквы алфавита подряд, не включая те, что были в ключе. Например, ключом будет выступать слово БАНДЕРОЛЬ, тогда таблица будет выглядеть следующим образом:

Б	А	Н	Д	Е	Р	О	Л
Ь	В	Г	Ж	З	И	Й	К
М	П	С	Т	У	Ф	Х	Ц
Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я

При шифровании для каждой буквы исходного открытого текста будет выбираться буква, которая находится ниже этой буквы в таблице. В случае, когда буква находится в нижней строке таблицы, будет выбирать буква в верхней строке этого столбца. Если сообщением будет «рукописи не горят», то шифротекст – «иыцйшфщф гз сйилъ».

Для расшифровки будет использоваться обратный алгоритм.

4. Задание:

1. Разработать алгоритм и написать программу шифрования и дешифрования по методу «шифра перестановки» для русского текста в соответствии с вариантом. При выводе шифра учесть возможность вывода таблицы шифрования.

2. Написать программу шифрования русского текста длиной 32 символа с ключом и исходным текстом в соответствии с вариантом. При выводе шифра учесть возможность вывода таблицы шифрования.

Вариант	**Ключ**	**Сообщение**
1	Сокровище	Загадочная надпись на стене
2	Подземелье	Первый этаж, вторая дверь слева
3	Лабиринт	Подарок в красной коробке на чердаке
4	Приключение	Драгоценный камень в сундуке
5	Артефакт	Операция «Черный кот»
6	Пергамент	Расследование крупного мошенничества
7	Открытие	Команда специалистов за работой
8	Подарок	Впереди терра инкогнита
9	Опасность	Император двигал армию на север
10	Загадка	Александрийский обелиск

5. Порядок выполнения.

1. Определить исходные данные для задания в соответствии с количеством букв в своих фамилии и имени.
2. Разработать и начертить блок-схему алгоритма задачи.
3. Написать исходный текст программы.
4. Отладить программу.
5. Протестировать программу на минимальных, средних и максимальных значениях.
6. Выполнить программу на заданных преподавателем данных.
7. Защитить работу с предоставлением блок-схемы и исходного текста программы.

6. Контрольные вопросы:

1. Какой принцип работы алгоритма шифра перестановки?
2. Понятие симметричного алгоритма.
3. Какие есть ключи в шифрующих таблицах?
4. Что такое алгоритм простой перестановки?
5. Что такое алгоритм одиночной перестановки по ключу?
6. Что такое алгоритм двойной перестановки?
7. Понятие ключа шифра замены.
8. Как формируется таблица шифрования?
9. Какие есть требования к ключу шифрования по составу и длине?
10. Что служит полным ключом для шифрования?
11. Какие есть возможные варианты размерности таблицы для русского алфавита?
12. Как есть варианты размерности таблицы для английского алфавита?
13. Какие возможны варианты расположения алфавита в таблице?
14. В чем отличие методов шифрования перестановкой и шифрования по ГОСТ?

Лабораторная работа № 3

Тема: Шифрование методом гаммирования.

1. Цель: Приобретение практических навыков по составлению алгоритма и разработке программы шифрования методом “гаммирования”.

Знать:

- метод гаммирования по ГОСТ 28147-89.

Уметь:

- программировать на алгоритмических языках.

2. Подготовка к работе:

- повторить ГОСТ 28147-89, раздел шифрования методом “гаммирования”;
- повторить теоретический материал согласно лекциям.

3. Теоретический материал:

Шифрование методом гаммирования – это метод шифрования, основанный на комбинировании открытого текста с гаммой, что приводит к шифрованному тексту. Этот метод является формой потокового шифрования, где каждый символ открытого текста комбинируется с соответствующим символом из гаммы для получения зашифрованного символа. Гаммой может выступать как последовательность чисел, созданная псевдослучайный генератором, так и заранее заданное слово.

Обычное шифрование методом гаммирования выполняется следующим образом:

1. Каждой букве алфавита присваивается число, соответствующее его номеру в алфавите.
2. Открытый буквенный текст переводится в цифровой в соответствии с таблицей перевода.
3. Генерируется ключ гамма и ему так же присваивается цифровое значение.
4. Затем каждое число буквы исходного текста складывается с числом буквы гаммы, а после чего делится по модулю на длину алфавита, получая значения шифротекста. Если гамма короче текста, то она будет повторяться по кругу.

5. Далее все значения шифротекста переводятся в текст по ранее составленной таблице перевода из алфавита.

Расшифровка происходит по обратному алфавиту.

Пример:

Таблица числовых значений алфавита:

а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

ч	ш	щ	ъ	ы	ь	э	ю	я
23	24	25	26	27	28	29	30	31

Исходным текстом будет «Телефон», а ключом «звонок», тогда:

Исходные значения:

Т	Е	Л	Е	Ф	О	Н
З	В	О	Н	О	К	З

Заменим буквы числовыми значениями. После преобразования получим:

18	5	11	5	20	14	13
7	2	14	13	14	10	7

После сложения:

25	7	25	18	3	24	20
щ	з	щ	т	г	ш	ф

Зашифрованный текст – «щзщтгшф»

4. Задание:

1. Написать программу шифрования русского текста длиной до 256 символов методом гаммированя с ключом “гаммой” по варианту в таблице.

2. При выводе результата учесть возможность вывода таблицы с результатами

3. Выполнить проверку в соответствии с вариантом.

Вариант	Гамма	Сообщение
1	Дефект	Город Петербург
2	Копия	Корабли не поплывут
3	Красота	Обычный ход вещей
4	Симулятор	Линия судьбы

5	Порядок	Герой нашего времени
6	Талант	Элементарная математика
7	Дискотека	Простейшая геометрия
8	Вариант	Цикл итерации
9	Автор	Условный дефект
10	Дубликат	Концепция выступления

5. Порядок выполнения.

1. Определить исходные данные для задания в соответствии с вариантом.
2. Разработать и начертить блок-схему алгоритма задачи.
3. Написать исходный текст программы.
4. Отладить программу.
5. Протестировать программу на минимальных, средних и максимальных значениях.
6. Выполнить программу на данных заданных преподавателем.
7. Защитить работу с предоставлением блок-схемы и исходного текста программы.

6. Контрольные вопросы:

1. Понятие метода гаммирования.
2. Что такое “гамма шифр”?
3. Как генерируется гамма?
4. Как происходит подготовка открытого текста для шифрования?
5. Что такое моногамное шифрование?
6. Что такое полигамное шифрование?
7. От чего зависит секретность ключа гаммы?

Лабораторная работа № 4

Тема: Шифрование с имитовставкой.

1. Цель: Приобретение практических навыков по составлению алгоритма и разработке программы шифрования с выборкой имитовставки.

Знать:

- метод имитовставки.

Уметь:

- программировать на алгоритмических языках.

2. Подготовка к работе:

- повторить теоретический материал согласно лекциям.

3. Теоретический материал:

Шифрование:

1. Открытый текст «КУЗНЕЦОВРЕЗИДЕНТ» перевести в цифровой текст (например, по цифровому алфавиту).
2. Числовой текст разбивается на блоки длиной, равной длине ключа.
3. Перед шифрованием каждый блок складывается по модулю – 32 с предыдущим.
4. Из полученной числовой последовательности в результате суммирования всех блоков выбирается имитовставка требуемой длины (например, длиной с ключ, полключа, четверть ключа или др.) и присоединяется к передаваемому сообщению.
5. Сообщение с имитовставкой шифруется одним из алгоритмов замены, перестановки или гаммирования в цифровом виде.
6. Цифровой вид сообщения преобразуется в буквенный зашифрованный текст и предоставляется для передачи.
7. Пример реализации алгоритма шифрования текста длиной 16 букв по шифру замены с ключом «ОЩУЩЕНИЕ» (сдвиг по алфавиту на длину ключа) и имитовставкой длиной 75% от ключа (шесть позиций, присоединяемых к концу текста) приведен в следующей таблице:

Открытый	К	У	З	Н	Е	Ц	О	В
текст	Р	Е	З	И	Д	Е	Н	Т
Числовой	10	19	7	13	5	22	14	2
Текст	16	5	7	8	4	5	13	18
Сумма (по модулю 32)	26	24	14	21	9	27	27	20
Шифротекст	Т	Ы	П	Х	Н	Ю	Ц	К
(ключ шифра замены «ОЩУЩЕНИЕ»)	Ш	Н	П	Р	М	Н	Х	Ъ
Имитовставка длиной равной ключу	Ъ	Ш	О	Х	Й	Ы	-	-
Зашифрованная имитовставка	**В**	**А**	**Ц**	**Э**	**С**	**Г**	-	-
	Т	Ы	П	Х	Н	Ю	Ц	К
Зашифрованный	Ш	Н	П	Р	М	Н	Х	Ъ
текст	В	А	Ц	Э	С	Г		

8. Зашифрованный текст, подготовленный для передачи:

ТЫПХ НЮЦК ШНПР МНХЪ ВАЦЭСГ

Дешифрование:

1. Шифротекст поблочно дешифруется вместе с имитовставкой по известному ключу из лабораторной работы № 9.

2. По известной длине и ее расположению в шифротексте, заданных в предыдущей работе, имитовставка выделяется в отдельный блок.

3. Отдельный блок программы дешифровки самостоятельно суммирует по модулю-32 блоки расшифрованного текста и формирует контрольную имитовставку на основе дешифрованного текста.

4. Если контрольная имитовставка равна переданной, то выдается сообщение, что текст истинный, если имитовставки не совпадают – то текст ложный.

4. Задание:

1. Написать программу шифрования одним из методов (замены, перестановки, гаммирования) используя иммитовставку, в соответствии с вариантом в таблице.

Вариант	**Ключ**	**Сообщение**
1	Театр	Не Санкт-Петербург
2	Яблоко	Телескоп Фантазия

3	Потолок	Авангард молодежи
4	Прогресс	Валентин партизан
5	Попытка	Ворчащий вахтовик
6	Одеяло	Вбросить вдогонку
7	Эволюция	Устроили кавардак
8	Порядок	Паковать пальтище
9	Эпоха	Натереть Пармезан
10	Эмоция	Парфюмер самоучка

5. Порядок выполнения.

1. Определить исходные данные.
2. Разработать и начертить блок-схему алгоритма задачи.
3. Написать исходный текст программы.
4. Отладить программу.
5. Протестировать программу на минимальных, средних и максимальных значениях.
6. Выполнить программу на данных заданных преподавателем.
7. Защитить работу с предоставлением блок-схемы и исходного текста программы.

6. Контрольные вопросы:

1. Что такое метод имитовставки и для чего он используется?
2. Каковы основные компоненты метода имитовставки?
3. Как происходит генерация имитовставки для сообщения?
4. Как происходит верификация имитовставки со стороны получателя?
5. Как обеспечивается целостность сообщения при использовании метода имитовставки? Какие преимущества у метода имитовставки перед обычными методами шифрования?

Лабораторная работа № 5

Тема: Ассиметричное шифрование методом RSA.

1. Цель: Изучить порядок шифрования методом RSA.

Знать:

- метод шифрования RSA.

Уметь:

- программировать на алгоритмических языках.

2. Подготовка к работе:

- повторить теоретический материал согласно лекциям.

3. Теоретический материал:

В зависимости от структуры используемых ключей методы шифрования подразделяются на:

1. Симметричное шифрование: посторонним лицам может быть известен алгоритм шифрования, но неизвестна небольшая порция секретной информации — ключа, одинакового для отправителя и получателя сообщения; Примеры: Шифр Цезаря, DES, 3DES, AES, Blowfish, Twofish, ГОСТ 28147-89.

2. Асимметричное шифрование: посторонним лицам может быть известен алгоритм шифрования, и, возможно открытый ключ, но неизвестен закрытый ключ, известный только получателю. Криптографические системы с открытым ключом в настоящее время широко применяются в различных сетевых протоколах, в частности, в протоколах TLS и его предшественнике SSL (лежащих в основе HTTPS), а так же SSH, PGP, S/MIME и т. д. Российский стандарт, использующий асимметричное шифрование - ГОСТ Р 34.10-2001.

RSA (Rivest-Shamir-Adleman) — один из самых распространенных методов асимметричного шифрования, который использует комбинацию открытого и закрытого ключей для шифрования и дешифрования данных. Назван в честь его создателей — Рональда Ривеста, Ади Шамира и Леонарда Адлемана.

Его криптостойкость основывается на сложности разложения на множители больших чисел, а именно - на исключительной трудности задачи определить секретный ключ на основании открытого, так как для этого потребуется решить

задачу о существовании делителей целого числа. Наиболее криптостойкие системы используют 1024-битовые и большие числа.

Принцип работы RSA.

RSA (Rivest-Shamir-Adleman) является одним из наиболее распространенных методов асимметричного шифрования, который использует комбинацию открытого и закрытого ключей для шифрования и дешифрования данных.

Генерация ключей:

первый шаг в методе RSA — генерация открытого и закрытого ключей. Это включает выбор двух больших простых чисел p и q, их перемножение для получения модуля $n = p \times q$, а также нахождение числа Эйлера $\varphi(n) = (p-1) \times (q-1)$. Затем выбирается целое число e, взаимно простое с $\varphi(n)$, и находится целое число d, обратное e по модулю $\varphi(n)$ (т.е. $d \times e = 1 \bmod \varphi(n) \Rightarrow d = e^{-1} \, mod \, \varphi(n)$). Открытый ключ состоит из чисел n и e, а закрытый ключ — из n и d.

Шифрование: Для шифрования сообщения M используется открытый ключ (n, e). Сообщение M преобразуется в целое число m, где $0 \leq m < n$, и вычисляется зашифрованное сообщение C по формуле: C= $m^e \, mod \, n$.

Дешифрование: Зашифрованное сообщение C дешифруется с помощью закрытого ключа (n, d). Расшифрованное сообщение M вычисляется по формуле: M=$c^d \, mod \, n$.

4. Задание:

1. Написать программу шифрования русского текста длиной до 256 символов методом RSA в соответствии с вариантом.
2. Выполнить проверку работы программы.

Вариант	n, q	Сообщение
1	829, 379	Операция «Черный кот»
2	541, 5039	Александрийский обелиск
3	7993, 4729	Император двигал армию на север
4	409, 7411	Драгоценный камень в сундуке

5	7393, 727	Загадочная надпись на стене
6	4663, 3037	Расследование крупного мошенничества
7	7901, 6547	Команда специалистов за работой
8	4253, 8501	Впереди терра инкогнита
9	1979, 967	Подарок в красной коробке на чердаке
10	5923, 7487	Первый этаж, вторая дверь слева

5. Порядок выполнения.

Рассмотрим алгоритм RSA с практической точки зрения.

Для начала необходимо сгенерировать открытый и секретные ключи:

- возьмем два больших простых числа p и q.
- определим n, как результат умножения p на q ($n = p \times q$).
- выберем случайное число, которое назовем d. Это число должно быть взаимно простым (не иметь ни одного общего делителя, кроме 1) с результатом умножения $(p-1) \times (q-1)$.
- определим такое число e, для которого является истинным следующее соотношение $(e \times d)\ mod\ ((p-1) \times (q-1)) = 1$.
- назовем открытым ключом числа e и n, а секретным - d и n.

Для того, чтобы зашифровать данные по открытому ключу {e, n}, необходимо следующее:

- разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа M(i)=0,1,2..., n-1 (т.е. только до n-1).
- зашифровать текст, рассматриваемый как последовательность чисел M(i) по формуле $C(i) = m(I)^e\ mod\ n$.

Чтобы расшифровать эти данные, используя секретный ключ {d, n}, необходимо выполнить следующие вычисления: M(i) = $c(i)^d\ mod\ n$. В результате будет получено множество чисел M(i), которые представляют собой исходный текст.

Следующий примеры наглядно демонстрируют алгоритм шифрования RSA:

Пример 1:

Зашифруем и расшифруем сообщение "CAB" по алгоритму RSA. Для простоты возьмем небольшие числа - это сократит наши расчеты.

1. Выберем $p = 3$ и $q = 11$.
2. Определим $n = 3 \times 11 = 33$.
3. Найдем $(p-1) \times (q-1) = 20$. Следовательно, d будет равно, например, 3: $(d = 3)$.
4. Выберем число e по следующей формуле: $(e \times 3)\ mod\ 20 = 1$. Значит e будет равно, например, 7: $(e = 7)$.
5. Представим шифруемое сообщение как последовательность чисел в диапазоне от 0 до 32 (не забывайте, что кончается на $n-1$). Буква $\mathrm{A} = 1, \mathrm{B} = 2, \mathrm{C} = 3$.

Теперь зашифруем сообщение, используя открытый ключ {7, 33}

C1 = 3^7 mod 33 = 2187 mod 33 = 9;

C2 = 1^7 mod 33 = 1 mod 33 = 1;

C3 = 2^7 mod 33 = 128 mod 33 = 29.

Шифротекст – 9, 1, 29.

Теперь расшифруем данные, используя закрытый ключ {3, 33}.

M1 = 9^3 mod 33 = 729 mod 33 = 3;

M2 = 1^3 mod 33 =1 mod 33 = 1;

M3 = 29^3 mod 33 = 24389 mod 33 = 2.

Расшифрованное сообщение - 3, 1, 2.

Пример 2:

Зашифруем сообщение «ЕЛЬ» по алгоритму RSA.

1. Выберем $p = 5$ и $q = 13$.
2. Определим $n = 5 \times 13 = 65$.
3. Найдем $(p-1) \times (q-1) = 48$. Следовательно, d будет равно, например, 5: (d = 5).
4. Выберем число e по следующей формуле: $5 \times mod\ 48 = 29$. Значит e будет равно, например, 29: (e = 29).

5. Представим шифруемое сообщение как последовательность чисел в диапазоне от 0 до 32 (не забывайте, что кончается на n-1). Буква Е =6, Л=13, Ь=30.

Теперь зашифруем сообщение, используя открытый ключ {29, 65}

$C_1 = 6^{29} \bmod 65 = 368456532867888929832 96 \bmod 65 = 41;$

$C_2 = 13^7 \bmod 65 = 20153812643461115079850395637 1773 \bmod 65 = 13;$

$C_3 = 30^7 \bmod 65 = 6863037736488300000000000000000000000000000 \bmod 65 = 10.$

Шифротекст – 41, 13, 10.

Теперь расшифруем данные, используя закрытый ключ {3, 33}.

$M_1 = 41^5 \bmod 65 = 115856201 \bmod 65 = 6;$

$M_2 = 13^5 \bmod 65 = 371293 \bmod 65 = 13;$

$M_3 = 10^5 \bmod 65 = 100000 \bmod 65 = 30.$

Расшифрованное сообщение - 6, 13, 30.

6. Контрольные вопросы:

1. Что такое криптография и какие основные методы шифрования существуют?
2. Как работает алгоритм RSA?
3. Какие основные математические операции используются в алгоритме RSA?
4. Как происходит генерация открытого и закрытого ключей в RSA?
5. Как осуществляется процесс шифрования в RSA?
6. Как происходит процесс дешифрования в RSA?
7. Какие угрозы безопасности могут быть связаны с использованием RSA?
8. Какие меры безопасности рекомендуется принимать при использовании RSA?

Лабораторная работа № 6

Тема: Алгоритмы поведения вирусных и других вредоносных программ**.**

1. Цель: Знакомство с некоторыми алгоритмами поведения вирусных и других вредоносных программ.

Знать:

- основы работы с операционной системой, используя языки программирования и основные принципы работы вирусов.

Уметь:

- программировать на алгоритмических языках.

2. Подготовка к работе:

- повторить теоретический материал согласно лекциям.

3. Теоретический материал:

Компьютерные вирусы - это программы, которые могут копировать себя и распространяться без ведома пользователя. Они могут нанести вред компьютеру, уничтожая данные или вызывая сбои в работе системы, а некоторые вирусы также могут нести несерьезный вред для пользователя.

Компьютерные вирусы могут распространяться через различные каналы: через интернет, электронную почту, съемные носители информации (флешки, диски), социальные сети и т.д.

Для написания программы, которая будет имитировать действия компьютерного вируса, можно использовать различные языки программирования, такие как Python, Java, C++ и другие.

В разработке программы для данной работы может быть полезна встроенная библиотека os на языке Python.

4. Задание:

1. Разработать программу, имитирующую некоторые (см. вариант задания в таблице 6.1) действия вируса или другой вредоносной программы и подготовить отчет о проделанной работе.

5. Порядок выполнения.

1. Определите язык программирования для написания программы.
2. Разработать и начертить блок-схему алгоритма задачи.

3. Написать исходный текст программы.

4. Отладить программу.

5. Выполнить программу на данных заданных преподавателем.

6. Защитить работу с предоставлением блок-схемы и исходного текста программы.

6. Контрольные вопросы:

1. Что такое компьютерный вирус?

2. Какие типы компьютерных вирусов вы знаете?

3. Как компьютерные вирусы могут распространяться?

4. Какие методы защиты от компьютерных вирусов вы знаете?

5. Какие инструменты можно использовать для обнаружения и удаления компьютерных вирусов?

6. Какие последствия могут быть вызваны компьютерными вирусами?

7. Какие меры предосторожности можно принять, чтобы защитить свой компьютер от компьютерных вирусов?

8. Какие шаги нужно выполнить, чтобы написать программу, имитирующую действия компьютерного вируса?

9. Какие языки программирования можно использовать для написания программы, имитирующей действия компьютерного вируса?

10. Какие действия можно сымитировать в программе, чтобы она выглядела, как настоящий компьютерный вирус?

Таблица 6.1 Варианты заданий

Вариант	Действие вирусной или другой вредоносной программы	Задание	Входные данные и процедуры	Выходные данные и процедуры	Примечание
1	«Поиск жертвы»	Разработать и отладить процедуру поиска файлов по заданной маске в текущей папке	Маска файла	Массив имен найденных файлов и количество	Маска файлов: *.exe
2	«Подмена файла»	Разработать и отладить процедуру замены указанного файла на другой указанный файл	Имена двух файлов	Признак успешности замены	Подмена путем удаления файла и использования его имени для изменения имени «вирусного» файла
3	«Модификация файла»	Разработать и отладить процедуру записи информации из одного указанного файла в конец другого указанного файла	Имена двух файлов	Преобразованный файл	
4	«Принцип действия программы-шпиона»	Разработать процедуру копирования содержимого всех файлов, к которым обращается пользователь в заданный файл	Имя файла	Преобразованный файл	В программе создать форму обращения к нескольким файлам
5	«Создание звуковых эффектов»	Разработать процедуру запуска звукового файла при наступлении какого-либо события	Имя файла, каталога, заданное время и др.	Запуск звукового файла	
6	«Проверка файла на зараженность»	Разработать и отладить процедуру поиска заданной строки в файле	Имя файла, строка текста	Логическая переменная	

	(вирус не должен заражать уже зараженные файлы)				
7	«Маскировка»	Разработать и отладить процедуру запуска указанной программы на выполнение	Имя файла с программой	Логическая переменная	Принцип маскировки: после выполнения заданного алгоритма вирус запускает на выполнение сам файл, чтобы скрыть от пользователя факт заражения
8	«Маскировка»	Разработать и отладить процедуру изменения даты и времени создания указанного файла на заданные значения	Имя файла, значения даты, времени	Логическая переменная	
9	«Маскировка»	Разработать и отладить процедуру изменения атрибутов и размера указанного файла на заданные значения	Имя файла, значения атрибутов и размера	Логическая переменная	
10	«Принцип работы логической бомбы»	Разработать и отладить процедуру удаления содержимого всех файлов в заданном каталоге при наступлении какого-либо события	Заданное значение времени, имя каталога	Логическая переменная	
11	«Имитация сбоев ОС и	Разработать и отладить процедуру появления	Имя файла (файлов)		

	аппаратуры»	сообщения о сбое ОС и/или аппаратуры при попытке доступа к какому-либо файлу			
12	«Маскировка»	Разработать и отладить процедуру шифрования и дешифрования указанного файла	Имя файла	Логическая переменная	
13	«Принцип работы логической бомбы»	Разработать и отладить процедуру запуска некоторой программы при наступлении какого-либо часа или минуты	Заданное значения времени, имя файла		
14	«Принцип работы баннера»	Разработать и отладить процедуру запуска баннера при попытке перехода по какой-либо ссылке	Ссылка, баннер		
15	«Принцип работы логической бомбы»	Разработать и отладить процедуру удаления всех файлов каталога при наступлении какого-либо времени	Заданное значение времени, имя каталога	Логическая переменная	
16	«Модификация файла»	Разработать и отладить процедуру записи информации из одного указанного файла в конец другого указанного файла	Имена двух файлов	Преобразованный файл	Место в середине файла должно выбираться случайным образом
17	«Принцип работы логической бомбы»	Разработать и отладить процедуру запуска некоторой программы при наступлении какой-либо	Дата, имя запускаемого файла		

		даты			
18	«Жадная программа»	Разработать и отладить процедуру создания копий заданного файла и размещения их в заданном каталоге	Имя файла, количество копий, заданный каталог		
19	«Принцип действия клавиатурного шпиона»	Разработать и отладить процедуру записи в заданный файл информации, вводимой пользователем в поля «логин» и «пароль»	Имя файла	Преобразованный файл	За основу взять программу разработанную при выполнении ПР №4
20	«Принцип работы логической бомбы»	Разработать и отладить процедуру запуска некоторой программы при наступлении какого-либо дня недели	День недели, имя запускаемого файла		
21	«Поиск жертвы»»	Разработать и отладить процедуру поиска файлов по заданной маске в текущей папке	Маска файла		Маска файлов: *.bat
22	«Модификация файла»	Разработать и отладить процедуру перемешивания символов в файле	Имя файла	Преобразованный файл	
23	«Маскировка»	Разработать и отладить процедуру подмены зараженного файла незараженным при обращении к нему пользователя	Имя файла	Открытый файл	Для проверки программы создать два файла: «зараженный» и «незараженный»
24	«Подмена файла»	Разработать и отладить процедуру замены	Имена двух файлов	Признак успешности замены	Подмена путем замены содержимого файла на

		указанного файла на другой указанный файл			содержимое «вирусного» файла
25	«Жадная программа»	Разработать и отладить процедуру создания копий заданного каталога	Заданный каталог, количество копий	Созданные копии	
26	«Проверка файла на зараженность» (вирус не должен заражать уже зараженные файлы)	Разработать и отладить процедуру проверки следующего факта: является ли содержимое двух файлов одинаковым	Имена двух имеющихся файлов	Логическая переменная	
27	«Проверка файла на зараженность» (вирус не может заражать сам себя)	Разработать и отладить процедуру проверки следующего факта: не является ли некоторый файл, заданным своим именем, той программой, которая эту проверку производить (т.е. самой запущенной программой)	Проверяемое имя файла	Логическая переменная	

Лабораторная работа № 7

Тема: Алгоритмы предупреждения и обнаружения вирусных угроз.

