**МИНОБРНАУКИ РОССИИ  
 Федеральное государственное бюджетное   
образовательное учреждение высшего образования**  
 «Ухтинский государственный технический университет»   
(УГТУ)

Кафедра вычислительной техники, информационных систем и технологий

**РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА**Дисциплина: «Сети и телекоммуникации»

Шифр 191407 Группа ИCТ-19о-Б Курс 3 Вариант 1  
  
Морданов Егор Владимирович

Проверил:  
доцент кафедры ВТИСиТ Григорьевых А.В.

Ухта

2021

СОДЕРЖАНИЕ

[Основная часть 3](#_Toc93704314)

[Теоретический вопрос 8](#_Toc93704315)

[Список использованной литературы 10](#_Toc93704316)

Основная часть

Задача: рассчитать характеристики подсетей (адреса, маски, диапазоны IP для машин) для сети: 192.168.11.0 и маски 255.255.255.0 (см. Таблица 1).

Подсети 1— 5 машина (докупят еще 13)

Подсеть 2 — 10 машин

Подсеть 3 — 4 машины

Подсеть 4 — 2 машин (докупят еще 40)

В двоичном представлении ip адрес – 1100000.10101100.00001011.00000000

В двоичном представлении маска – 11111111. 111111111. 111111111. 11000000

Максимальное число хостов в подсетях = 42. Соответственно на одного хоста уйдет 6 бит. 2^5 = 32 и 2^6 = 64. Берем большее значение. Таким образом, на ip адрес подсети уйдет 2 бита, так как 2^2 = 4.

Рассчитаем ip адреса подсетей:

* Первая подсеть: 192.168.11.0/26

Ip адрес - 11000000. 10101000.00001011.00000000

Маска - 111111111. 111111111. 111111111. 11000000

* Вторая подсеть: 192.168.11.64/26

Ip адрес - 1100000.10101100.00001011. 01000000

Маска - 111111111. 111111111. 111111111. 11000000

* Первая подсеть: 192.168.11.128/26

Ip адрес - 1100000.10101100.00001011. 10000000

Маска - 111111111. 111111111. 111111111. 11000000

* Первая подсеть: 192.168.11.192/26

Ip адрес - 1100000.10101100.00001011. 11000000

Маска - 111111111. 111111111. 111111111. 11000000

Маска подсетей в десятичном представлении равна 255.255.255.192.

Таблица 1 - Характеристики подсетей

|  |  |  |
| --- | --- | --- |
| Диапазон ip адресов подсети | Адрес подсети | Широковещательный адрес |
| 192.168.11.1 – 192.168.11.63 | 192.168.11.1 | 192.168.11.63 |
| 192.168.11.64 – 192.168.11.127 | 192.168.11.64 | 192.168.11.127 |
| 192.168.11.128– 192.168.11.191 | 192.168.11.128 | 192.168.11.191 |
| 192.168.11.192– 192.168.11.255 | 192.168.11.192 | 192.168.11.255 |

Таким образом, получаем, что в каждой подсети доступно 62 хоста.

Далее приступим к проектированию сети в Cisco Packet Tracer. Выставим на диаграмму все хосты каждой подсети. Далее объединим компьютеры каждой подсети коммутаторами (см. Рисунок 1).