1. Цель: Знакомство с некоторыми алгоритмами предупреждения и обнаружения вирусных угроз.

Знать:

- основы работы с операционной системой, используя языки программирования и основные принципы работы вирусов.

Уметь:

- программировать на алгоритмических языках.

2. Подготовка к работе:

- повторить теоретический материал согласно лекциям.

3. Теоретический материал:

Алгоритмы предупреждения и обнаружения вирусных угроз - это методы и технологии, которые используются для защиты компьютеров и сетей от вредоносных программ, таких как компьютерные вирусы, трояны, черви и другие.

Существует несколько основных алгоритмов предупреждения и обнаружения вирусных угроз:

1. Эвристический анализ: этот метод использует набор правил и шаблонов для обнаружения вредоносных программ. Он основан на анализе поведения программы и ее взаимодействия с операционной системой.
2. Сигнатурный анализ: этот метод использует базу данных известных вредоносных программ для обнаружения новых экземпляров. Он сравнивает сигнатуры (уникальные характеристики) вредоносных программ с сигнатурами в базе данных.
3. Heuristic-based detection: этот метод использует эвристический анализ для обнаружения новых вредоносных программ, которые еще не были добавлены в базу данных. Он анализирует поведение программы и ее взаимодействие с операционной системой.
4. Sandboxing: этот метод использует виртуальную среду для запуска подозрительных программ. Если программа ведет себя подозрительно, она блокируется.

5. Белые списки: этот метод использует список разрешенных программ для запуска на компьютере. Если программа не находится в списке, она блокируется.

6. Обнаружение на основе поведения: этот метод использует анализ поведения программы для обнаружения вредоносных действий. Он отслеживает, какие файлы программа открывает, какие процессы запускает и т.д.

7. Обнаружение на основе репутации: этот метод использует базу данных репутации для обнаружения вредоносных программ. Он анализирует репутацию файла или программы на основе отзывов пользователей и других факторов.

8. Обнаружение на основе сетевого трафика: этот метод использует анализ сетевого трафика для обнаружения вредоносных программ. Он отслеживает, какие данные программа отправляет и получает.

Существуют также другие методы и технологии для предупреждения и обнаружения вирусных угроз, но эти являются наиболее распространенными.

4. Задание:

1. Разработать программу имитирующую некоторые (см. таблицу 7.1) действия по предупреждению вирусных угроз, обнаружению и удалению вирусных и других вредоносных программ и подготовить отчет о проделанной работе.

5. Порядок выполнения.

1. Определите язык программирования для написания программы.
2. Разработать и начертить блок-схему алгоритма задачи.
3. Написать исходный текст программы.
4. Отладить программу.
5. Выполнить программу на данных заданных преподавателем.
6. Защитить работу с предоставлением блок-схемы и исходного текста программы.

6. Контрольные вопросы:

1. Что такое алгоритмы предупреждения и обнаружения вирусных угроз?

2. Какие методы используются для защиты компьютеров и сетей от вредоносных программ?

3. Что такое эвристический анализ и как он работает?

4. Что такое сигнатурный анализ и как он работает?

5. Что такое heuristic-based detection и как он работает?

6. Что такое sandboxing и как он работает?

7. Что такое белые списки и как они используются для защиты от вредоносных программ?

8. Что такое обнаружение на основе поведения и как оно работает?

9. Что такое обнаружение на основе репутации и как оно работает?

10. Что такое обнаружение на основе сетевого трафика и как оно работает?

Таблица 7.1 Варианты заданий

Вариант		Задание	Входные данные процедуры	Выходные данные процедуры	Дополнительные условия
1	Алгоритм работы антивирусной программы-ревизора	Ревизоры запоминают исходное состояние файлов/каталогов, тогда, когда компьютер еще не заражен вирусом, а затем периодически сравнивают текущее состояние с исходным. Если обнаружены изменения, то на экран дисплея выводятся сообщения. Разработать процедуру поиска заданных изменений в файле/каталоге.	Имя файла (файлов)/Имя каталога (каталогов)	Сообщение о наличии/отсутствии изменений	Поиск изменений в дате и времени создания файла
2					Поиск изменений в атрибутах и размере файла
3					Поиск изменений в содержании файла
4					Поиск изменений в

					содержании каталога
5	Обнаружение файлов-компаньонов	Программа должна осуществлять поиск файлов-компаньонов (исполняемые файлы с тем же названием, что и исходный файл, но другим расширением) и по решению пользователя осуществлять следующие действия: (см. доп усл.)	Имя файла	Список обнаруженных файлов-компаньонов	Удаление файлов-компаньонов
6					Перемещение файлов-компаньонов в другой каталог (на карантин)
7	Обнаружение признаков заражения вирусом	Разработать процедуру обнаружения копий файлов в заданном каталоге. Осуществлять поиск по имени файла и по содержимому Информировать пользователя. каталог (на карантин) Предлагать на выбор следующие действия:	Имя файла, каталога	Список обнаруженных копий	Удаление обнаруженных копий
8					Перемещение обнаруженных копий в другой
					каталог (на карантин)
9	Профилактика заражения вирусом (Резервное копирование	Разработать процедуру создания резервных копий. Предусмотреть возможность выбора пользователем	Имя каталога	Логическая переменная	Периодичность копирования: раз в неделю (предоставить возможность выбора дня недели)

10		периодичности создания резервных копий (см. доп. усл.). При этом должны делаться копии только тех файлов, которые были созданы или изменены в период после предыдущей процедуры копирования.			Периодичность копирования: через день (предоставить возможность выбора четных или нечетных чисел)
11					Периодичность копирования: раз в несколько часов (предоставить возможность выбора интервала времени, проходящего между процедурами копирования)
12	Обнаружение вирусного кода в теле файла	Разработать и отладить процедуру поиска заданной строки целиком или частично в заданных файлах (см. доп. усл.). В случае обнаружения вирусного кода в теле файла реализовать следующий алгоритм «лечения»: (см. доп. усл.)	Строка, имя файла (файлов, каталога)	Логическая переменная	Поиск заданной строки и ее фрагментов (слов) в указанном файле. Алгоритм "лечения": удаление строки или ее Фрагментов
13					Поиск заданной строки и ее фрагментов (слов) в указанном файле. Алгоритм "лечения": перемещение зараженного файла в другой каталог (на карантин)
14					Поиск заданной строки и ее фрагментов (слов) в указанном файле. Алгоритм "лечения": удаление зараженного файла
15					Поиск заданной строки во

					всех текстовых файлах заданного каталога. Алгоритм "лечения": удаление строки из всех файлов
16					Поиск заданной строки во всех текстовых
17					файлах заданного каталога. Алгоритм "лечения": перемещение зараженных файлов в другой каталог (на карантин)
18	Защита от клавиатурных шпионов	Разработать генератор одноразового пароля на основе псевдослучайного выбора символов из данных, введенных пользователем. Применить следующий алгоритм ГПСЧ: (см. доп. усл.)	Массив с набором данных пользователя (для упрощения задачи: каждый элемент массива - цифра из данных пользователя: номера паспорта, даты рождения и Т.П.)	Одноразовы й пароль длинной N символов	X=round(10* sin(i* sin(i/Y)) +10), где X -выбираемый номер элемента в массиве i - счетчик [1;N]; У- элемент в массиве данных пользователя, N=10. Минимальное количество элементов массива: 20
19					Линейный конгруэнтный метод (функция Random), N=12
20					Метод Фибоначчи с запаздываниями": $X_i =$ $\begin{cases} Y_{i-a} - Y_{i-b}, \text{если } Y_{i-a} \geq Y_{i-} \\ -(Y_{i-a} - Y_{i-b}), \text{если } Y_{i-a} \leq Y \end{cases}$ где X -выбираемый номер

					элемента в массиве i - счетчик [max(a,b)+1; N+ max(a,b)+1]; У- элемент в массиве данных пользователя, a,b - целые положительные числа, называемые лагами, рекомендуемые значения (a,b)=(17,5), N=7. Минимальное количество элементов массива: max(a,b)
21	Защита от массовой рассылки спама методом CAPTCHA	Разработать программу регистрации пользователя с проверкой методом CAPTCHA. Для завершения регистрации пользователю должна быть предложена такая задача, которую с лёгкостью может решить человек, но которую несоизмеримо сложнее решить компьютеру (см. доп. усл.).	Данные пользователя	Логическая переменная	В качестве задачи предложить пользователю ввести число (слово) с картинки (одной или нескольких)
22					В качестве задачи предложить пользователю произвести показанную на картинке простую арифметическую операцию
23					В качестве задачи предложить пользователю выбрать из нескольких картинок одну, соответствующую определенному условию
24	Защита от программ	Разработать "умную" программу запроса паролей	Пароль (см. доп. усл.)	Логическая переменная	Разработать программу, запрашивающую только

	открытия пароля	(smart password asker). Этот метод предполагает использование специальной программы запроса паролей, которая работает не по стандартному алгоритму, а по алгоритму с псевдослучайным исходом (см. доп. усл.). Реализовать 7-10 запросов пароля			часть пароля: первые три символа
25					Разработать программу, запрашивающую не сам пароль, а сумму цифр, входящих в пароль
26					Разработать программу, запрашивающую не сам пароль, а сумму части пароля и числа месяца текущей даты
27					Разработать программу, запрашивающую только часть пароля: первый, третий и последний символ

Лабораторная работа № 8

Тема: Работа протокола STP на коммутаторах

1. Цель: закрепить теоретические знания и получить практические навыки по работе протокола связующего дерева на коммутаторах.

Знать:

- назначение коммутатора;
- общее устройство и принцип работы коммутатора;
- назначение и принцип работы протокола STP.

Уметь:

- работать в Cisco Packet Tracer;
- строить локальную сеть в сетевом симуляторе.

2. Подготовка к работе:

- повторить назначение и принцип работа протокола STP;

3. Задание:

3.1 Построить локальную сеть (рис. 8.1), состоящую из пяти компьютеров и пяти коммутаторов (номер сети для своего варианта смотреть в таблице 8.1).

3.2 Определить адресные параметры сети своего варианта

3.3 Ввести команду на коммутаторе, которая покажет наличие корневого моста.

3.4 Отметить коммутатор, который является приоритетным.

3.5 Отсоединить от корневого коммутатора все соединения, кроме одного. Определить какой коммутатор стал корневым. С компьютера этого коммутатора пропинговать в режиме симуляции остальные компьютеры.

3.6 Посмотреть путь передачи пакета, объяснить, как связаны между собой протоколы ARP и ICMP, а также почему выбираются такие маршруты передачи пакета ICMP.

3.7 Составить отчет в электронном виде, приложив свою схему при полносвязанной топологии и схему после изменения.

3.8 Составить подробные объяснения принципа работы протокола связующего дерева применительно к своей схеме.

3.9 Описать, что могло бы произойти, если бы коммутаторы не поддерживали протокол STP.

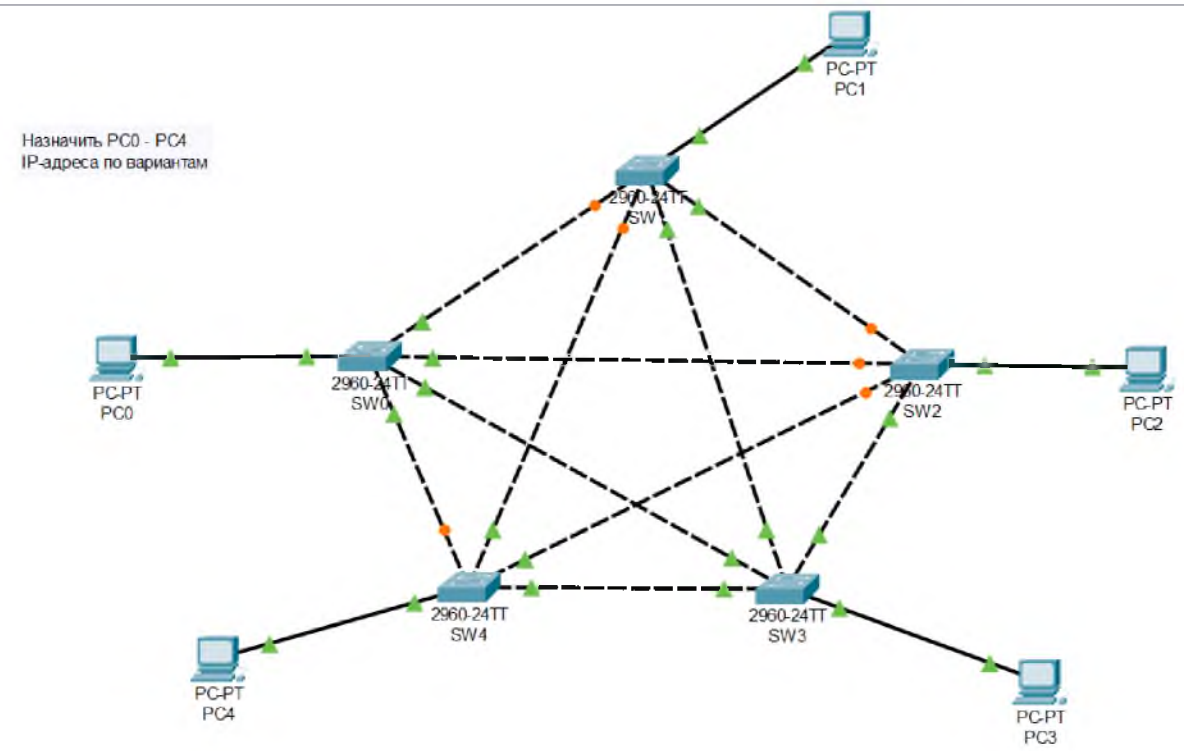


Рисунок 8.1. Схема сети

Таблица 8.1 – Таблица вариантов

№	IP-Адрес сети
1	125.106.202.0/21
2	241.193.119.0/24
3	144.168.0.0/17
4	41.107.1.0/23
5	8.167.11.0/24
6	113.133.0.0/21
7	225.105.77.0/24
8	218.230.23.0/25
9	105.126.0.0/19
10	200.129.1.0/23
11	180.129.205.0/24
12	17.8.0.0/16
13	107.202.246.0/25
14	195.120.1.0/19
15	160.6.0.0/18
16	47.122.1.0/21
17	136.17.214.0/24
18	81.252.110.0/19
19	100.63.150.0/24
20	132.109.20.0/21
21	20.240.0.0/23
22	215.179.12.0/16
23	54.11.0.0/21

24	134.172.0.0/23
25	242.156.122.0/24
26	74.217.13.0/19
27	20.135.0.0/21
28	252.63.239.0/28
29	82.140.0.0/24
30	59.15.1.0/23

4. Порядок выполнения

4.1 Построить локальную сеть, состоящую из пяти компьютеров и пяти коммутаторов (рисунок 4.1). Обязательно подписать IP-адресацию для компьютеров сети.

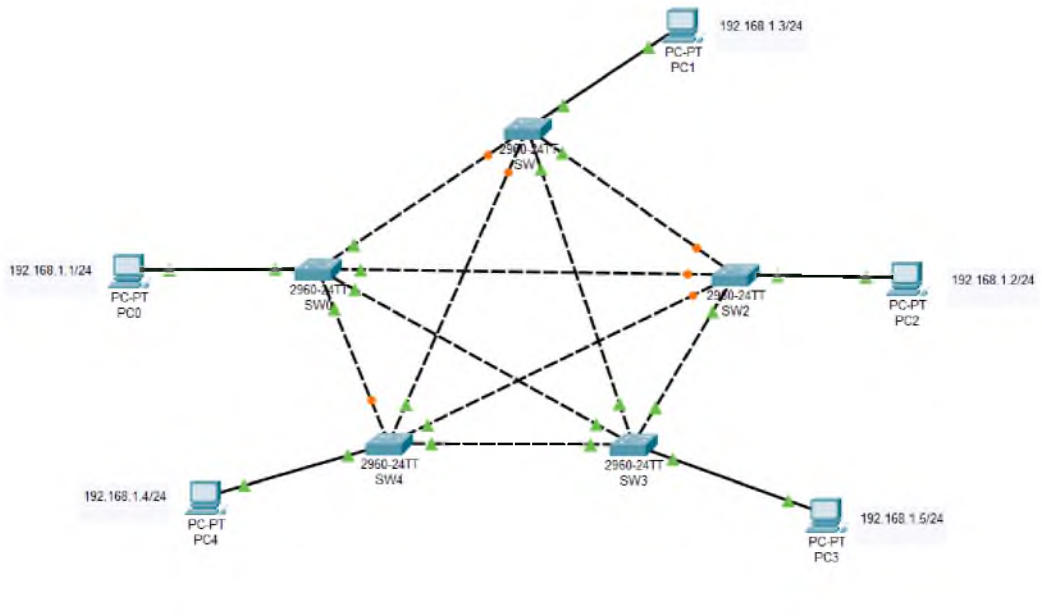


Рисунок 4.1 – Топология локальной сети

4.2 Для определения адресных параметров сети требуется зайти на статью «Определение адресных параметров сети по IP-адресу и префиксу» - https://dzen.ru/a/Yyv82Gg8qkr7OWYE

На примере сети 10.0.0.0/25

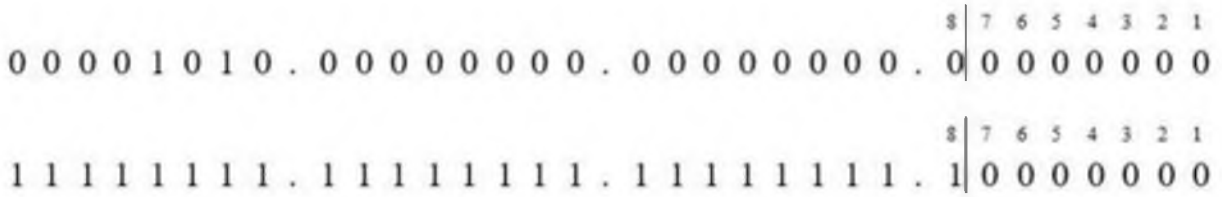
Необходимо определить номер сети, маску сети, широковещательный адрес сети, диапазон и количество адресов.

Ход решения:

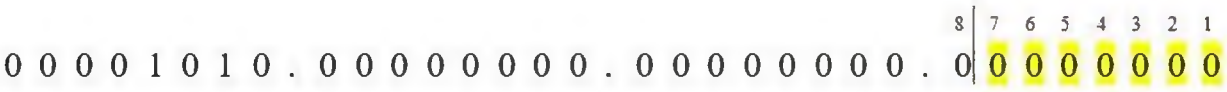
1. Переведем IP-адрес и префикс сети в двоичную систему счисления. Двоичный код адреса запишем первым, ниже запишем префикс. Число, обозначающее префикс показывает количество бит, отведенных под номер сети. В данном случае

это 25 единиц, остальное нули (так как IP-адрес четвертой версии протокола IP состоит из 32 бит). В данном виде записывается маска в двоичной системе счисления. Биты адреса и префикса записываем на одной вертикальной линии.

Принимаем нумерацию бит справа налево. То есть самый правый бит нумеруем как первый, а самый левый как тридцать второй. Затем определим границу сети в соответствии с маской (по правую сторону от границы должны быть только нули, по левую сторону – только единицы), в данном случае граница сети проходит между восьмым и седьмым битами (под номер сети отводится 25 бит).



2. Определяем номер сети и маску сети. Для этого все биты, принадлежащие IP-адресу узла и находящиеся справа от границы сети, заменяем нулями, а те биты, что слева, – переписываем без изменений:



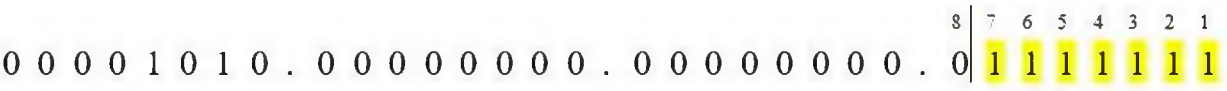
Переводим номер сети в десятичную систему счисления:

10.0.0.0.

Префикс, записанный в первом пункте в двоичном коде, также переводим в десятичную систему счисления и вычисляем маску сети:

255.255.255.128

3. Находим широковещательный адрес данной сети. Для этого все, что в номере сети находится слева от границы, записываем без изменений, а все, что справа, - заполняем единицами:



Переводим широковещательный адрес в десятичную систему счисления:

10.0.0.127.

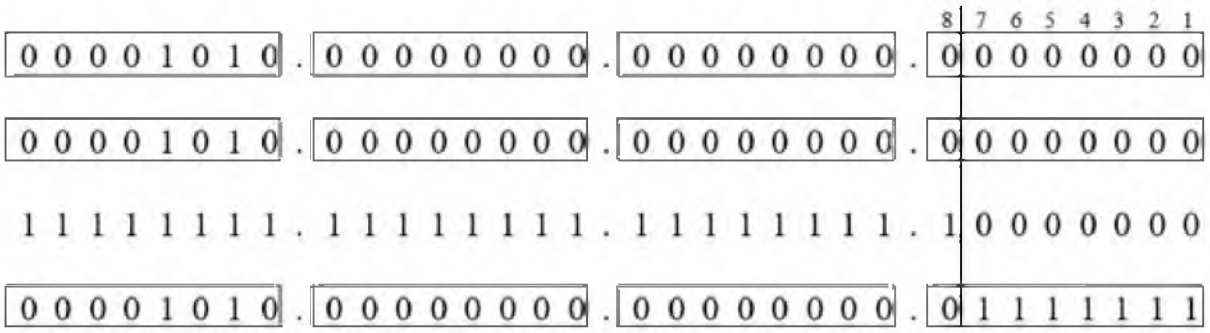
4. Теперь необходимо определить диапазон и количество адресов узлов в сети. Нужно понимать, что нумерация сети состоит из непрерывного диапазона

адресов. При этом самый первый адрес (не обязательно заканчивающийся на ноль) - это адрес сети, а самый последний - это широковещательный адрес сети (для групповой рассылки всем узлам сети). Соответственно адресация узлов каждой сети находится между этими двумя значениями. Таким образом, для того чтобы вычислить адрес первого узла в сети, необходимо к номеру сети прибавить единицу (10.0.0.1), а для того чтобы определить адрес последнего узла, - от широковещательного адреса сети отнять единицу (10.0.0.126). Получаем следующий диапазон адресов узлов:

10.0.0.1 – 10.0.0.126.

Таким образом, максимальное количество адресов в сети 10.0.0.0/25 составляет 126 (от 10.0.0.1 до 10.0.0.126).

Пример записи решения:



(1 строка – IP-адрес узла, 2 – номер сети, 3 – маска сети, 4 – широковещательный адрес сети)

Преобразуем все записи из двоичной системы счисления в десятичную:

Номер сети: 10.0.0.0

Маска: 255.255.255.128

Широковещательный IP-адрес: 10.0.0.127

Адрес первого узла в сети: 10.0.0.1

Адрес последнего узла в сети: 10.0.0.126

Количество адресов (максимально возможное количество узлов в данной сети) составляет 126 единиц.

4.3 Для определения корневого коммутатора на схеме руководствуйтесь правилом: все порты коммутатора должны гореть зелёным сигналом, порты соединённого коммутатора должны гореть зелёным сигналом (рисунок 4.2).

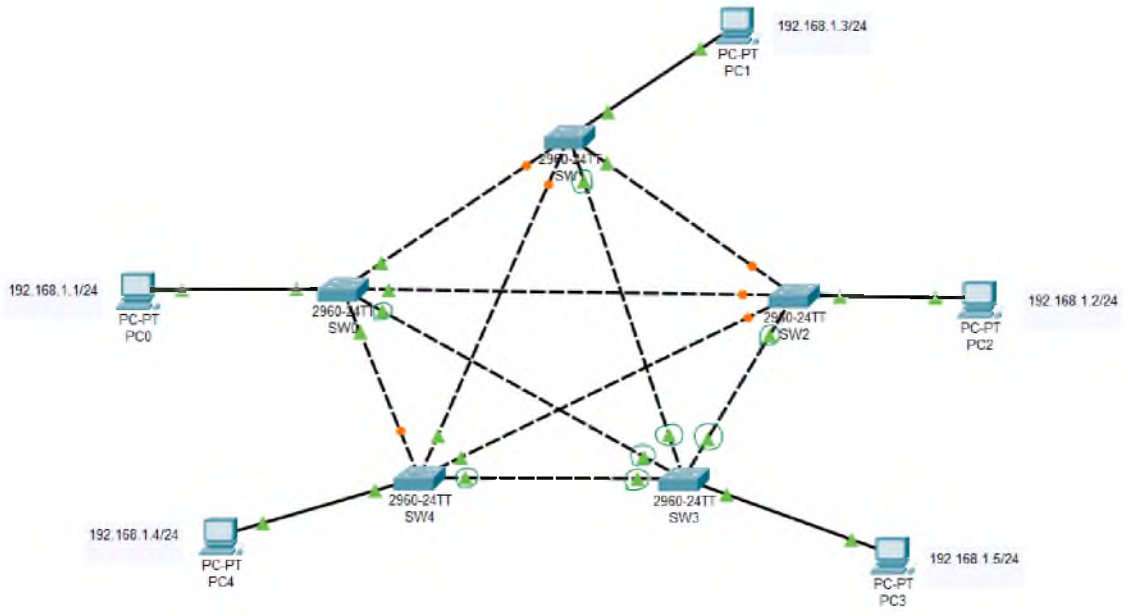


Рисунок 4.2 – Определение корневого коммутатора по схеме

Для просмотра корневого коммутатора воспользуйтесь командой: ***show spanning-tree*** (рисунок 4.3).

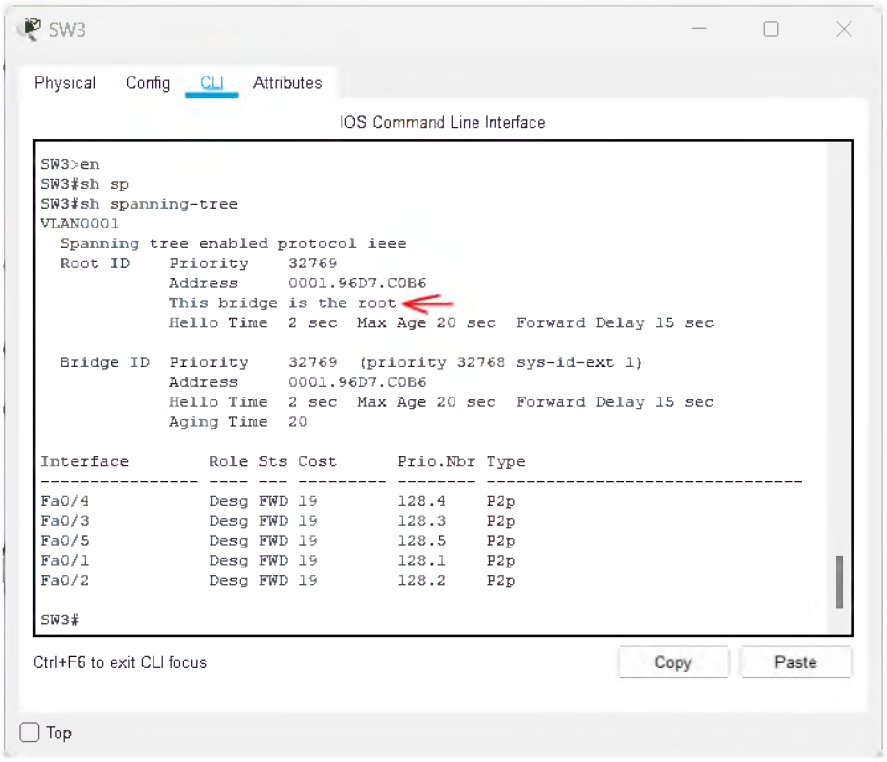


Рисунок 4.3 – Вывод команды show spanning-tree на корневом коммутаторе

Сообщение «This bridge is the root» говорит о том, что данный коммутатор является корневым в топологии сети. Протокол STP предусматривает несколько ролей работы с трафиком:

1. Root (корень): это роль, которая присваивается коммутатору, который является центральным элементом в топологии сети. Корневой коммутатор выбирается на основе наименьшего значения Bridge ID (идентификатора моста) в сети. Он является отправной точкой для определения путей к другим коммутаторам в сети.

2. Designated (desg - назначенный): это роль, которая присваивается порту на коммутаторе, который имеет наименьшую стоимость пути к корневому коммутатору в топологии для определенного сегмента сети. Порт с ролью "desg" используется для передачи трафика внутри сегмента.

3. Alternate (alt - альтернативный): это роль, которая присваивается порту, который является резервным для порта с ролью "desg" на том же коммутаторе. Порт с ролью "alt" готов к работе, если порт с ролью "desg" выйдет из строя или перестанет функционировать корректно.

4.4 Корневой коммутатор можно также определить исходя из вывода ***команды show spanning-tree*** на любом другом коммутаторе. Тип интерфейса который подключён к кореновому коммутатору будет иметь роль «Root» (рисунок 4.4.1).

Как видно из таблицы, коммутатор SW4 подключён к корневому коммутатору на интерфейсе FastEthernet0/2, на рисунке 4.3 указан корневой коммутатор (рисунок 4.4.2).

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     000C.CF10.E21C
              Cost        19
              Port        2(FastEthernet0/2)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15
sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     00D0.5854.6E65
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15
sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- --------
--------------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/2            Root FWD 19        128.2    P2p
Fa0/4            Desg FWD 19        128.4    P2p
Fa0/3            Desg FWD 19        128.3    P2p
```

Рисунок 4.4.1 – Определение корневого коммутатора

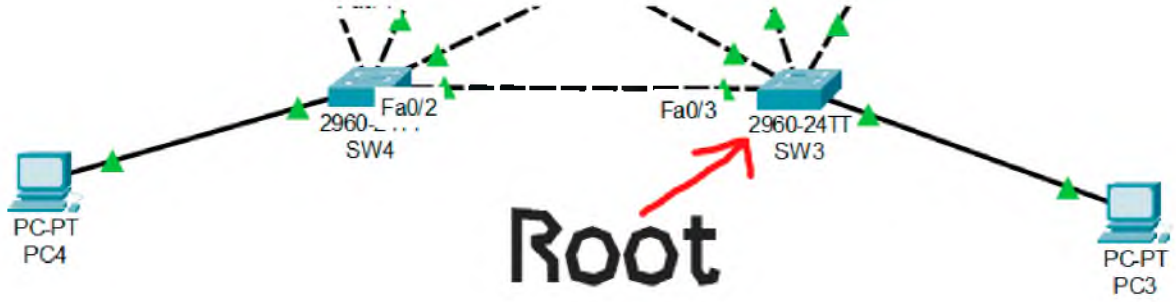


Рисунок 4.4.2 – Выявление корневого коммутатора

4.5 Произвольно выберите коммутатор, для которого отключите все порты кроме одного. Пропингуйте с компьютера, подключённого к этому коммутатору остальные компьютеры в сети (рисунки 4.5 – 4.7).

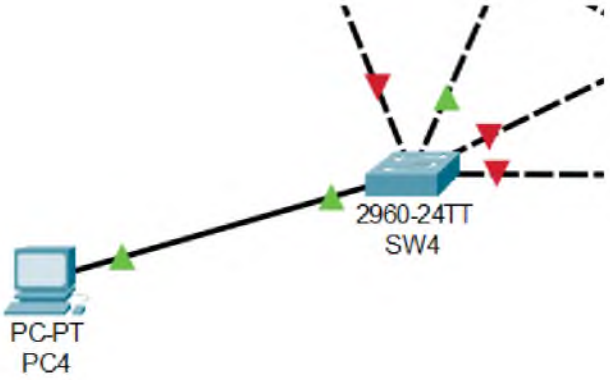


Рисунок 4.5 – Отключение портов на коммутаторе SW4

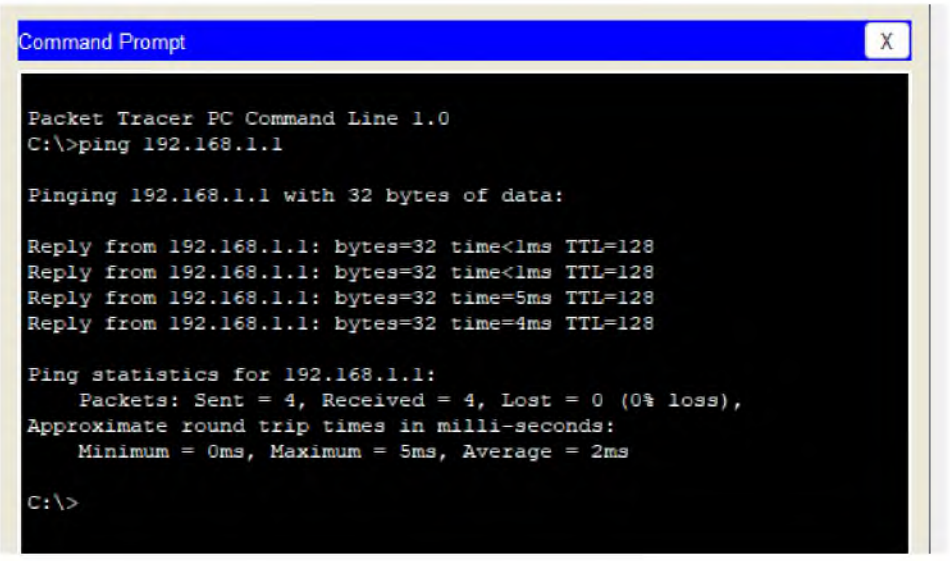


Рисунок 4.6 – Пинг до ПК из командной строки

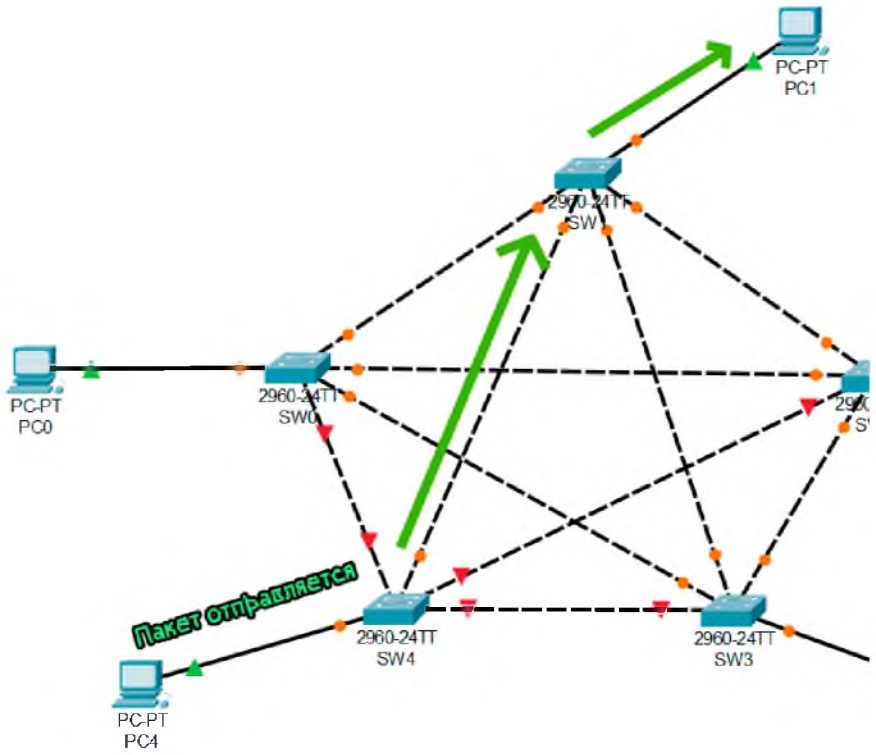


Рисунок 4.7 – Команда ping с PC4 на PC1

Отключите все порты корневого коммутатора и проверьте какой коммутатор стал корневым. Для этого определите корневой коммутатор по пунктам 4.3 – 4.4. Далее отключите рабочие порты используя команды (рисунок 4.8):

int fa0/1
shutdown

Используя команду ***show spanning-tree***, определите новый корневой коммутатор (рисунки 4.9 – 4.10).

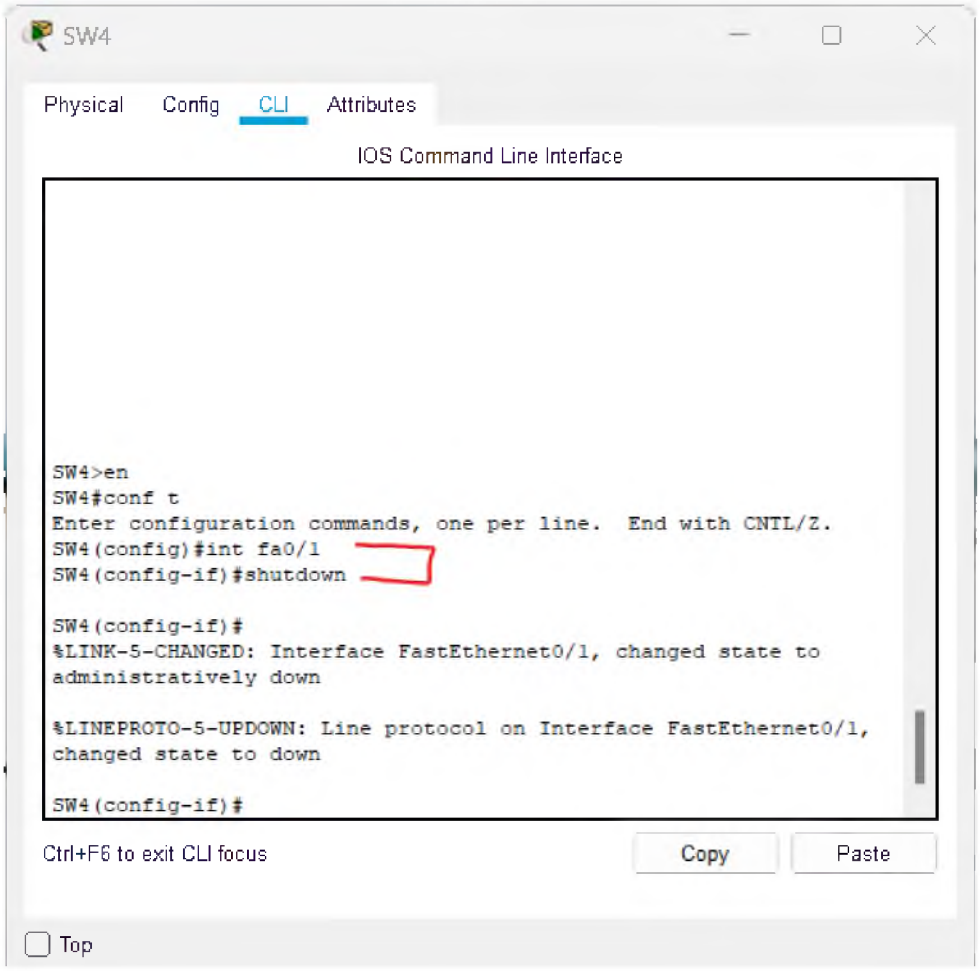


Рисунок 4.8 – Отключение портов на корневом коммутаторе

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     000C.CF10.E21C
              Cost        19
              Port        2(FastEthernet0/2)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15
sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     00D0.5854.6E65
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15
sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- --------
--------------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/2            Root FWD 19        128.2    P2p
Fa0/4            Desg FWD 19        128.4    P2p
Fa0/3            Desg FWD 19        128.3    P2p
```

Рисунок 4.9 – Определение корневого коммутатора

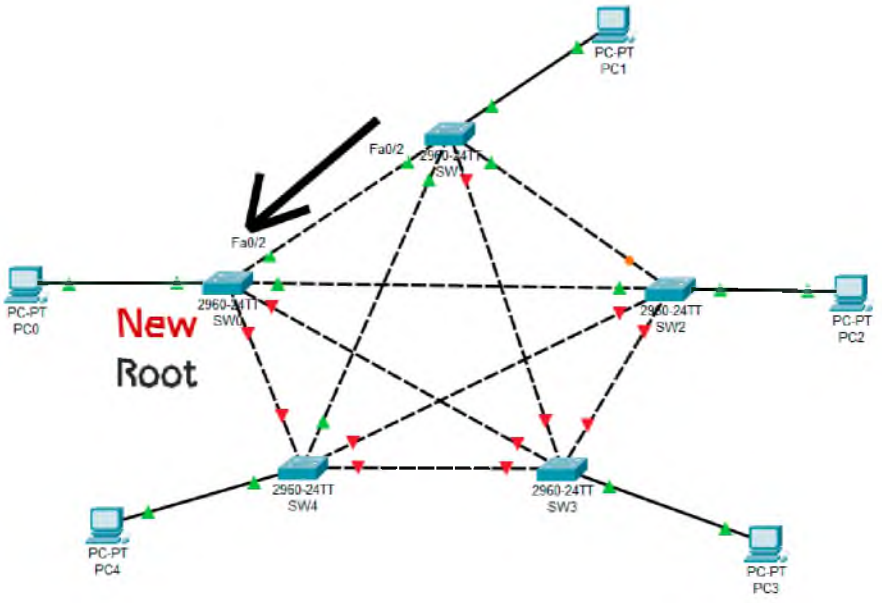


Рисунок 4.10 – Изменённый корневой коммутатор

Для изменения корневого коммутатора требуется написать команду –

spanning-tree vlan 1 priority 0

В данном случае на коммутаторе изменится приоритет BRIDGE ID для VLAN1 после чего коммутатор станет корневым. Зайдите в режим конфигурирования SW1 и измените приоритетность для VLAN1. Данный коммутатор станет корневым в топологии сети (рисунок 4.11).

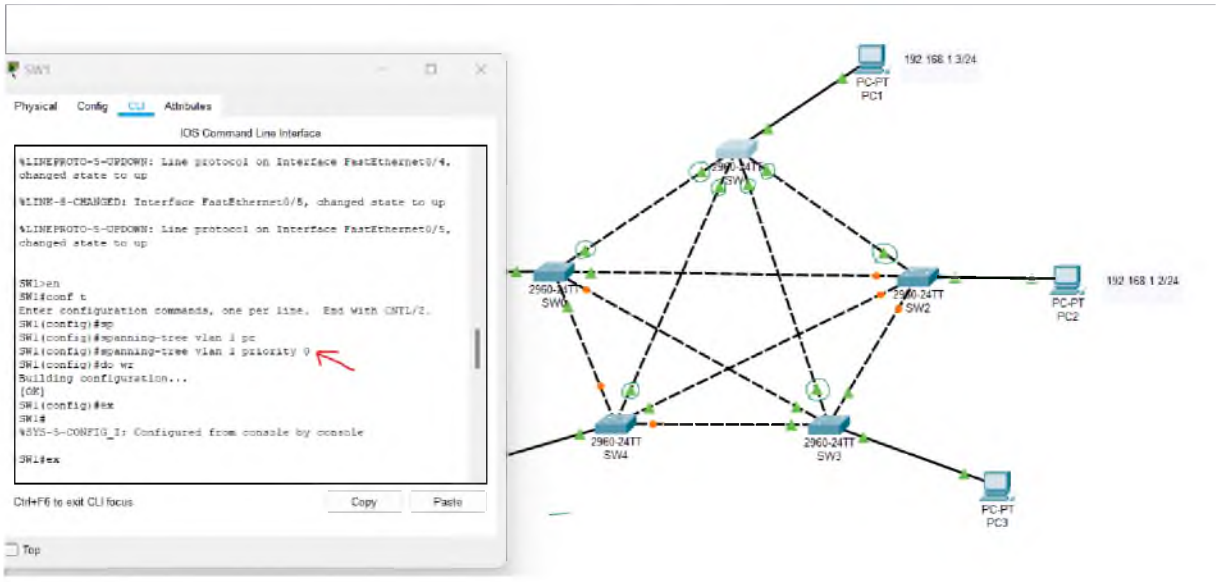


Рисунок 4.11 – Изменение корневого коммутатора

4.6 Перейти в режим симуляции, включить протоколы ARP и ICMP, посмотреть отправку пакетов между ПК и сделать вывод об отправки пакетов по маршрутам между коммутаторами (рисунок 4.12).

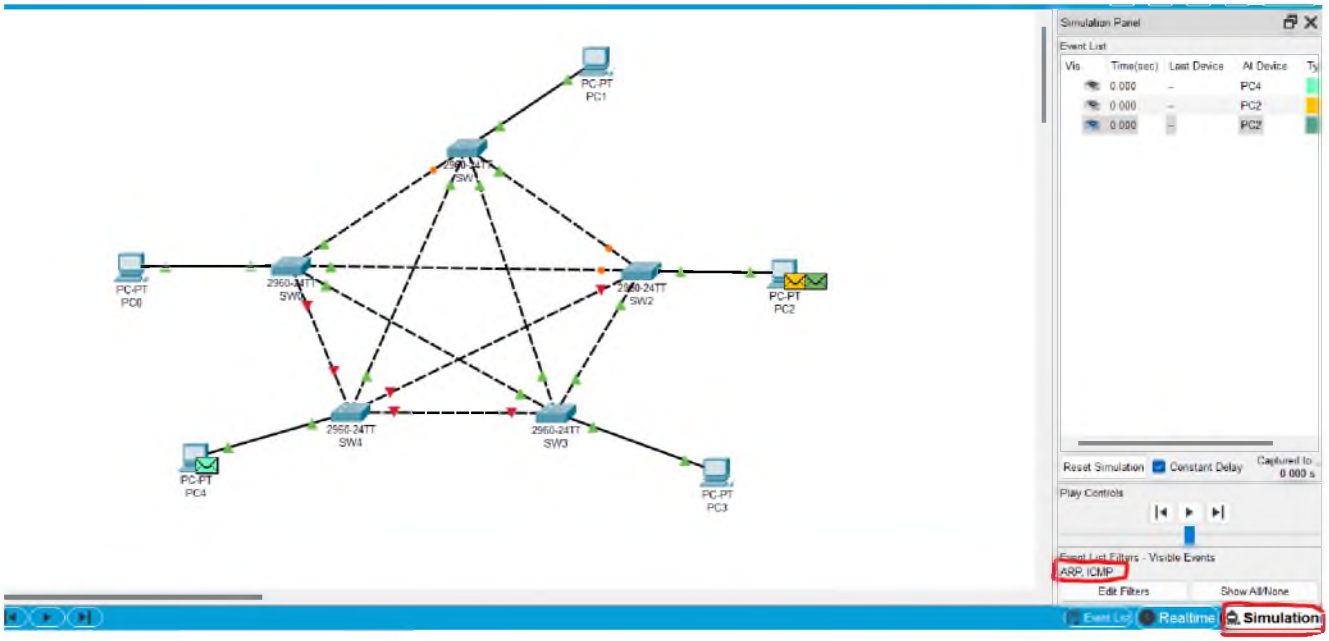


Рисунок 4.12 – Режим симуляции отправки пакетов

5. Контрольные вопросы

1. Что такое протокол связующего дерева (STP), и какова его роль в сети?
2. Что означает понятие "приоритетный коммутатор"?
3. Как связаны между собой протоколы ARP и ICMP?
4. Какую роль играет протокол STP в поддержании безопасности и стабильности сети?
5. Какие последствия могли бы возникнуть, если бы коммутаторы в сети не поддерживали протокол STP?

Лабораторная работа № 9

Тема: Распределение сетей на сетевом уровне

1. Цель: закрепить теоретические знания и получить практические навыки по настройке сети с разделением на сетевом уровне.

Знать:

- общее устройство и принцип работы маршрутизатора;
- порядок распределения IP-сети на подсети.

Уметь:

- работать в Cisco Packet Tracer;
- строить локальную сеть в сетевом симуляторе.

2. Подготовка к работе:

- повторить материал по распределению сети на подсети и решить эту задачу для сети А (таблица 9.1). Сеть А необходимо разделить на четыре подсети, с учетом минимального необходимого количества адресов для каждой сети.

3. Задание:

3.1 Построить сеть в сетевом симуляторе (рис. 9.1) используя данные, рассчитанные для сети А. Параметры для сетей В и С взять из таблицы 2.1.

3.1.1 Построить сеть А (рис 9.2). Прописать на компьютерах IP-адрес и маску.

3.1.2 Добавить к схеме два маршрутизатора (рис. 9.1) и оборудование сети С. Соединить оборудование сети между собой. Настроить IP-адресацию исходя из своего варианта IP-адреса всех компьютеров и портов маршрутизатора.

3.2 Настроить на маршрутизаторах статические маршруты из сети А в сеть С и обратно.

3.3 Проверить соединение из всех подсетей сети А к сети С (с использованием команды ping).

3.4 Проверить доступ из сети С ко всем четырем подсетям сети А.

3.5 Проверить передачу данных при помощи команды ping между компьютерами, находящимся в разных подсетях сети А.

3.6 На маршрутизаторе R1 настроить списки доступа таким образом, чтобы подсети сети А, не смогли передавать данные между собой, но, чтобы имели возможность выходить за пределы сети А в сеть С.

3.7 На маршрутизаторе R2 настроить списки доступа таким образом, чтобы сеть C, не могла передавать пакеты сети A.

3.8 Отразить в отчёте пошаговый алгоритм создания списков доступа и включение этого списка на выбранном интерфейсе.

3.9 Подробно описать принцип работы

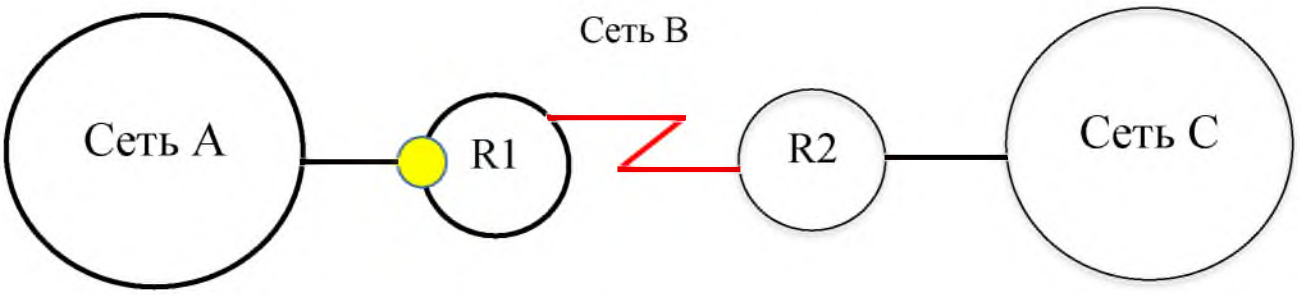


Рисунок 9.1 – Схема сети

Таблица 9.1 – Таблица вариантов

№	IP-адрес сети A	Количество узлов в каждой подсети сети A	IP-адрес сети B	IP-адрес сети C
1	10.1.0.0	7 узлов; 15 узлов; 53 узла; 285 узлов	192.168.1.0/29	172.16.1.0/25
2	10.2.0.0	22 узла; 63 узла; 150 узлов; 257 узлов	192.168.2.0/30	172.16.2.0/26
3	10.3.0.0	15 узлов; 105 узлов; 153 узла; 258 узлов	192.168.3.0/29	172.16.3.0/27
4	10.4.0.0	61 узел; 113 узлов; 182 узла; 259 узлов	192.168.4.0/28	172.16.4.0/28
5	10.5.0.0	31 узел; 297 узлов; 130 узлов; 13 узлов	192.168.5.0/27	172.16.5.0/29
6	10.6.0.0	42 узла; 88 узлов; 264 узла; 31 узел	192.168.6.0/26	172.16.6.0/28
7	10.7.0.0	12 узлов; 15 узлов; 212 узлов; 274 узла;	192.168.7.0/25	172.16.7.0/27
8	10.8.0.0	15 узлов; 29 узлов; 103 узла; 265 узлов	192.168.8.0/24	172.16.8.0/26
9	10.9.0.0	7 узлов; 259 узлов; 33 узла; 92 узла	192.168.9.0/27	172.16.9.0/25
10	10.10.0.0	6 узлов; 21 узел; 127 узлов; 261 узел;	192.168.10.0/28	172.16.10.0/26
11	10.11.0.0	30 узлов; 59 узлов; 63 узла; 334 узла	192.168.11.0/25	172.16.11.0/27
12	10.12.0.0	15 узлов; 126 узлов; 42 узла; 257узлов	192.168.12.0/26	172.16.12.0/28
13	10.13.0.0	7 узлов; 48 узлов; 265 узлов; 20 узлов	192.168.13.0/27	172.16.13.0/29

14	10.14.0.0	13 узлов; 29 узлов; 63 узла; 264 узла	192.168.14.0/25	172.16.14.0/28
15	10.15.0.0	11 узлов; 25 узлов; 262 узла; 127 узлов	192.168.15.0/28	172.16.15.0/27
16	10.16.0.0	5 узлов; 23 узла; 61 узел; 300 узлов	192.168.16.0/29	172.16.16.0/26
17	10.17.0.0	10 узлов; 2 узла; 31 узел; 258 узлов	192.168.17.0/30	172.16.17.0/25
18	10.18.0.0	4 узла; 20 узлов; 260 узлов; 127 узлов	192.168.18.0/24	172.16.18.0/26
19	10.19.0.0	18 узлов; 31 узел; 147 узлов; 270 узлов	192.168.19.0/25	172.16.19.0/27
20	10.20.0.0	17 узлов; 31 узел; 261 узел; 117 узлов	192.168.20.0/26	172.16.20.0/28
21	10.21.0.0	6 узлов; 14 узлов; 63 узла; 298 узлов	192.168.21.0/24	172.16.21.0/29
22	10.22.0.0	33 узла; 11 узлов; 63 узла; 310 узлов	192.168.22.0/25	172.16.22.0/28
23	10.23.0.0	10 узлов; 30 узлов; 257 узлов; 127 узлов	192.168.23.0/26	172.16.23.0/27
24	10.24.0.0	12 узлов; 288 узлов; 15 узлов; 113 узлов	192.168.24.0/27	172.16.24.0/26
25	10.25.0.0	22 узла; 28 узлов; 63 узла; 276 узлов	192.168.25.0/28	172.16.25.0/25
26	10.26.0.0	13 узлов; 19 узлов; 63 узла; 268 узлов	192.168.26.0/29	172.16.26.0/26
27	10.27.0.0	88 узлов; 15 узлов; 261 узел; 53 узла	192.168.27.0/30	172.16.27.0/27
28	10.28.0.0	29 узлов; 17 узлов; 31 узел; 259 узлов	192.168.28.0/26	172.16.28.0/28
29	10.29.0.0	15 узлов; 11 узлов; 271 узел; 124 узла	192.168.29.0/27	172.16.29.0/29
30	10.30.0.0	31 узел; 29 узлов; 264 узла; 115 узлов	192.168.30.0/25	172.16.30.0/28

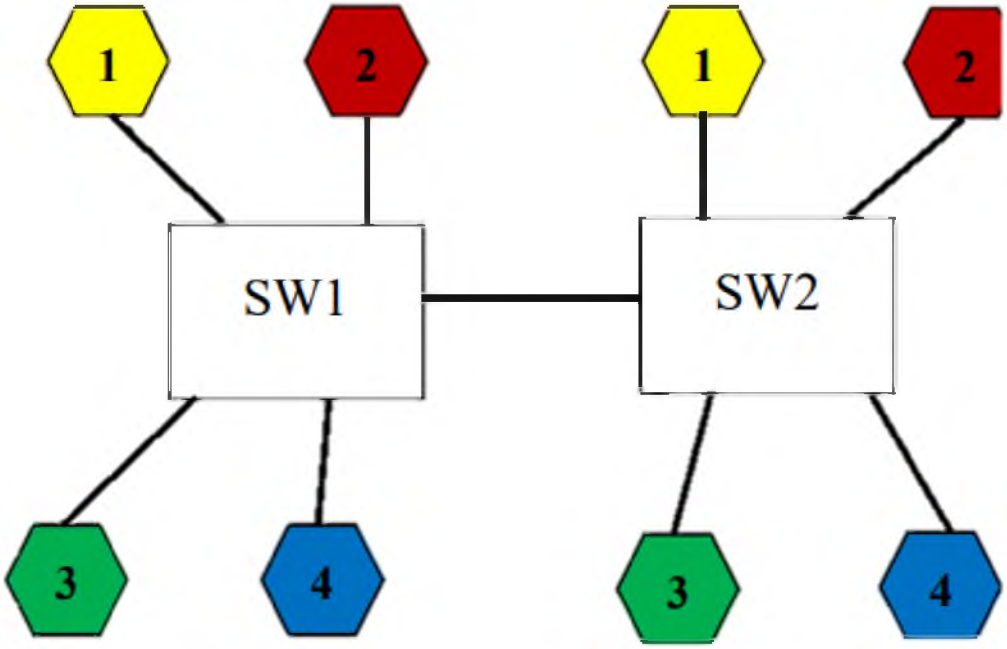


Рисунок 9.2 – Схема сети А. Цифрами 1, 2, 3 и 4 обозначены номера IP-сетей.

4. Порядок выполнения

4.1.1 Для начала создайте топологию сети А (рисунок 4.1).

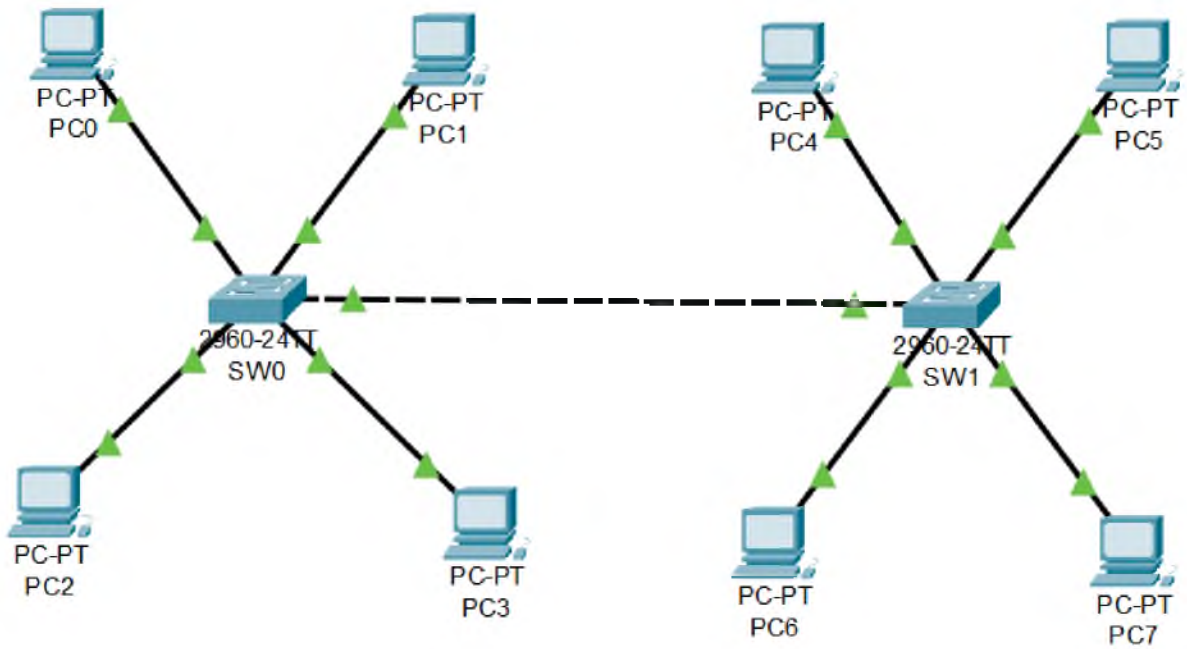
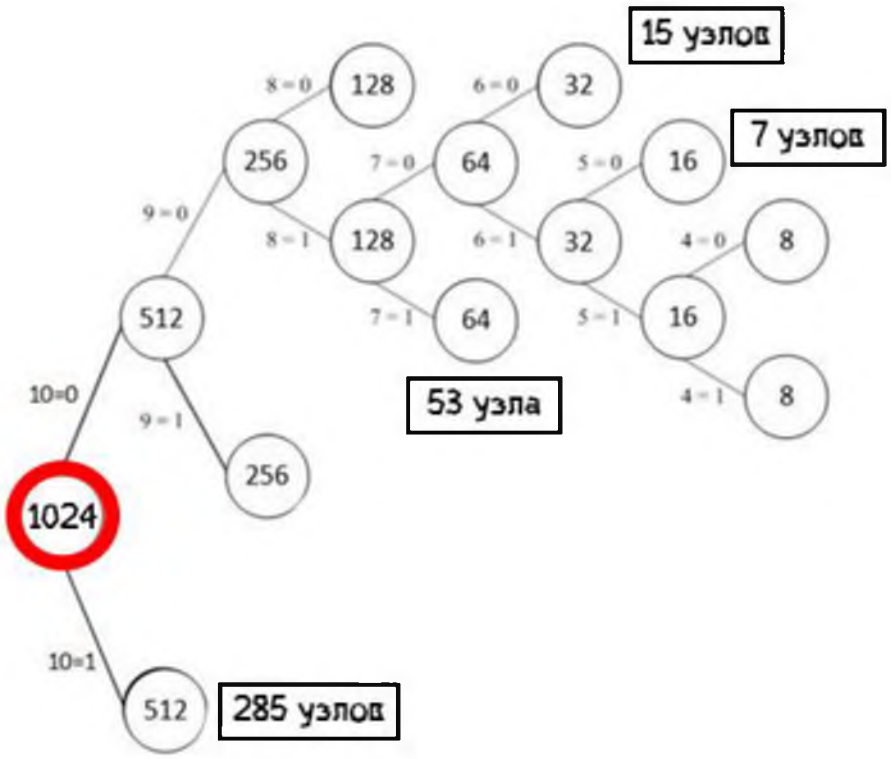


Рисунок 4.1 – Топология сети А

Для расчёта IP-адресов воспользуйтесь ресурсом:

https://dzen.ru/a/Y1Zn8UAXPV9VuxPY.



Пример. Пусть дана сеть 10.0.0.0. Необходимо разбить данную сеть на подсети. В каждой подсети планируется использовать следующее количество узлов:

1 сеть – 7 узлов;

2 сеть – 15 узлов;

3 сеть – 53 узла;

4 сеть – 285 узлов;

Необходимо понимать, что в данном случае такие понятия как сеть и подсеть идентичны. То есть IP-подсеть обладает всеми параметрами сети и ее можно также рассматривать как отдельную сеть.

Ход решения.

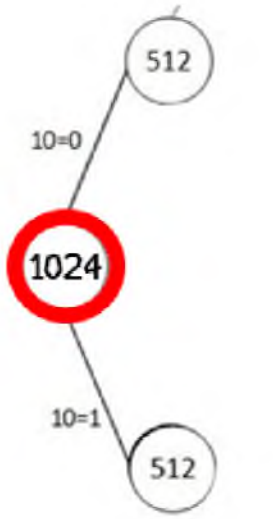
Переведем заданный номер сети в двоичный код и определимся с номерами бит. Предлагаю нумеровать биты в IP-адресе справа налево, чтобы работать с меньшими цифрами (Можно нумеровать и слева направо, ошибки не будет).

Для решения задачи необходимо построить дерево IP-адресов. Оно строится произвольным образом, с соблюдением определенных правил. Следует учитывать, что используется двоичный код, поэтому все числа являются

степенями двойки (то есть 2, 4, 8, 16, 32 и так далее). Для начала можно сложить все то количество узлов, для которых необходимо составить IP-адресацию.

7 + 15 + 53 + 285 = 360

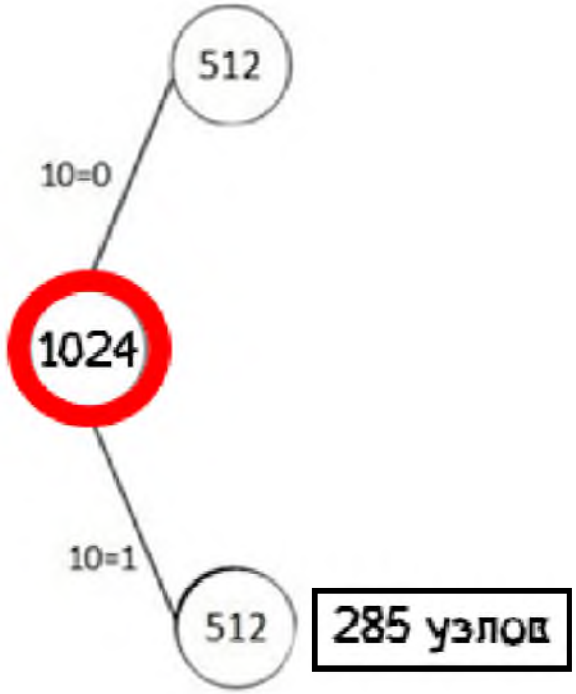
Ближайшее большее число степени двойки это 1024 (два в десятой степени - 2^10). Это число можно поделить на два значения по 512 каждый (2^9). При этом десятый бит будет принимать значения либо ноль, либо один. Нарисуем получившееся начало дерева. Не имеет значения какой бит (верхний или нижний) примет нулевое значение. Желательно при создании дерева определится, чтобы было одинаково, например, все верхние биты равны нулю, а все нижние единицы (как в данном рисунке).



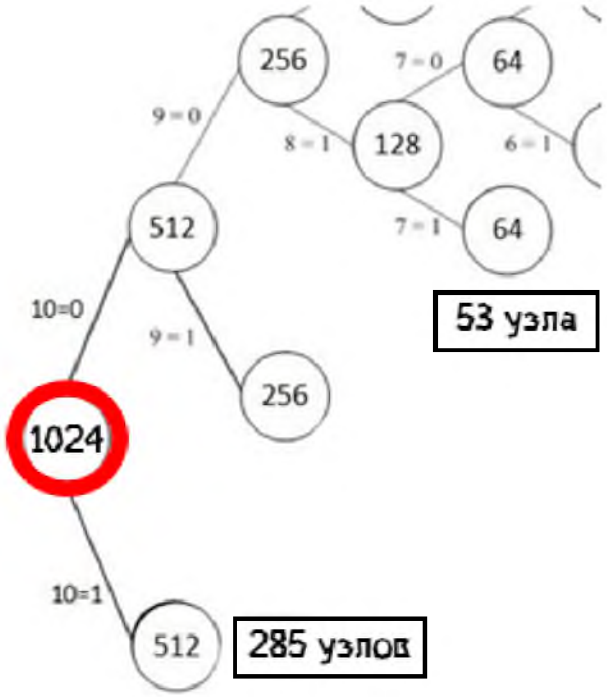
Таким образом продолжаем делить каждую ячейку до необходимых значений. Здесь нужно пояснить, что пытаемся решить задачу с минимально необходимым выделением адресов. То есть, если нужно выделить сеть на 52 узла, например, то нужно выбрать ячейку на 64 адреса. Можно выбрать и 128 адресов, но тогда будет избыточное значение незадействованных адресов. Также нужно понимать, что в любой сети (большой или маленькой) два IP-адреса являются служебными и их нельзя назначать в качестве IP-адресов узлов. Это самый первый адрес, который является номером сети и самый последний адрес – широковещательный адрес сети. То есть если нужно выделить сеть, например, для 63 узлов, то ячейки на 64 адреса будет недостаточно, нужно выбирать ячейку на 128 адресов.

Также необходимо по мере деления адресного пространства назначать определенные ячейки в качестве сетей. Самое большое количество адресов требуется для четвёртой сети (285 адресов), для этой сети необходимо выделить

ячейку в 512 адресов. Выберем для этого нижнюю ячейку (хотя можно и нижнюю, не имеет значение).



Таким образом нижняя ячейка получается полностью задействована для адресации четвёртой сети в 285 адресов. То, что часть адресов останется свободной не имеет значение. Их все равно уже нельзя использовать в других сетях. Далее делим только верхнюю ячейку в 512 адресов, так как она осталась свободной. Не забываем на линии перехода подписывать номера бит и их значения. Соответственно между 512 и 256 уже будет находится девятый бит. Для третьей сети нужно 53 адреса, ближайшая наибольшая ячейка это 64. Выберем нижнюю для адресации третьей сети.



Продолжаем делить оставшуюся часть адресного пространства по тому же принципу, что и раньше. Подписываем номера бит и выделяем адресное пространство для остальных подсетей.

Теперь необходимо исходя из построенного дерева определить адресные параметры для каждой сети: номер сети, префикс и маску сети, широковещательный адрес сети. А также диапазон и количество адресов, которые будут назначаться в качестве IP-адресов узлов.

Алгоритм решения можно представить следующим образом:

1. Определение границы сети.
2. Определение номера сети.
3. Определение маски (префикса) сети.
4. Определение широковещательного адреса сети.
5. Определение диапазона и количества адресов.

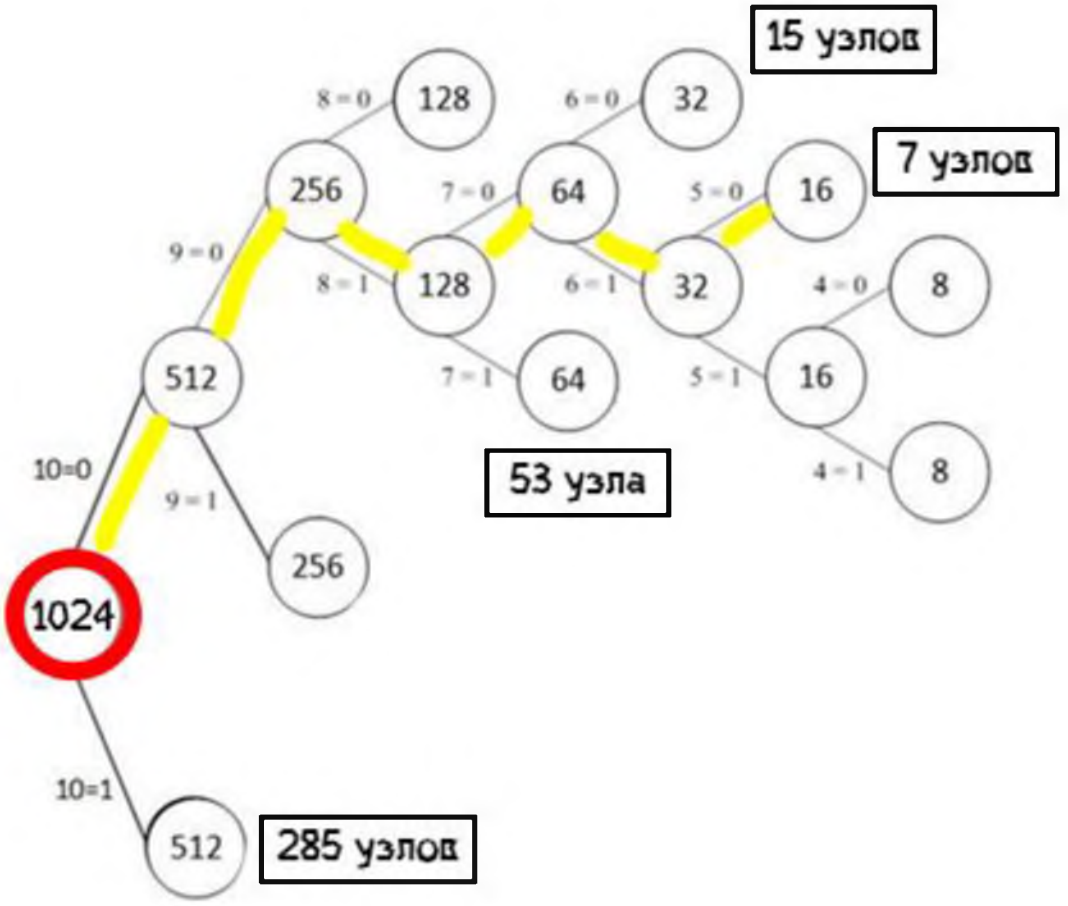
Рассмотрим на примере первой сети для семи узлов. Для этой сети выбрано адресное пространство на 16 узлов.

1. Определение границы сети. Речь идет об определении границы в IP-адресе между номером сети и номером узла (хоста). Для этого необходимо найти по дереву IP-адресов (рисунок №2) номер бита, который является последним для данной подсети (это четвертый бит), соответственно граница проходит между пятым и нижележащим (четвёртым) битом.

2. Определение номера сети. Для этого значения бит справа от границы сети заполняем нулями (это биты 4, 3, 2 и 1). Для заполнения значения бит слева от границы используем дерево IP-адресов (рисунок №2). Нужно передвигаться от выбранной ячейки (16 адресов) по ветвям дерева к вершине и записывать значения бит. Четвертый бит равен нулю, пятый бит – единице, шестой – единице, седьмой – нулю и так далее. До одиннадцатого бита, который равен единице. Когда дерево закончилось, то значения бит необходимо переписать исходя из первоначально заданной сети (рисунок 1). Получаем следующую запись:

Теперь переводим номер подсети в десятичную систему: 10.0.0.160.

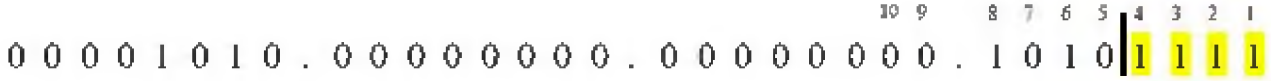
Перемещение от ячейки в 16 адресов до вершины дерева представлено на следующем рисунке



3. Определение маски (префикса) сети. Маска – второй параметр сети, поэтому его записываем отдельно. Для определения маски необходимо все биты, находящиеся слева от границы, заполнить единицами, а справа – нулями:

Переводим маску подсети в десятичную систему: 255.255.255.240. Префикс — это еще один вид записи маски, он указывает количество бит в IP-адресе отведенных под номер сети. В данном случае это количество единиц в двоичном коде маске (то есть префикс равен 28).

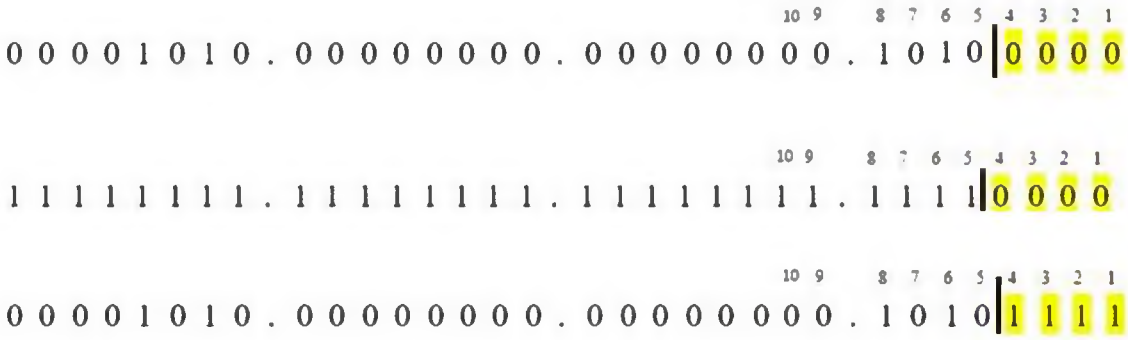
4. Определение широковещательного адреса сети. Широковещательный адрес сети – это самый последний адрес в данной сети. Для его определения переписываем биты относящиеся к адресу сети (рисунок 3) без изменения, а биты, относящиеся к адресу узла (справа от границы сети), заменяем на единицы:



Переводим широковещательный адрес в десятичную систему счисления: 10.0.0.175.

5. Определение диапазона и количества адресов. Для того чтобы вычислить адрес первого узла в подсети, необходимо к номеру подсети прибавить единицу (10.0.0.160), а для того, чтобы определить адрес последнего узла, - от широковещательного узла подсети отнять единицу (10.0.0.175). Получаем следующий диапазон адресов узлов: 10.0.0.161 – 10.0.0.174. Таким образом, максимальное количество адресов в подсети 10.0.1.161 с маской 255.255.255.240 составляет 13. Данное количество адресов достаточно для того, чтобы в подсети разместить 7 узлов.

Все вышеизложенные рассуждения можно пропустить и записать все решение в компактной форме:



Номер подсети и префикс: 10.0.0.160/28

Маска подсети: 255.255.255.240

Широковещательный адрес: 10.0.0.175

Адрес первого узла: 10.0.0.161

Адрес последнего узла: 10.0.0.174

Диапазон адресов: с 10.0.0.161 до 10.0.0.174

Количество IP-адресов для адресации узлов: 13

Еще раз обращаю внимание, что понятие сеть и подсеть почти одинаковы, так как каждая из них имеет полный набор параметров. В данном случае рассматривается деление большой сети на несколько более мелких (по количеству узлов в сети).

Для определения IP-адресации подсетей с 25, 62 и 120 узлами повторяем все те же действия, что описаны выше.

В примере рассмотрен IP-адрес сети А 10.1.0.0 на 7, 15, 53, 285 узлов. Для этого воспользовавшись информацией выше, получим следующие значения:

7 узлов – 255.255.255.240 (/28)

15 узлов – 255.255.255.224 (/27)

53 узла – 255.255.255.192 (/26)

285 узлов – 255.255.254.0 (/23)

Присвойте полученные IP-адреса компьютерам в соответствии с рисунком 2.1.

Пример схемы изображён на рисунке 4.2.

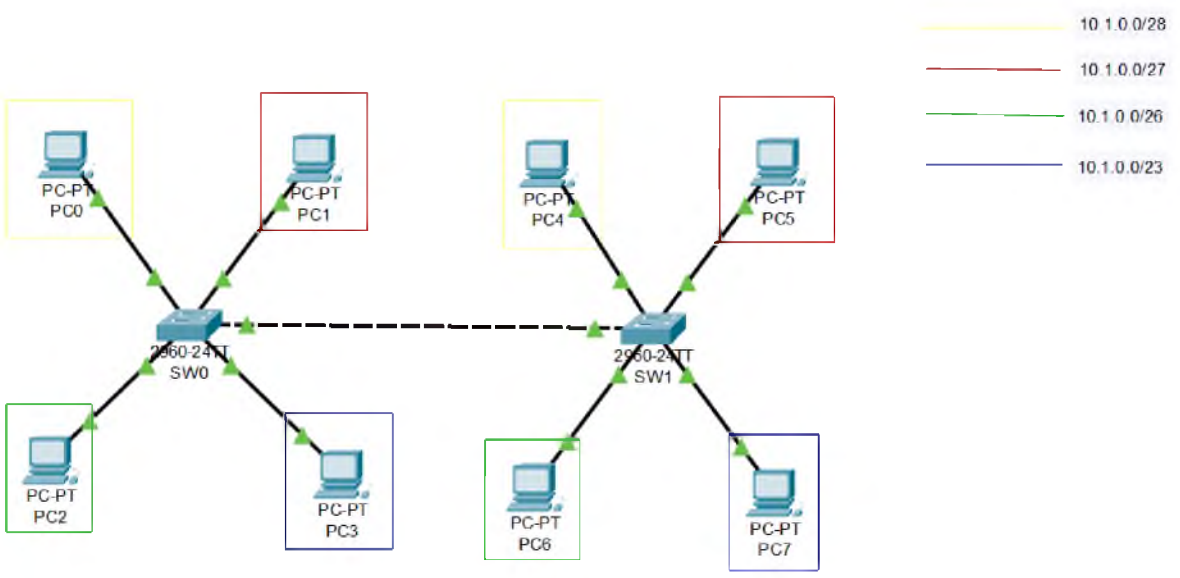


Рисунок 4.2 – Топология сети с пометками подсетей

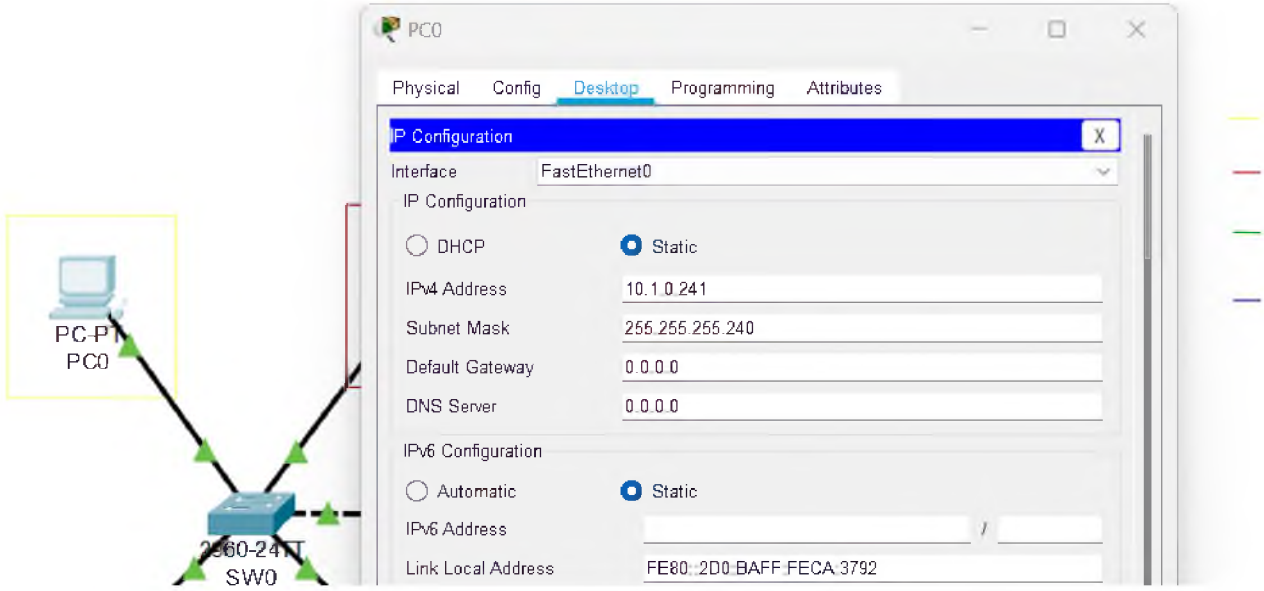


Рисунок 4.3 – Пример заполнения IP-адреса для PC0

4.1.2 Добавьте на схему два маршрутизатора 1941 и добавьте порты HWIC-2T для соединения точка-точка (рисунок 4.4).

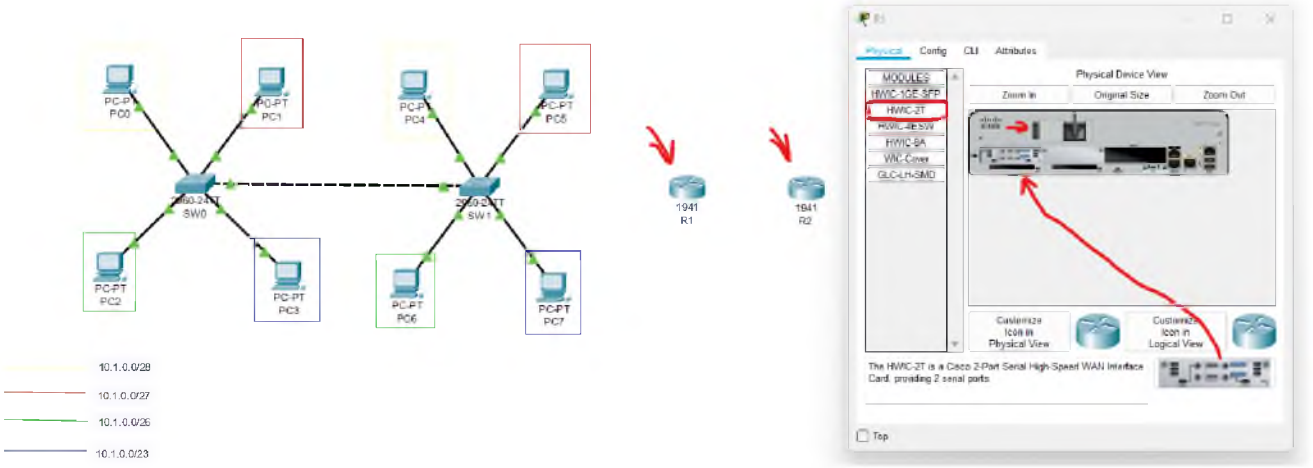


Рисунок 4.4 – Добавление портов HWIC-2T для маршрутизаторов Cisco 1941

После добавления новых портов маршрутизаторам достройте топологию сети (рисунок 4.5).

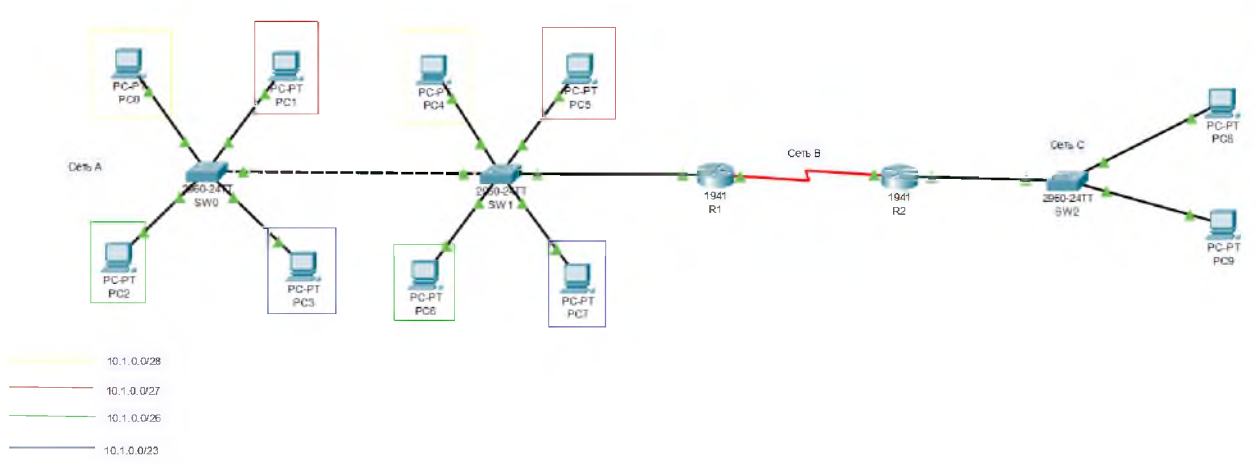


Рисунок 4.5 – Полная топология сети

Далее добавьте IP-адресацию сети B и C (рисунки 4.6 – 4.7).

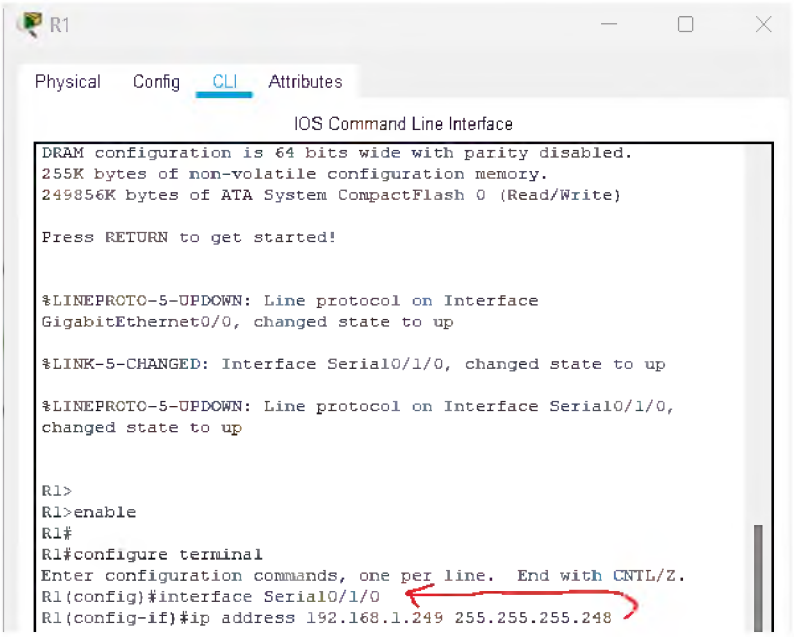


Рисунок 4.6 – Настройка IP-адреса маршрутизатора R1 (сеть B)

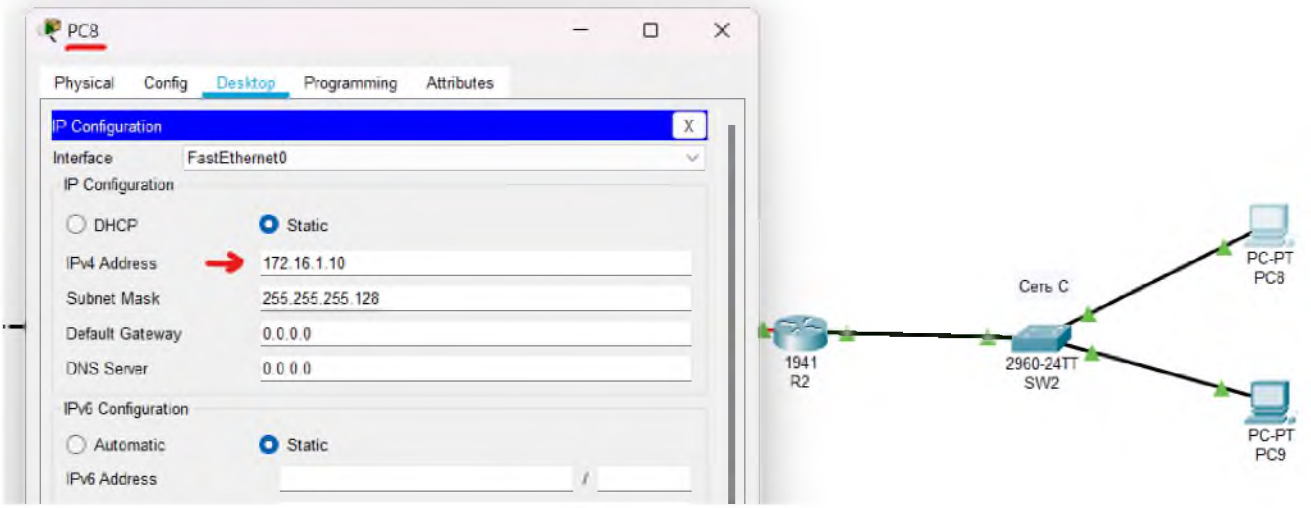


Рисунок 4.7 – Настройка IP-адреса компьютера PC8 (сеть C)

4.2 Для настройки статической маршрутизации зайдите в роутер R1 вкладка «CLI» зайдите в режим конфигурации и введите команду маршрутизации к сети C (рисунок 4.8).

ip route 172.16.1.0 255.255.255.128 192.168.1.250

Удостовериться в правильной настройке статической маршрутизации можно перейдя во вкладку «Config» группы «Routing» типа «Static» (рисунок 4.9).

Затем настройте статическую маршрутизацию на роутере R2 (рисунок 4.10). Для настройки введите команду:

ip route 10.1.0.0 255.255.254.0 192.168.1.249

Обратите внимание, что требуется написать общую сеть 10.*.0.0 с соответствующей маской.

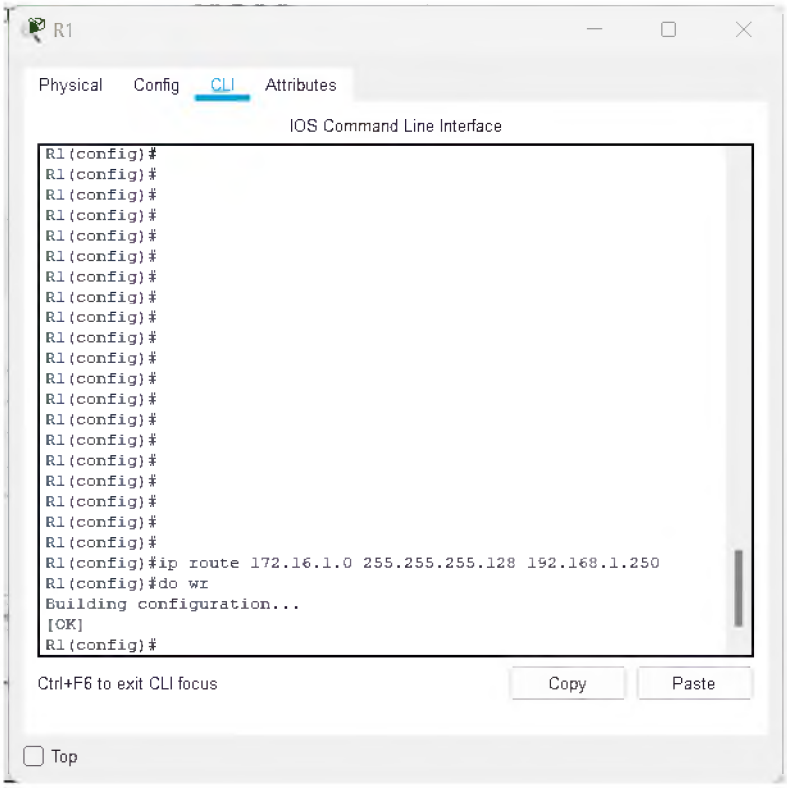


Рисунок 4.8 – Настройка статической маршрутизации из сети A в C

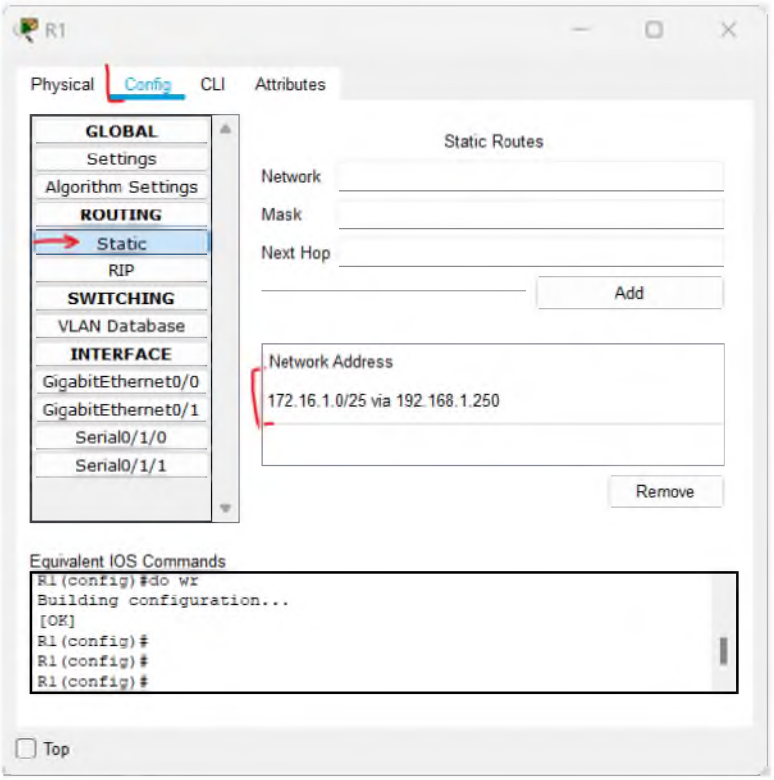


Рисунок 4.9 – Проверка настройки статической маршрутизации

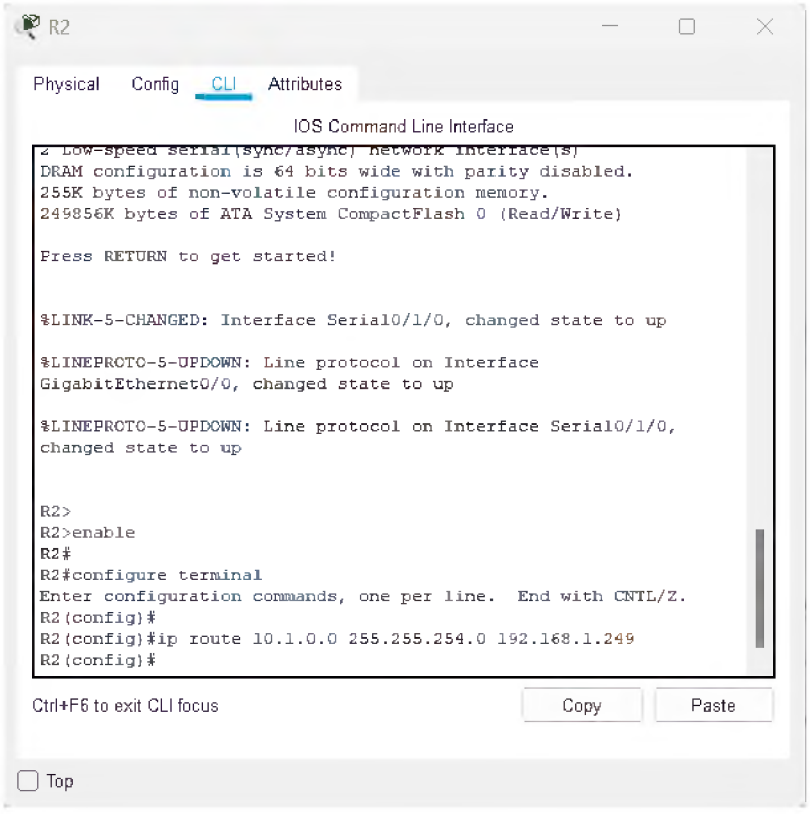
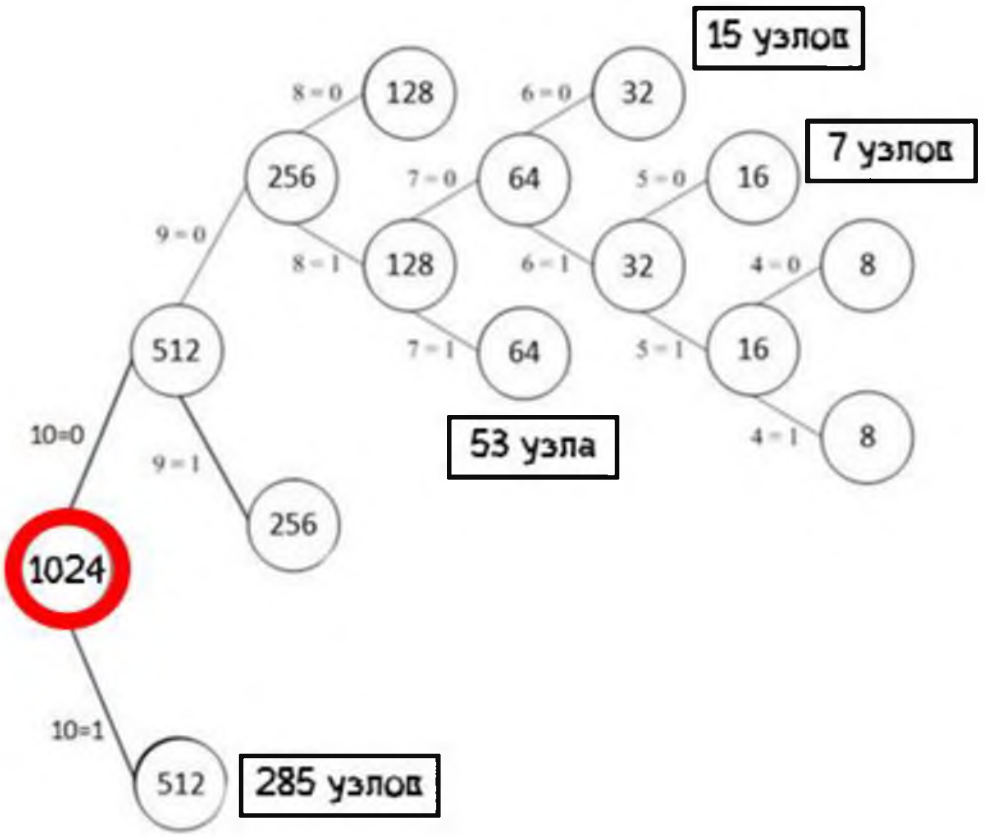


Рисунок 4.10 – Настройка статической маршрутизации из сети C в A

После настройки статической маршрутизации настройте на компьютерах шлюз роутеров. Поскольку сеть A имеет подсети требуется произвести расчёт общей сети для определения правильного IP-адреса шлюза.

Рассмотрим каждую подсеть и найдем их общую сеть.



1. Подсеть 1: 10.1.0.0/28 (Маска: 255.255.255.240)
2. Подсеть 2: 10.1.0.0/27 (Маска: 255.255.255.224)
3. Подсеть 3: 10.1.0.0/26 (Маска: 255.255.255.192)
4. Подсеть 4: 10.1.0.0/23 (Маска: 255.255.254.0)

В данном случае, вы видите, что первые три октета в каждой подсети также одинаковы (10.1.0.X). Давайте найдем общую сеть, которая включает в себя все эти подсети.

Сначала, определите общую маску для всех четырех подсетей:

1. Подсеть 1: 1111 1111.1111 1111.1111 1111.1111 0000 (28 бит)
2. Подсеть 2: 1111 1111.1111 1111.1111 1111.1110 0000 (27 бит)
3. Подсеть 3: 1111 1111.1111 1111.1111 1111.1100 0000 (26 бит)
4. Подсеть 4: 1111 1111.1111 1111.1111 1111.1111 1000 (23 бит)

Общая маска, которая включает в себя все эти подсети, будет содержать первые 23 бита, потому что это минимальная маска, которая покрывает все эти подсети.

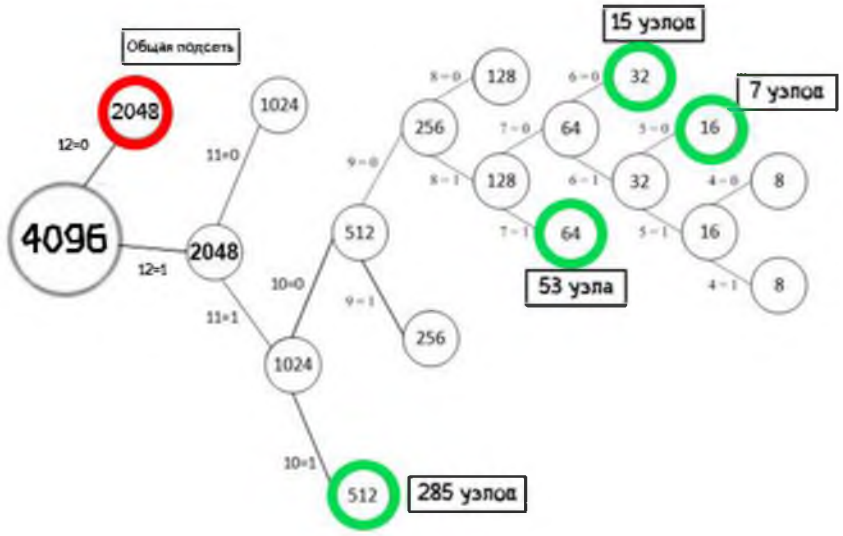
Но для правильного расчёта нам следует учесть, что маска /23 уже используется четвёртой подсетью, поэтому мы должны выбрать маску /22.

Маска для общей сети: 255.255.252.0 (/22)

Теперь, когда у нас есть маска /22 (255.255.252.0), вы можете выбрать любой доступный IP-адрес в этой сети, который не присутствует в одной из наших подсетей. Например, вы можете взять 10.1.1.1.

Таким образом, IP-адрес и маска шлюза для объединенной сети будут:

- IP-адрес: 10.1.1.1
- Маска шлюза: 255.255.252.0 (/22)



После определения шлюза, запишите IP-адрес в порт, подключённый к коммутатору сети A (рисунок 4.11). Настройте основной шлюз для компьютеров PC0-PC3 (рисунок 4.12). **Аналогичные** действия для настройки шлюза произведите с маршрутизатором R2 и компьютерами PC8, PC9 сети C (рисунок 4.13).

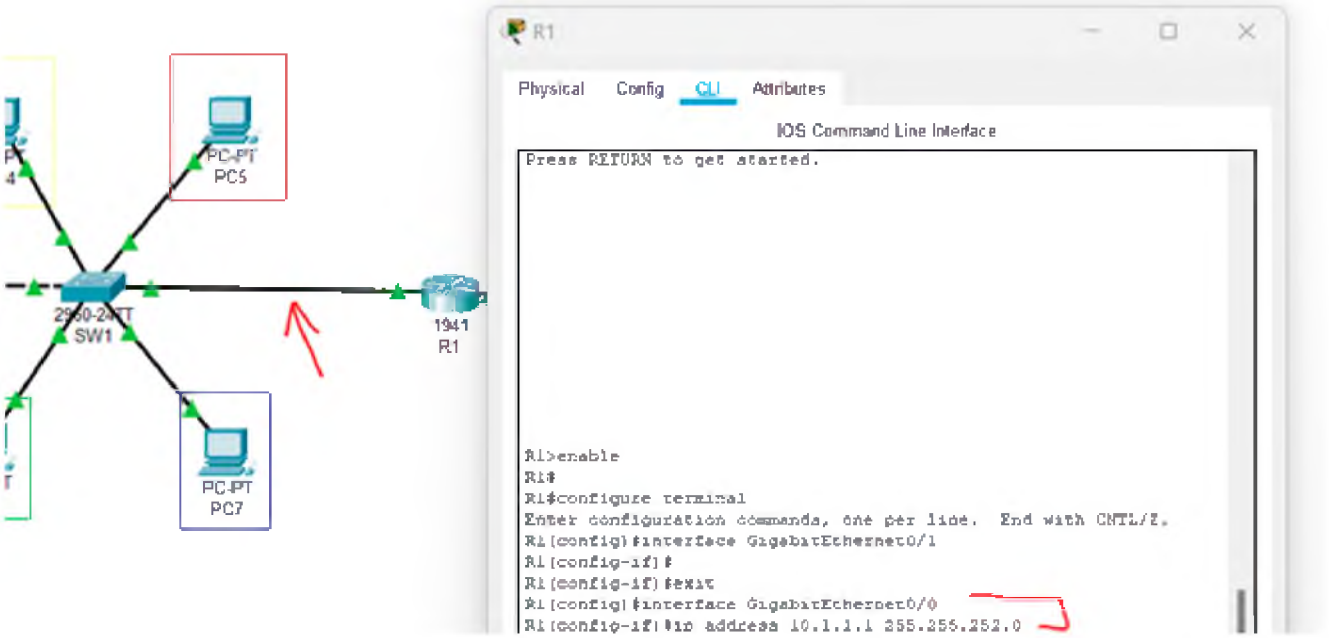


Рисунок 4.11 – Настройка шлюза на маршрутизаторе R1

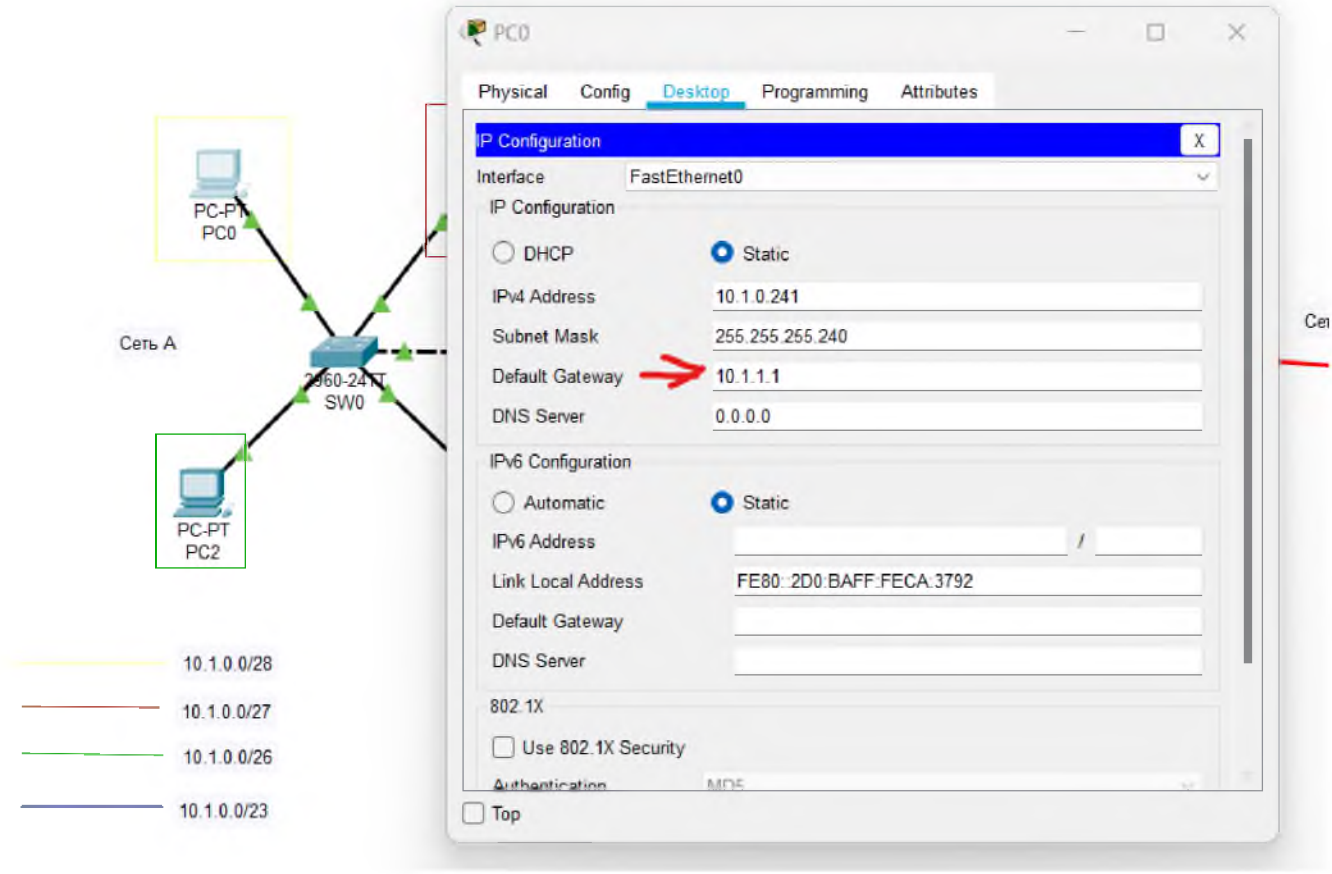


Рисунок 4.12 – Настройка шлюза на компьютере PC0

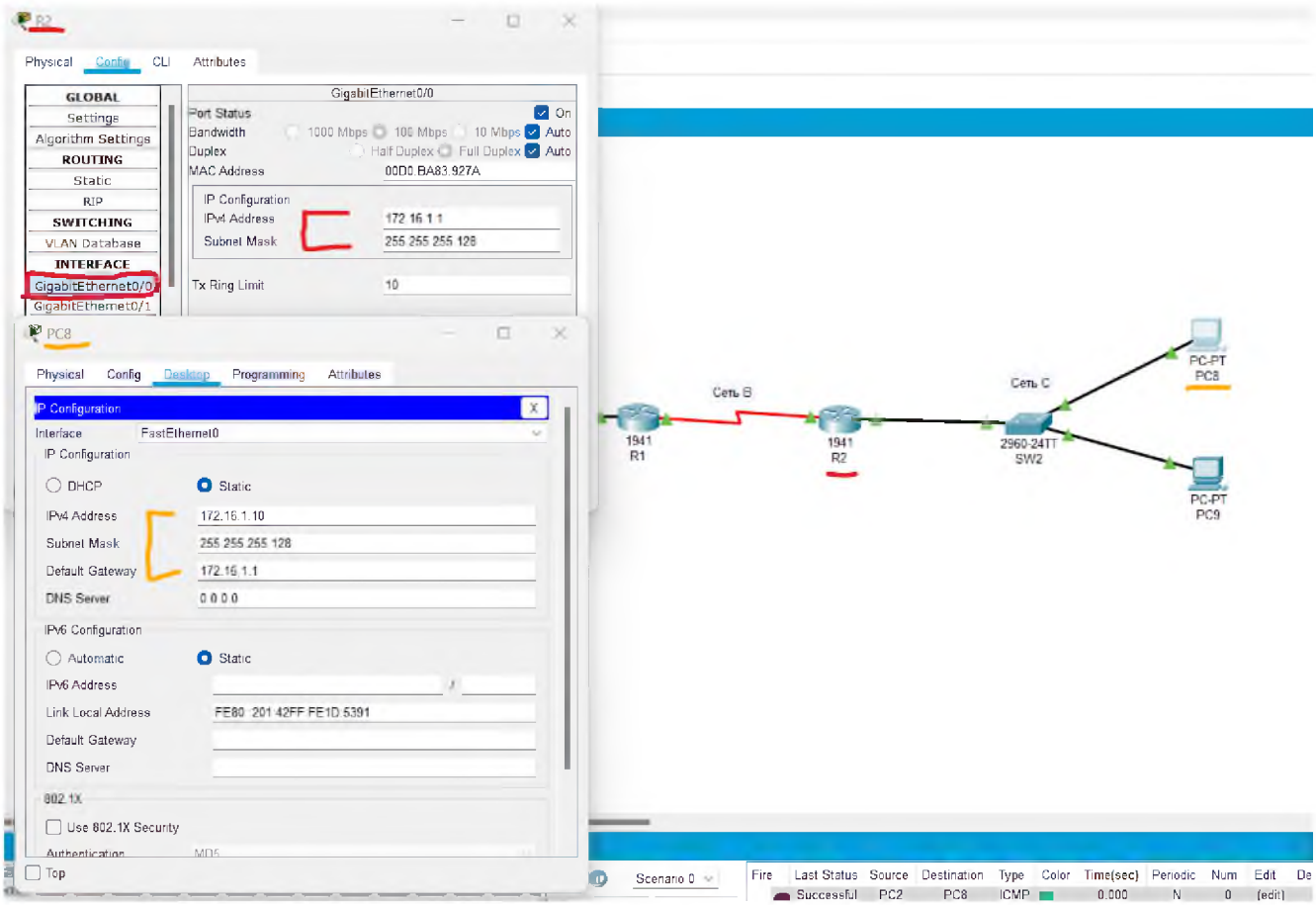


Рисунок 4.13 – Настройка шлюза на маршрутизаторе R2 и компьютере PC8

4.3 Для проверки соединения из всех подсетей сети А к сети С требуется зайти в режим симуляции и с каждого компьютера в командной строке написать команду ***ping * IP-адрес сети С ****. Пример отправки пакетов с компьютера PC2 к компьютеру PC8 представлен на рисунке 4.14. Правильная передача пакетов показана на рисунках 4.15-4.18. Для проверки доступа с других компьютеров сети А требуется произвести **аналогичные** действия.

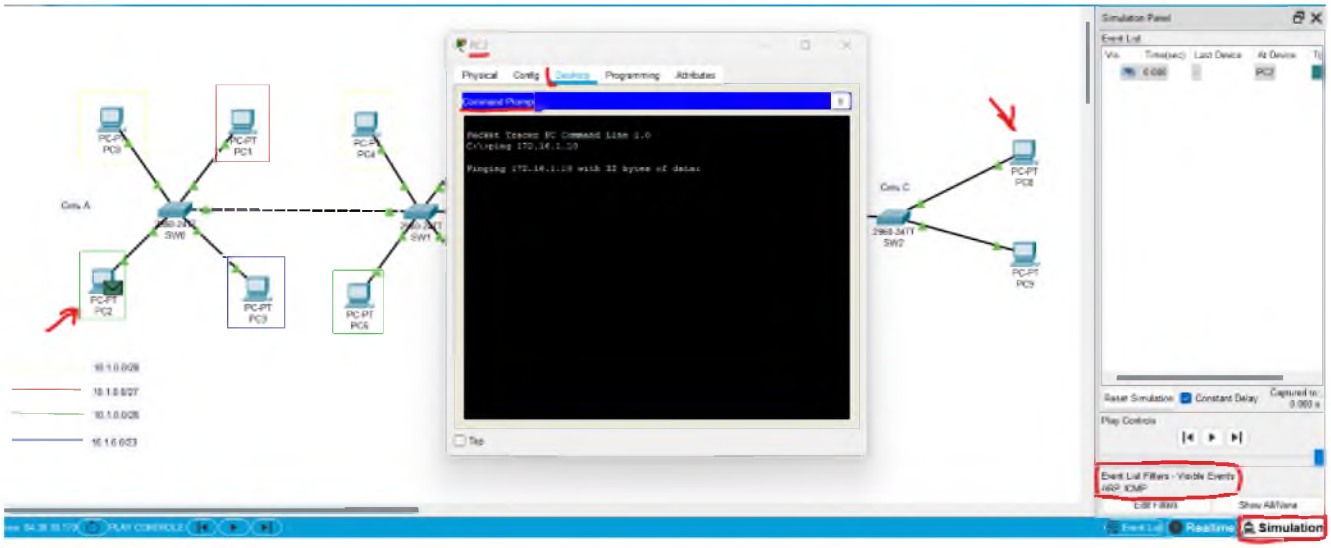


Рисунок 4.15 – Отправка пакетов командой ping с PC2 – PC8

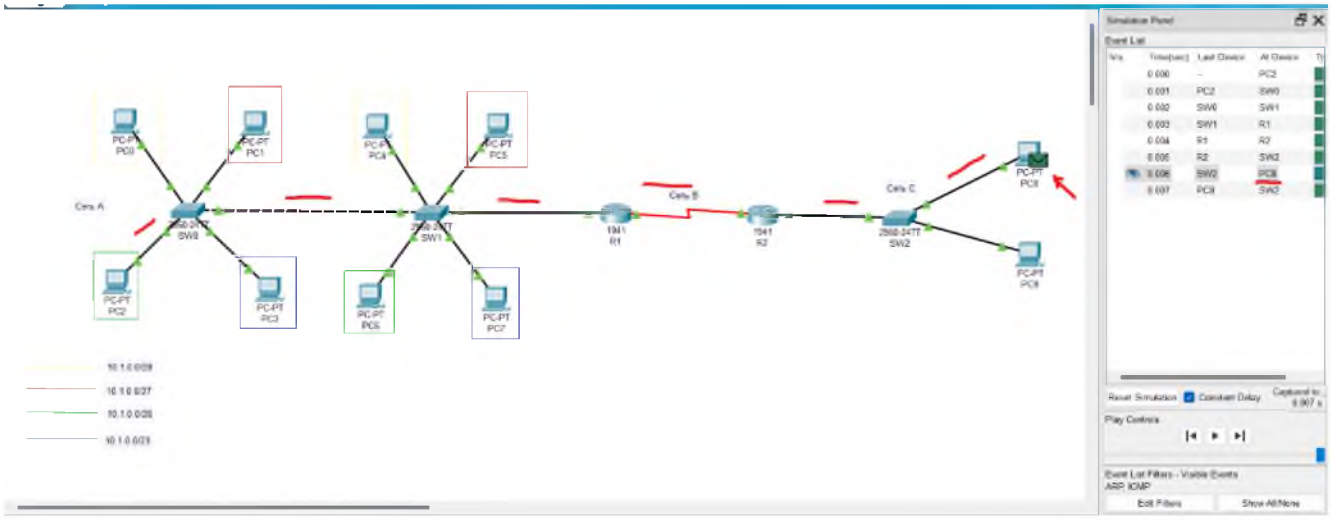


Рисунок 4.16– Путь отправки пакета с PC2 до PC8

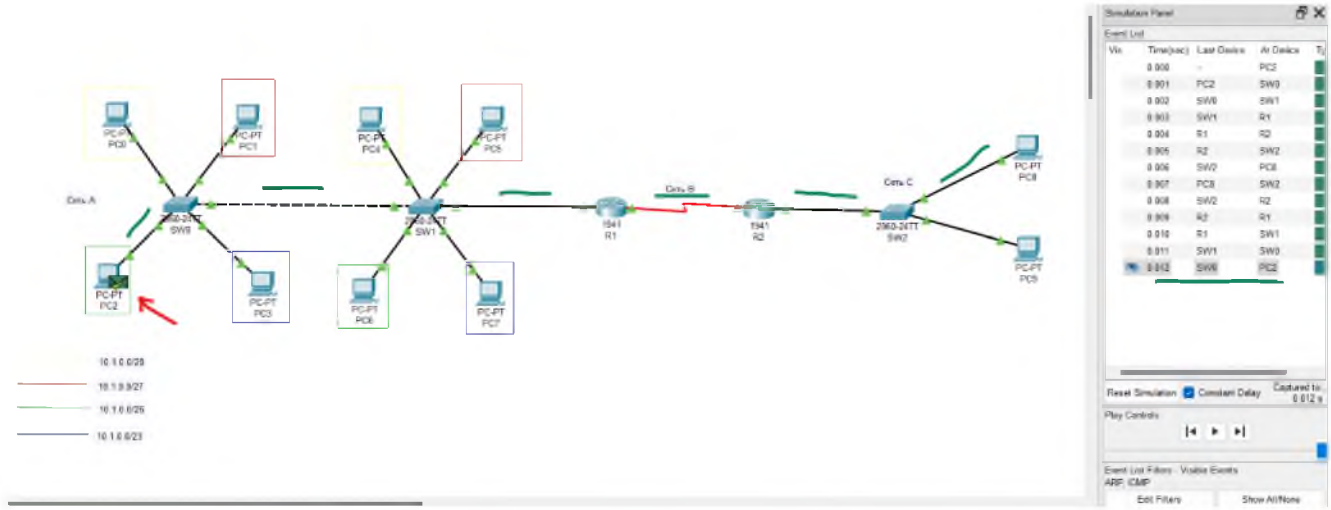


Рисунок 4.17 – Получение ответа от PC8

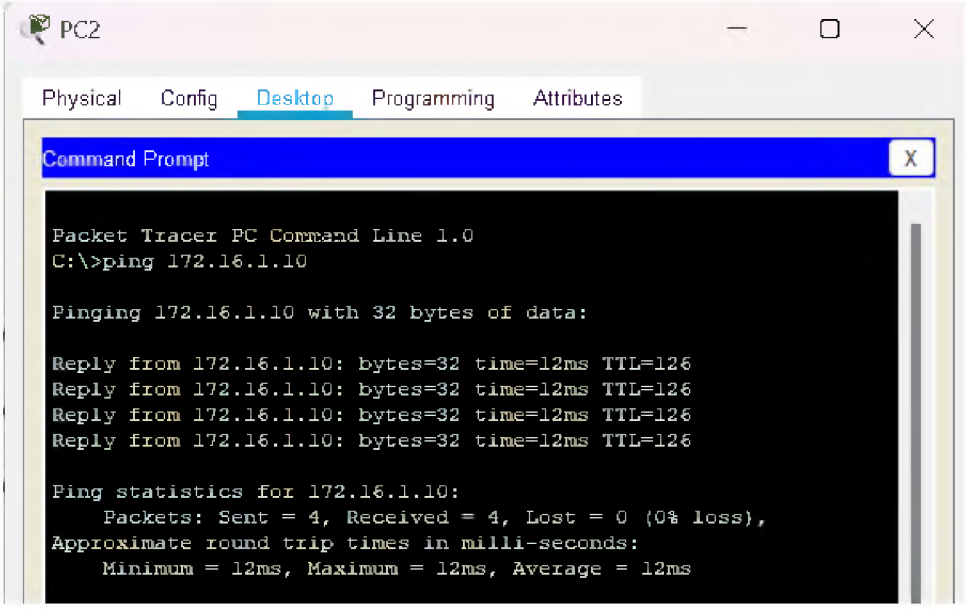


Рисунок 4.18 – Получение пакетов от PC8 в командной строке PC2

4.4 Для проверки соединения из всех подсетей сети С к сети А требуется зайти в режим симуляции и с каждого компьютера в командной строке написать команду ***ping * IP-адрес сети А *.*** Действия для проверки аналогичны пункту 4.3.

4.5 Для проверки соединения всех подсетей сети А требуется зайти в режим симуляции и с каждого компьютера в командной строке написать команду ***ping * IP-адрес сети А *.*** Действия для проверки аналогичны пункту 4.3. **Если** при проверке пинга различные подсети могут получать друг от друга ответы, то это говорит о неправильной настройке, требуется удостовериться что настройки IP-адресации правильные.

4.6 Для настройки списков доступа (ACL) на маршрутизаторе R1 требуется перейти во вкладку «CLI» и ввести следующие команды:

Для настройки выхода из сети А в сеть С напишите следующие команды:

```
R1(config)#ip access-list standard A_TO_C
R1(config-std-nacl)#permit 172.16.1.0 0.0.0.127
R1(config-std-nacl)#do wr
R1(config-std-nacl)#exit
R1(config)#int serial0/1/0
R1(config-if)#ip access-group A_TO_C in
R1(config-if)#do wr
```

Данные команды создадут список доступа с названием «A_TO_C» и разрешат подключение на порту Serial0/1/0 маршрутизатора R1 к сети С.

Для настройки ограничения обмена пакетами подсетей сети А требуется написать следующие команды:

```
R1(config)#ip access-list standard LAN
R1(config-std-nacl)#deny 10.1.0.128 0.0.0.127
R1(config-std-nacl)#deny 10.1.0.240 0.0.0.15
R1(config-std-nacl)#deny 10.1.0.224 0.0.0.31
R1(config-std-nacl)#permit any
R1(config-std-nacl)#do wr
R1(config-std-nacl)#exit
R1(config)#int gig0/0
R1(config-if)#ip access-group LAN out
R1(config-if)#do wr
```

Данные команды создадут список доступа с названием «LAN» и запретят подсетям сети А отправлять пакеты друг другу. Настройка производится на порту GigabitEthernet0/0.

На рисунках 4.19-4.20 представлена правильная работа списков для выхода сети А в сеть С и ограничения отправки пакетов между подсетями сети А. При работе с списками доступа используйте команды для проверки ACL в привилегированном режиме: ***sh ip access-lists*** и ***show ip interface | include line protocol|access list***

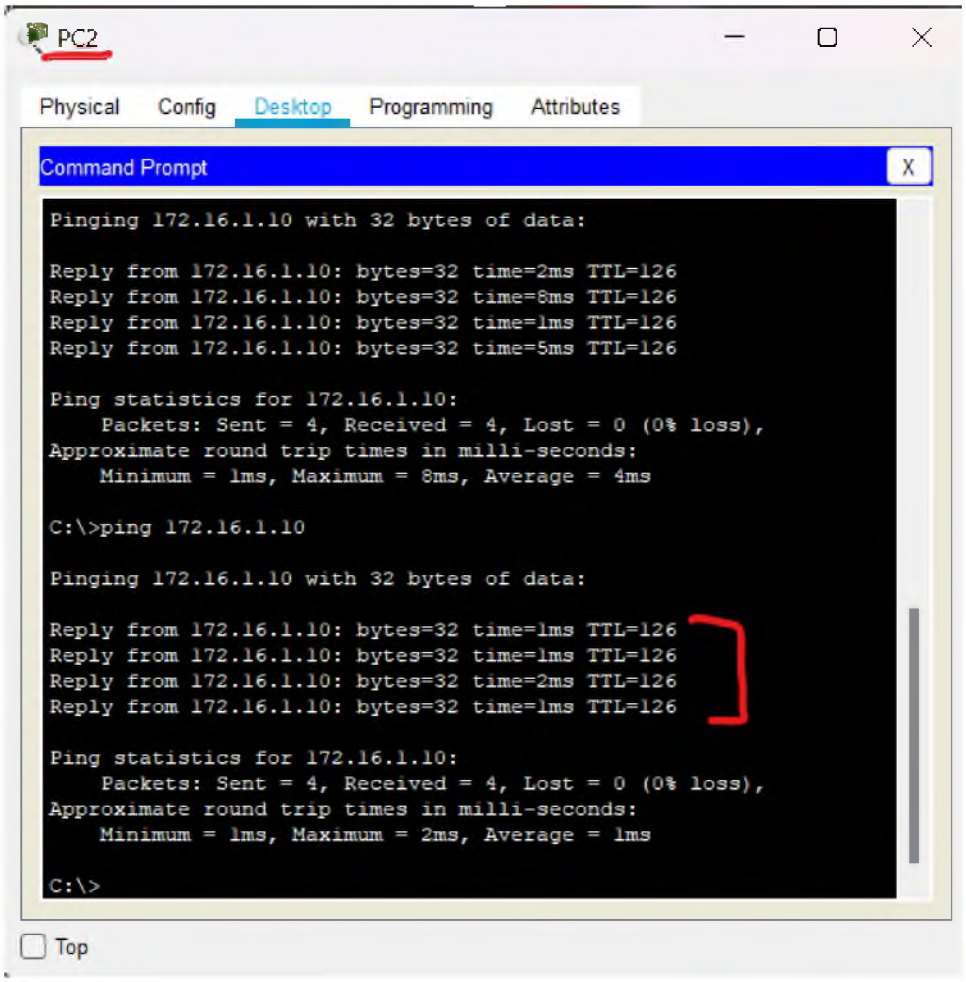


Рисунок 4.19 – Доступ к сети С из сети А

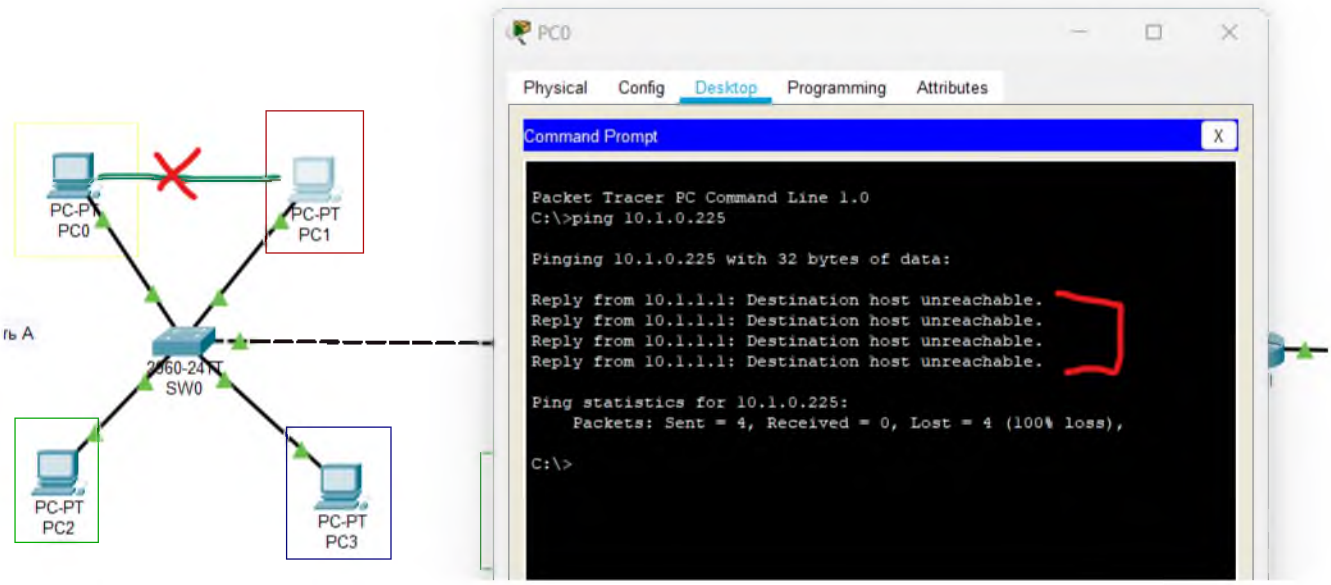


Рисунок 4.20 – Ограничение отправки пакетов подсетям сети

4.7 Для блокировки передачи пакетов из сети C в сеть A требуется создать на маршрутизаторе R1 новый расширенный список доступа (рисунок 4.21). Для этого используйте следующие команды:

R1(config)#ip access-list extended A_TO_C_WITH_BLOCK
R1(config-ext-nacl)# permit icmp 10.1.0.0 0.0.3.255 any echo
R1(config-ext-nacl)# deny icmp any 10.1.0.0 0.0.3.255 echo
R1(config-ext-nacl)# permit ip any any
R1(config-ext-nacl)#do wr
R1(config)#int s0/1/0
R1(config-if)#ip access-group A_TO_C_WITH_BLOCK in
R1(config-if)#do wr

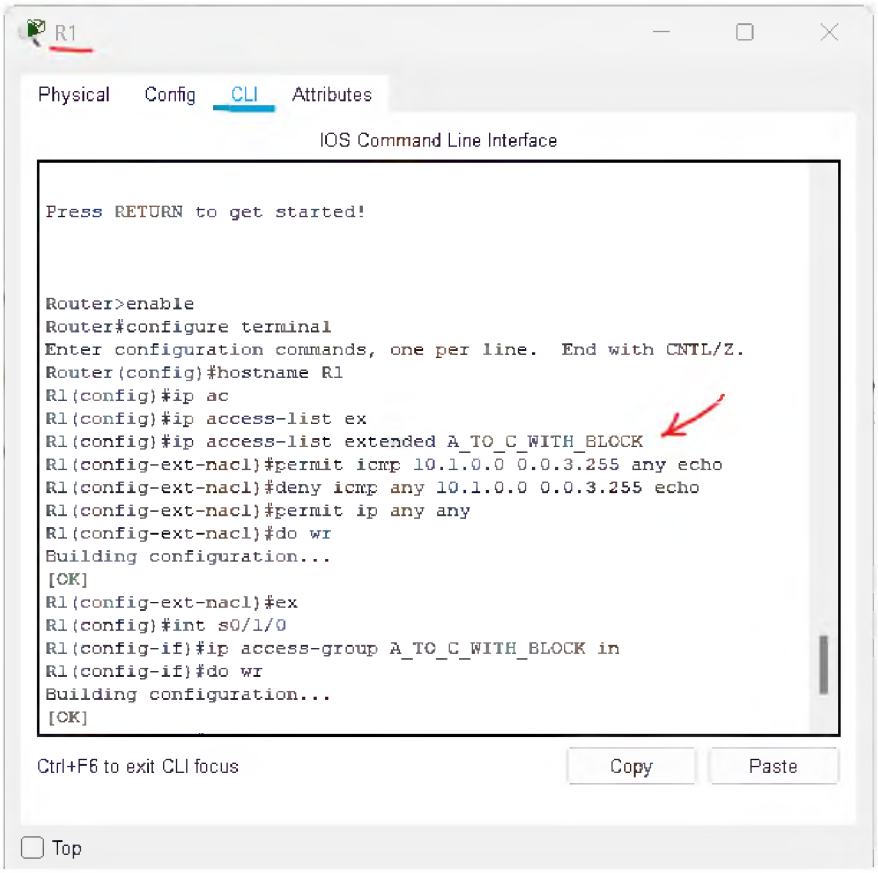


Рисунок 4.21 – Создание расширенного списка доступа

Протестируйте сеть отправив ICMP-пакет из компьютера сети A компьютеру сети C используя команду ping (рисунок 4.22). При отправке ICMP-запросов вы получите ICMP-ответ так как список доступа ACL настроен таким образом, чтобы сеть A могла посылать ICMP-запросы сети C. Теперь проверьте отправку ICMP-пакетов из сети C к сети A (рисунок 4.23). В командной строке вы увидите блокировку ICMP-запросов к сети A, так как список доступа ACL настроен таким образом, что сеть C не может посылать ICMP-запросы в сеть A.

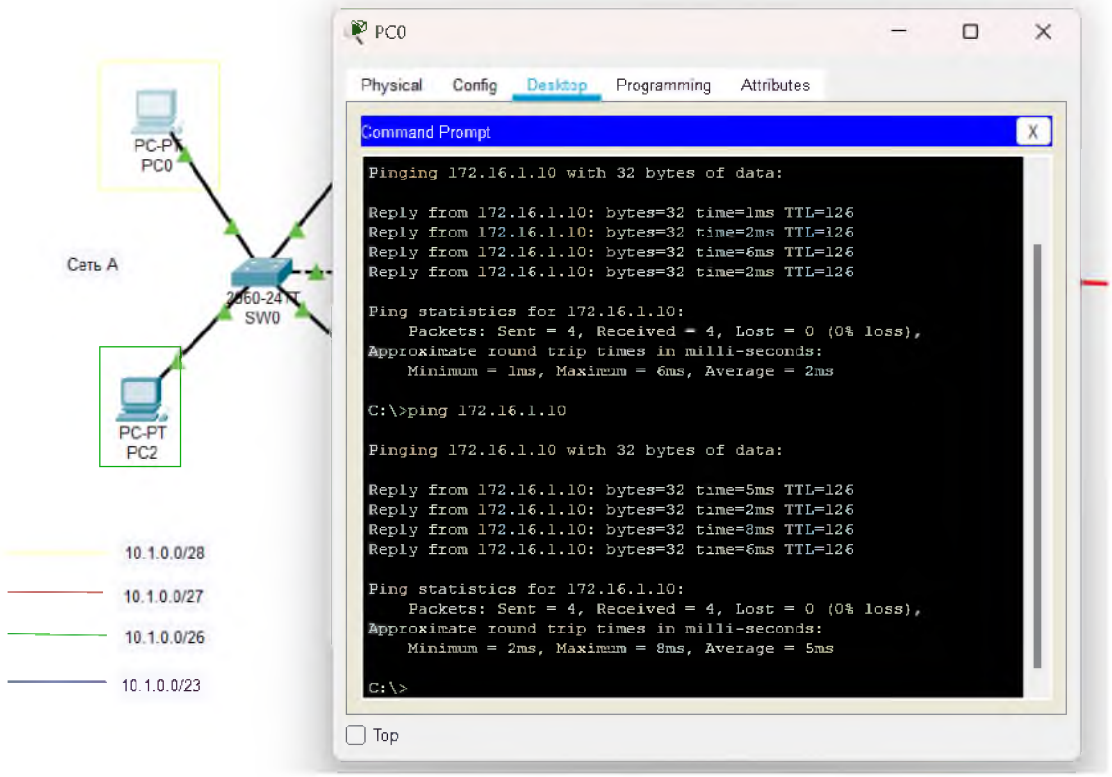


Рисунок 4.22 – Отправка пакетов из сети A в сеть C

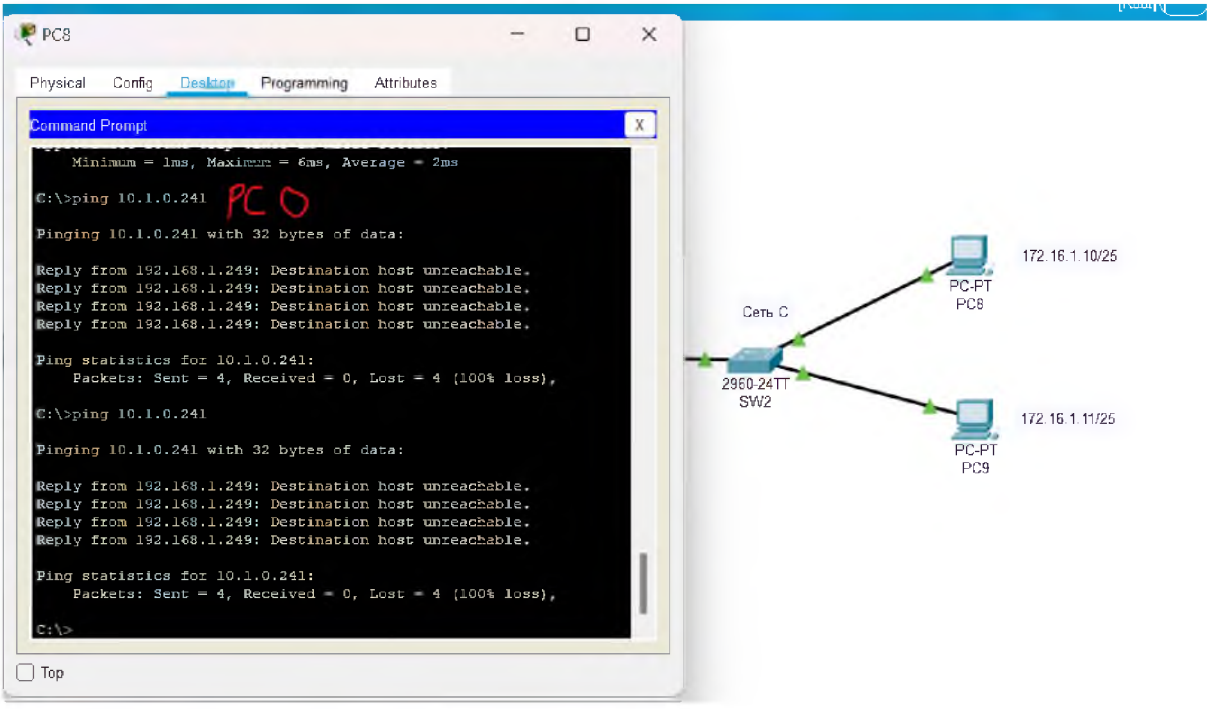


Рисунок 4.23 – Отправка пакетов из сети C в сеть A

5. Контрольные вопросы

1. Каким образом настраиваются статические маршруты на маршрутизаторах для связи между сетью A и сетью C?

2. Как проверить соединение из всех подсетей сети A к сети C с использованием команды ping?

3. Подробно опишите принцип работы ACL (списков доступа) на маршрутизаторе и как они реализуют контроль трафика в сети?

4. Какова роль статических маршрутов в сети, и как они обеспечивают маршрутизацию трафика между различными сетями?

Лабораторная работа № 10

Тема: Организация VPN-туннеля в сети

1. Цель: получить навыки организации VPN-туннеля в сети

Знать:

- назначение VPN-сервисов;
- применение VPN-туннеля.

Уметь:

- настраивать параметры VPN-туннеля.

2. Подготовка к работе:

- повторить теоретический материал по теме.

3. Задание:

3.1 Построить локальную сеть (рис. 10.1) в зависимости от вариантов (таб. 10.1) и настроить VPN тоннель между R2 и R3.

3.2 Настроить маршрутизацию в сети по протоколу RIPv2

3.3 Настроить лицензию пакета технологий безопасности

3.4 Настроить IPsec на маршрутизаторе R2

3.5 Настроить IPsec на маршрутизаторе R3

3.6 Проверить VPN туннель до прохождения по нему трафика

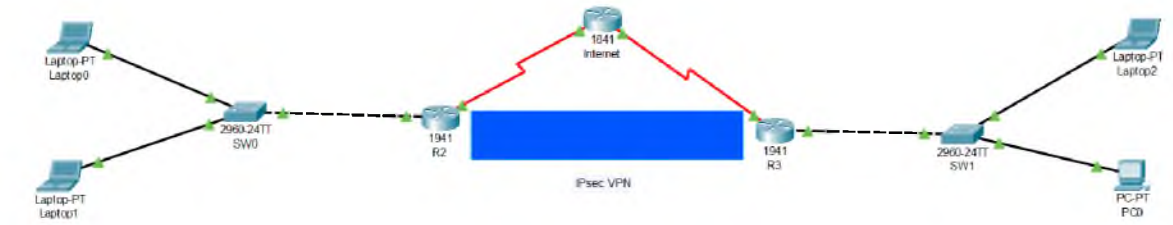


Рисунок 10.1 – Схема сети

Таблица 10.1. – Таблица вариантов

Номер варианта	Адрес сети A	Адрес сети B	Адрес сети C	Адрес сети D
1	192.168.1.0/25	11.0.0.0/23	21.0.0.0/22	172.16.1.0/26
2	192.168.2.0/26	12.0.0.0/22	22.0.0.0/21	172.16.2.0/27
3	192.168.3.0/27	13.0.0.0/21	23.0.0.0/20	172.16.3.0/28
4	192.168.4.0/28	14.0.0.0/20	24.0.0.0/19	172.16.4.0/29

5	192.168.5.0/29	15.0.0.0/19	25.0.0.0/18	172.16.5.0/28
6	192.168.6.0/28	16.0.0.0/18	26.0.0.0/17	172.16.6.0/27
7	192.168.7.0/27	17.0.0.0/17	27.0.0.0/16	172.16.7.0/26
8	192.168.8.0/26	18.0.0.0/16	28.0.0.0/17	172.16.8.0/25
9	192.168.9.0/25	19.0.0.0/17	29.0.0.0/18	172.16.9.0/26
10	192.168.10.0/26	20.0.0.0/18	30.0.0.0/19	172.16.10.0/27
11	192.168.11.0/27	21.0.0.0/19	31.0.0.0/20	172.16.11.0/28
12	192.168.12.0/28	22.0.0.0/20	32.0.0.0/21	172.16.12.0/29
13	192.168.13.0/29	23.0.0.0/21	33.0.0.0/22	172.16.13.0/28
14	192.168.14.0/28	24.0.0.0/22	34.0.0.0/23	172.16.14.0/27
15	192.168.15.0/27	25.0.0.0/23	35.0.0.0/22	172.16.15.0/26
16	192.168.16.0/26	26.0.0.0/22	36.0.0.0/21	172.16.16.0/25
17	192.168.17.0/25	27.0.0.0/21	37.0.0.0/20	172.16.17.0/26
18	192.168.18.0/26	28.0.0.0/20	38.0.0.0/19	172.16.18.0/27
19	192.168.19.0/27	29.0.0.0/19	39.0.0.0/18	172.16.19.0/28
20	192.168.20.0/28	30.0.0.0/18	40.0.0.0/17	172.16.20.0/29
21	192.168.21.0/29	31.0.0.0/17	41.0.0.0/18	172.16.21.0/28
22	192.168.22.0/28	32.0.0.0/18	42.0.0.0/19	172.16.22.0/27
23	192.168.23.0/27	33.0.0.0/19	43.0.0.0/20	172.16.23.0/26
24	192.168.24.0/26	34.0.0.0/20	44.0.0.0/21	172.16.24.0/25
25	192.168.25.0/25	35.0.0.0/21	45.0.0.0/22	172.16.25.0/26
26	192.168.26.0/26	36.0.0.0/22	46.0.0.0/23	172.16.26.0/27
27	192.168.27.0/27	37.0.0.0/23	47.0.0.0/22	172.16.27.0/28
28	192.168.28.0/28	38.0.0.0/22	48.0.0.0/23	172.16.28.0/29
29	192.168.29.0/29	39.0.0.0/21	49.0.0.0/22	172.16.29.0/28
30	192.168.30.0/28	40.0.0.0/20	50.0.0.0/21	172.16.30.0/27

4. Порядок выполнения

4.1 Построить топологию сети, приведённую на рисунке 10.1. Присвоить IP-адреса согласно своему варианту. На рисунке 4.1 приведена схема сети с надписями присвоенных IP-адресов.

Рисунок 4.1 – Топология сети

4.2 Для настройки маршрутизации по протоколу RIPv2 перейдите в режим конфигурирования маршрутизаторов и напишите следующие команды:

router rip

version 2

network <номер сети, которая **НЕПОСРЕДСТВЕННО** подключена к данному маршрутизатору>

Пример команд для настройки RIPv2 для маршрутизатора R2:

R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 11.0.0.0
R2(config-router)#network 192.168.1.0
R2(config-router)#do wr

Настройка маршрутизации для роутеров R3 и Internet осуществляется аналогично.

4.3 Введите команду ***show version*** на Cisco 1941 в пользовательском или привилегированном режиме, чтобы убедиться, что лицензия пакета технологий безопасности активирована, на маршрутизаторах R2 и R3 требуется нажимать клавишу ***ENTER*** чтобы дойти до соответствующего раздела (рисунок 4.2). Если в графе стоит «None» требуется активировать модуль securityk9, принять лицензию, сохранить настройку и перезагрузить маршрутизатор.

Требуется согласиться на использование securityk9 (рисунок 4.3). Для этого введите команды:

R2(config)# license boot module c1900 technology-package securityk9
R2(config)# end
R2# reload

А затем напишите “yes” чтобы установить лицензию.

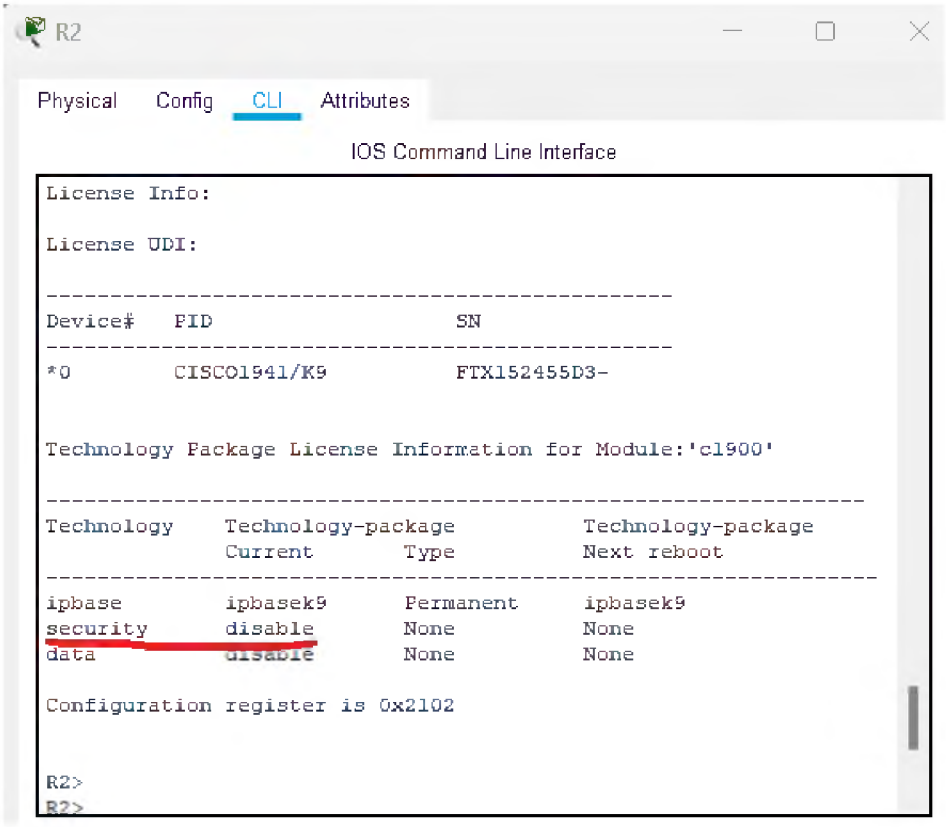


Рисунок 4.2 – Проверка лицензий пакета технологий безопаности

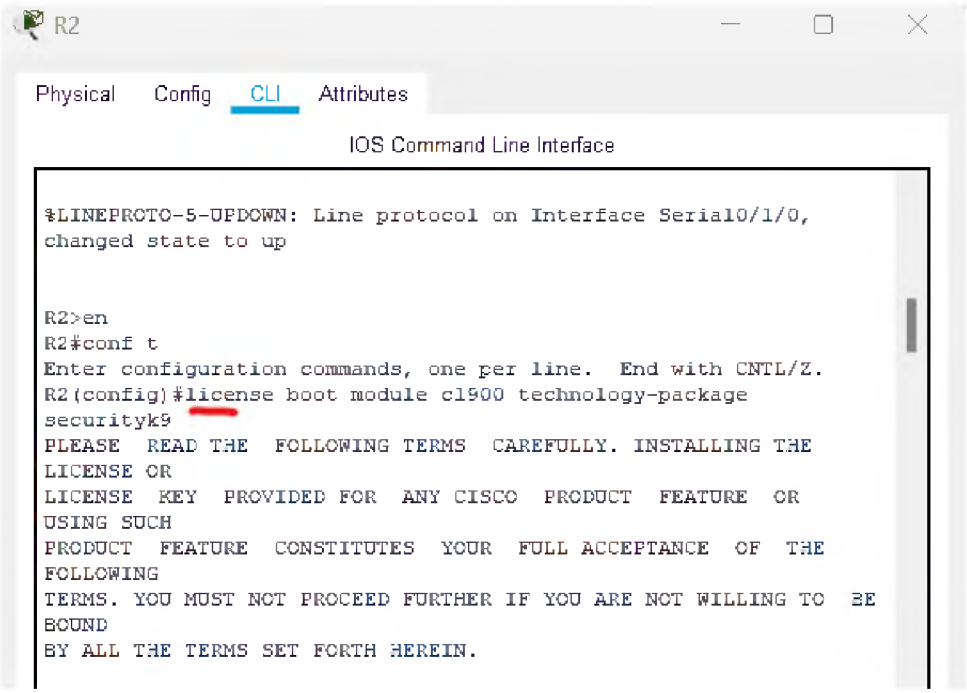


Рисунок 4.3 – Добавление лицензии securityk9

После установки и перезагрузки роутера, напишите команду ***show version*** чтобы удостовериться что всё установилось (рисунок 4.4).

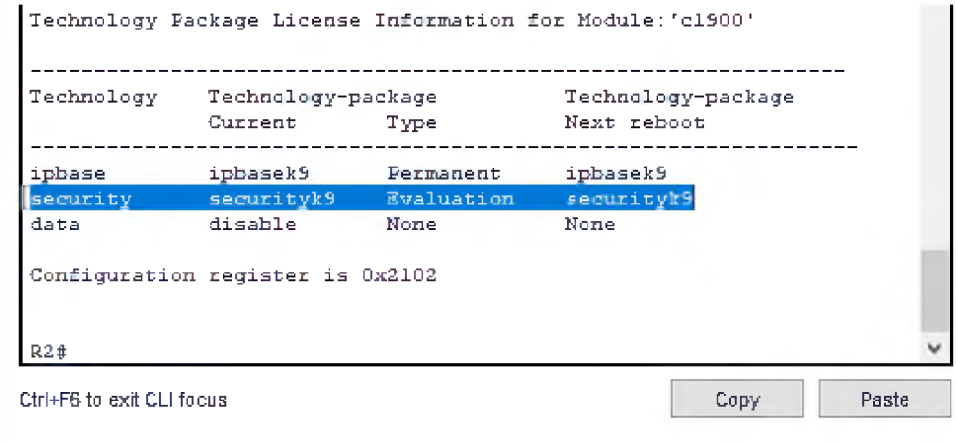


Рисунок 4.4 – Отображение лицензии securityk9

4.4 Для настройки IPsec на маршрутизаторе R2 требуется задать ACL-список чтобы определить трафик между маршрутизаторами R2 и R3 (рисунок 4.5). Для этого введите следующие команды:

R2(config)#access-list 110 permit ip 192.168.1.0 0.0.0.127 172.16.1.0 0.0.0.63
R2(config)#do wr

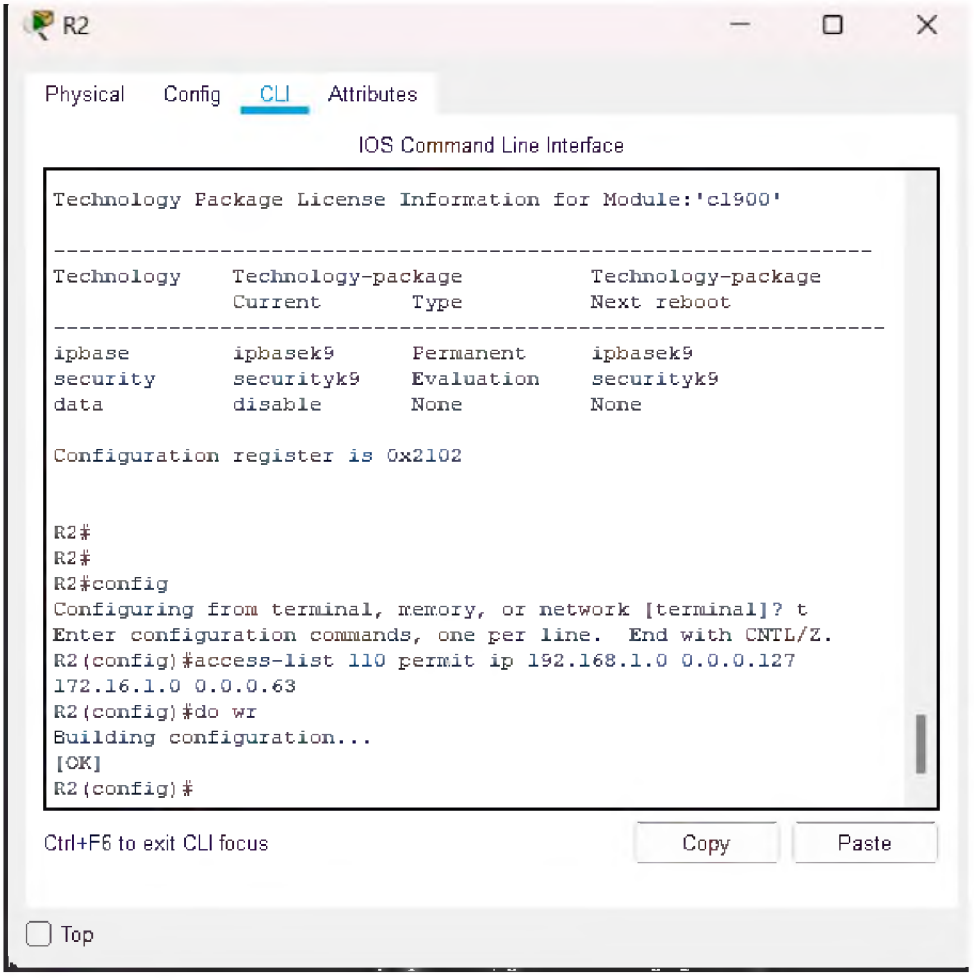


Рисунок 4.5 – Создание списка доступа

Далее перейдите к настройке 1 фазы ISAKMP на маршрутизаторе R2. Необходимо настроить на маршрутизаторе R2 свойства криптографической политики ISAKMP 10, а также общий ключ шифрования cisco (рисунок 4.6). В качестве адреса криптографии указывается адрес порта R3, подключенного к сети Internet-R3 (сеть C). Для этого напишите следующие команды на маршрутизаторе R2:

R2(config)# crypto isakmp policy 10
R2(config-isakmp)# encryption aes
R2(config-isakmp)# authentication pre-share
R2(config-isakmp)# group 2
R2(config-isakmp)# exit
R2(config)#crypto isakmp key cisco address 21.0.0.2

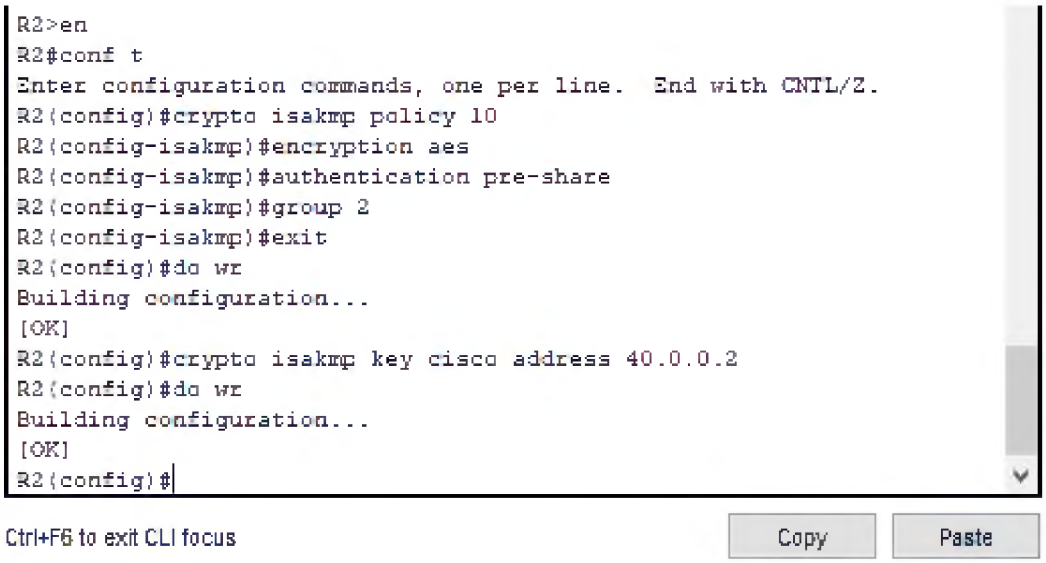


Рисунок 4.6 – Настройка криптографической политики ISAKMP

Перейдите к настройке 2 фазы ISAKMP на маршрутизаторе R2. Требуется создать набор преобразований (transform-set) VPN-SET для использования esp-3des и esp-shahmac. Затем создать криптографическое сопоставление (crypto map) VPN-MAP, которое связывает вместе все параметры 2 фазы. Нужно использовать порядковый номер 10 и определить его в качестве сопоставления ipsec-isakmp (рисунок 4.7).

R2(config)# crypto ipsec transform-set VPN-SET esp-3des esp-sha-hmac
R2(config)# crypto map VPN-MAP 10 ipsec-isakmp
R2(config-crypto-map)# description VPN connection to R3
R2(config-crypto-map)# set peer 21.0.0.2
R2(config-crypto-map)# set transform-set VPN-SET
R2(config-crypto-map)# match address 110
R2(config-crypto-map)# exit

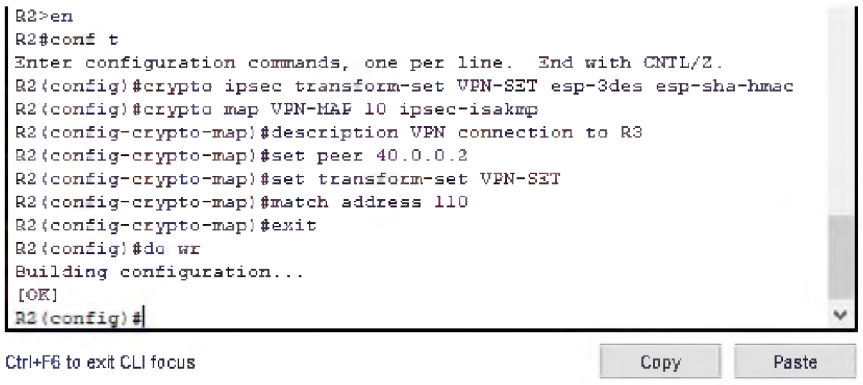


Рисунок 4.7 – Настройка криптографической политики ISAKMP

Далее перейдите к настройке криптографического сопоставления для исходящего интерфейса (рисунок 4.8)

R2(config)# interface S0/1/0
R2(config-if)# crypto map VPN-MAP

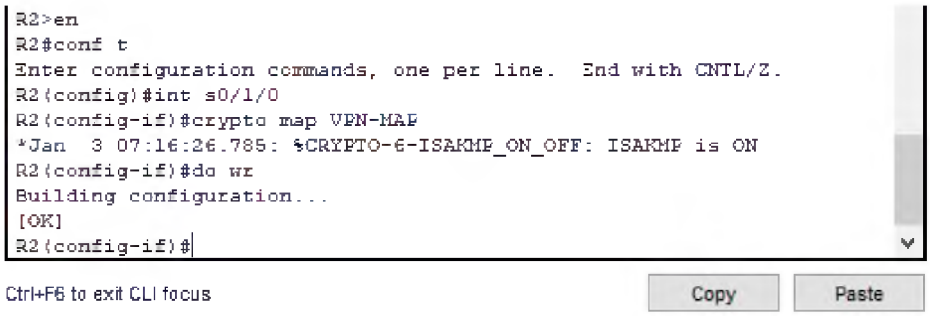


Рисунок 4.8 – Настройка криптографического сопоставления ISAKMP

4.5 Для настройки IPsec на R3 создайте ACL-список, который будет определять проходимый трафик между маршрутизаторами R2 и R3 (рисунок 4.9).

R3(config)#access-list 110 permit ip 172.16.1.0 0.0.0.63 192.168.1.0 0.0.0.127
R3(config)#do wr

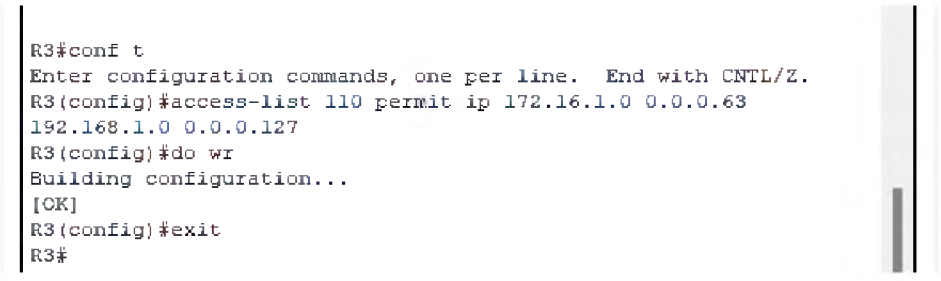


Рисунок 4.9 – Создание списка доступа

Перейдите к настройке 1 фазы ISAKMP на маршрутизаторе R3. Требуется настроить на маршрутизаторе R3 свойства криптографической политики ISAKMP 10, а также общий ключ шифрования cisco. В качестве адреса

криптографии указывается адрес порта R2, подключенного к сети Internet-R2 (сеть B).

R3(config)# crypto isakmp policy 10
R3(config-isakmp)# encryption aes
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# group 2
R3(config-isakmp)# exit
R3(config)# crypto isakmp key cisco address 11.0.0.1

Затем перейдите к настройке 2 фазы ISAKMP на маршрутизаторе R3 аналогично пункту 4.4.

R3(config)# crypto ipsec transform-set VPN-SET esp-3des esp-sha-hmac
R3(config)# crypto map VPN-MAP 10 ipsec-isakmp
R3(config-crypto-map)# description VPN connection to R2
R3(config-crypto-map)# set peer 11.0.0.1
R3(config-crypto-map)# set transform-set VPN-SET
R3(config-crypto-map)# match address 110
R3(config-crypto-map)# exit

Далее перейдите к настройке криптографического сопоставления для исходящего интерфейса.

R3(config)# interface S0/1/1
R3(config-if)# crypto map VPN-MAP

4.6 Требуется проверить проверить туннель до прохождения по нему интересующего трафика. Для этого вводится команда **show crypto ipsec sa** на маршрутизаторе R2 (рис 4.10). Необходимо обратить внимание, что количество всех пакетов (инкапсулированных, зашифрованных, декапсулированных и дешифрованных) равно 0.

```
R2#show crypto ipsec sa

interface: Serial0/1/0
    Crypto map tag: VPN-MAP, local addr 11.0.0.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.128/0/0)
   remote  ident (addr/mask/prot/port): (172.16.1.0/255.255.255.192/0/0)
   current_peer 21.0.0.2 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
   #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 0, #pkts compr. failed: 0
   #pkts not decompressed: 0, #pkts decompress failed: 0
   #send errors 0, #recv errors 0

     local crypto endpt.: 11.0.0.1, remote crypto endpt.:21.0.0.2
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/1/0
     current outbound spi: 0x0(0)
```

Рисунок 4.10 – Информация о трафике маршрутизатора R2

Создать интересующий трафик. Для этого нужно прописать *ping* с компьютера сети A до ПК в сети D. Как видно, приходит ответ от хоста 172.16.1.1 (рисунок 4.11).

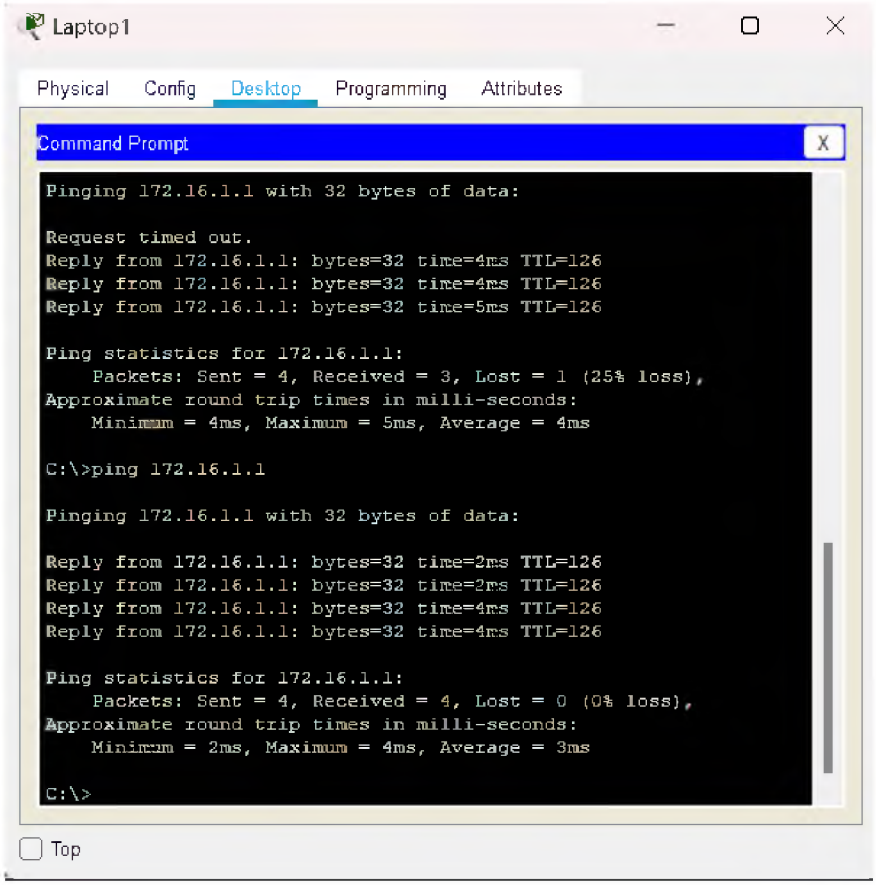


Рисунок 4.11 – Создание трафика от сети A к сети D

Чтобы проверить работоспособность VPN-IPsec канала, нужно прописать заново ***show crypto ipsec sa.*** Число инкапсулированных и декапсулированных пакетов увеличилось (рис 4.12).

Рисунок 4.12 – Вывод инкапсулированных и декапсулированных пакетов

Для проверки, в режиме симуляции необходимо отправить пакеты протокола ICMP (команда *ping*) от 192.168.1.1 (ПК) к 172.16.1.1 (ПК), как видно пакеты доставляются обратно (рисунок 4.13), значит сеть работает исправно. Остановить симуляцию на третьем пакете ICMP на маршрутизаторе R3. Рассмотреть содержимое пакета, вошедшего в маршрутизатор (*Inbound PDU Details*) и вышедшего из маршрутизатора (*Outbound PDU Details*). Объяснить разницу.

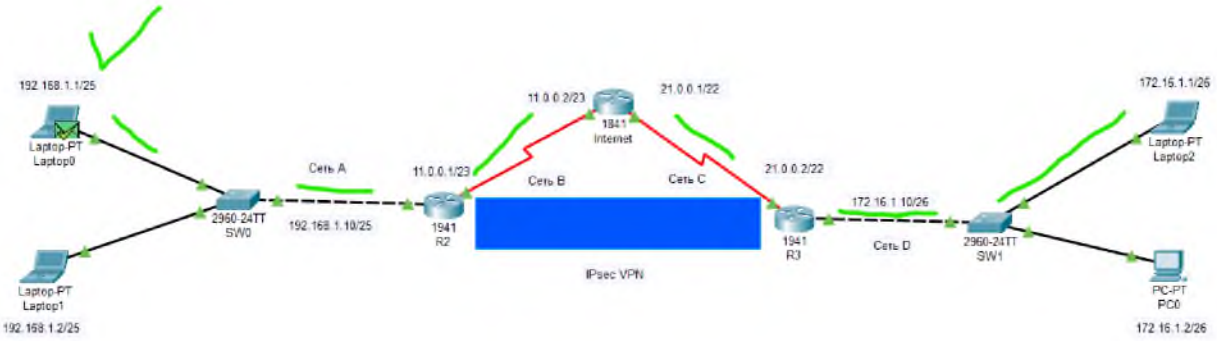


Рисунок 4.13 – Тестирование IPsec канала в режиме симуляции

Рассмотрим сигнатуру PDU-фрагмента, вошедшего в маршрутизатор R2 (рисунки 4.14-4.15). На входе в маршрутизатор PDU-фрагмент не был инкапсулирован и защищён системой шифрования данных. После того как PDU-фрагмент пройдёт этап шифрования данных к нему добавится ESP-заголовок (рисунок 4.16).

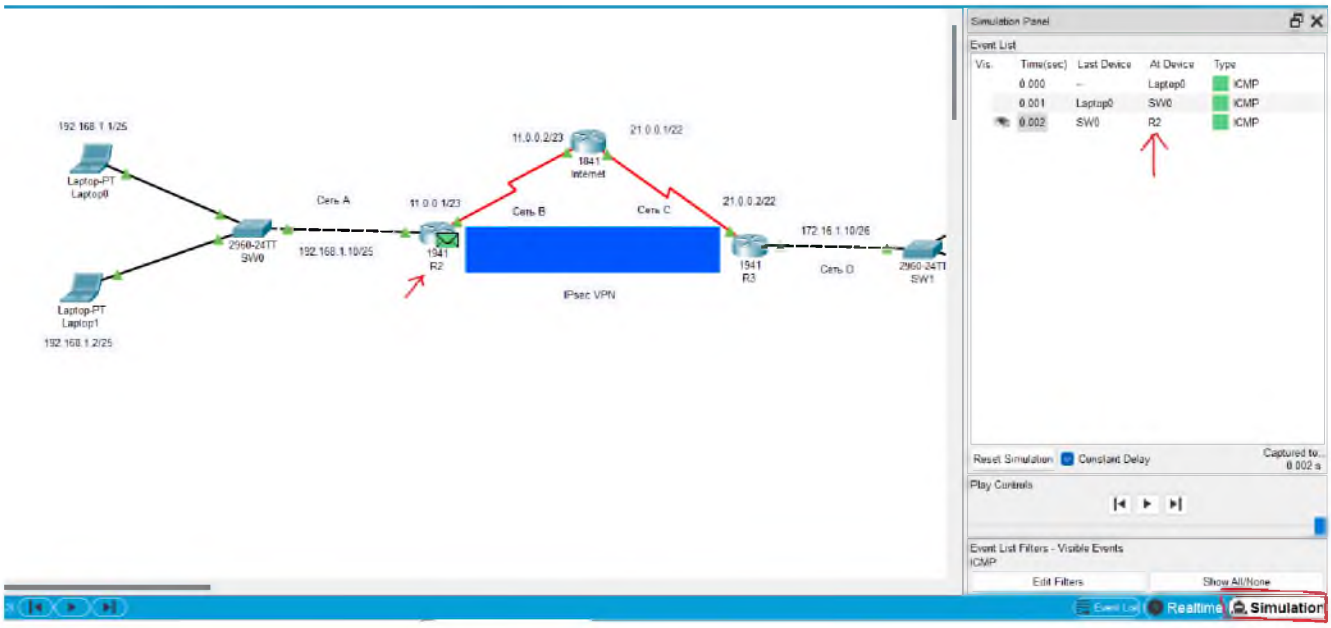


Рисунок 4.14 – Отображение PDU-фрагмента на схеме

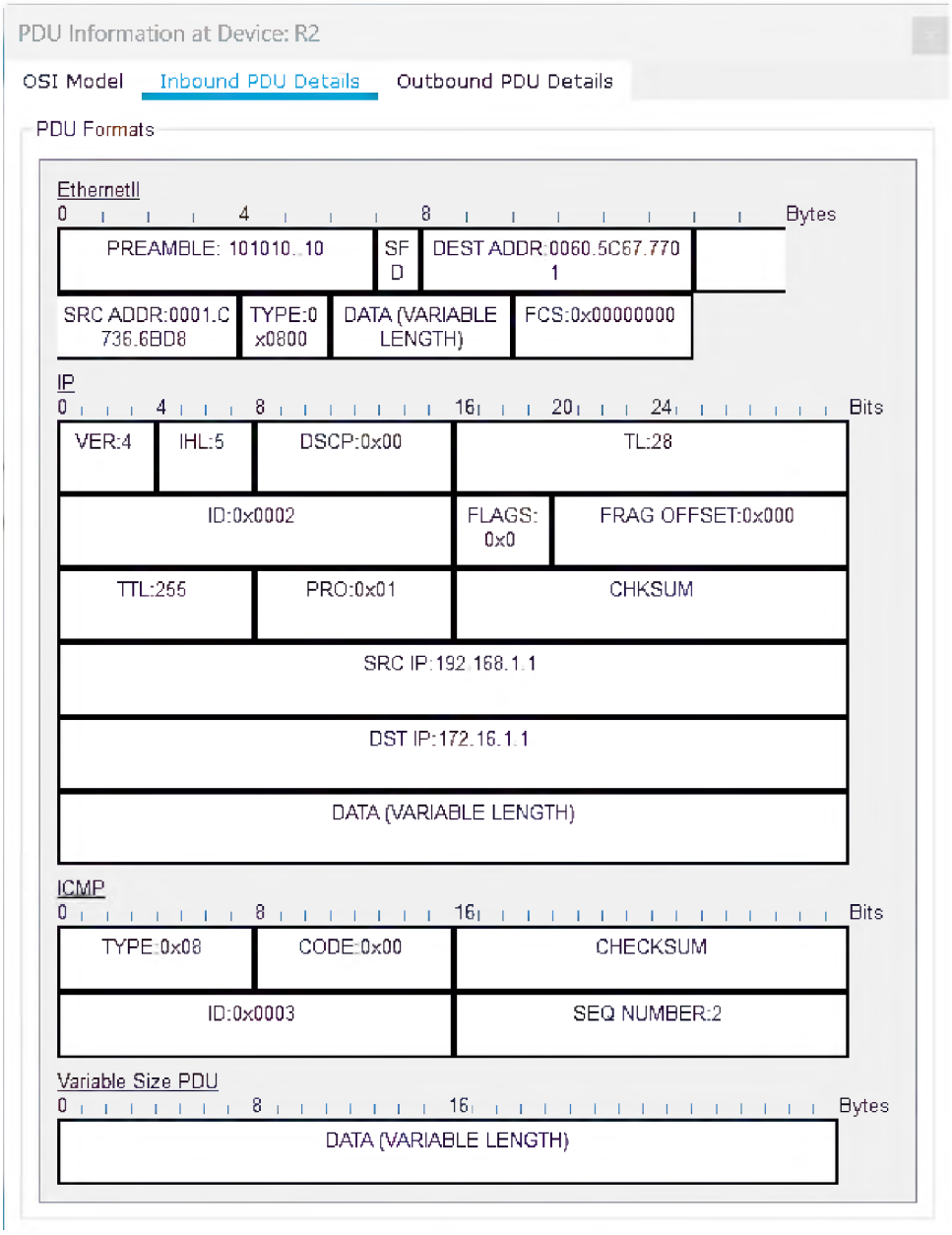


Рисунок 4.15 – PDU-фрагмент, пришедший на маршрутизатор R2

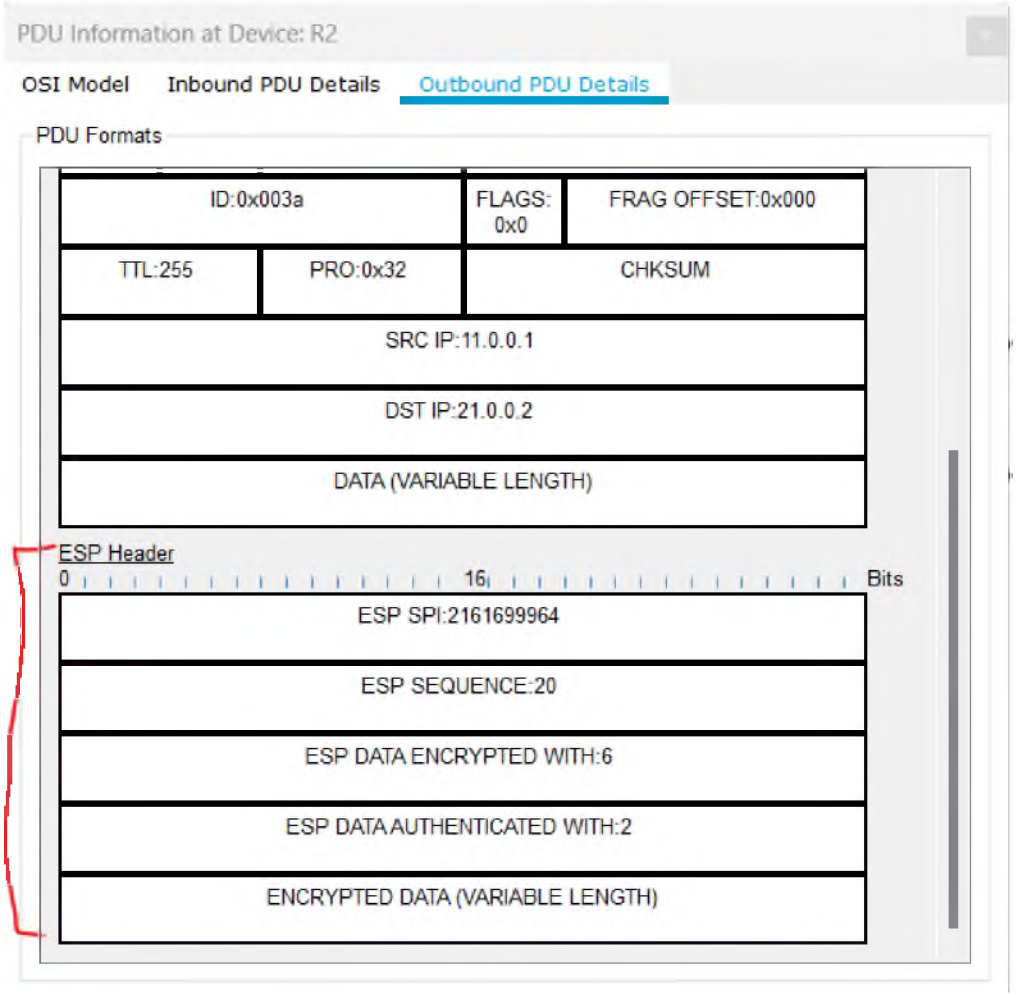


Рисунок 4.16 – Содержимое вышедшего из маршрутизатора R2 PDU-фрагмента

ESP (Encapsulating Security Payload) — это один из протоколов, используемых в IPsec (Internet Protocol Security) для обеспечения конфиденциальности, целостности данных и аутентификации в сетях IP. В рамках ESP используются следующие элементы:

ESP SPI (Security Parameter Index) — это 32-битное поле, которое идентифицирует конкретное соединение между двумя узлами. SPI позволяет получателю правильно обработать зашифрованные и аутентифицированные пакеты, указывая на соответствующие параметры безопасности.

ESP Sequence Number — это поле, которое используется для предотвращения повторных или воспроизведенных пакетов. Каждый пакет ESP содержит уникальный номер последовательности, который позволяет получателю упорядочивать и проверять целостность пакетов.

ESP Data Encrypted — это часть ESP заголовка, которая содержит зашифрованные данные. Зашифрованные данные представляют собой полезную

нагрузку пакета, которая была защищена от несанкционированного доступа с использованием алгоритмов шифрования.

ESP Data Authenticated with Encrypted Data — это процесс аутентификации данных в ESP заголовке с использованием алгоритмов хэширования или кодов аутентичности сообщений (HMAC). Это обеспечивает подтверждение целостности данных и подлинности отправителя.

Encrypted Data (Variable Length) — это часть ESP пакета, которая содержит зашифрованные данные переменной длины. Зашифрованные данные могут включать в себя полезную информацию, такую как IP-пакеты, TCP/UDP сегменты и другие данные, которые должны быть защищены при передаче по сети.

5. Контрольные вопросы

1. Что такое VPN-сервисы и каково их назначение?
2. В чем заключается применение VPN-туннеля?
3. Какие этапы подготовки к работе рекомендуется выполнить перед началом настройки VPN-туннеля?
4. Что такое IPsec и для чего он используется в данной работе?
5. Как можно проверить работоспособность VPN-туннеля?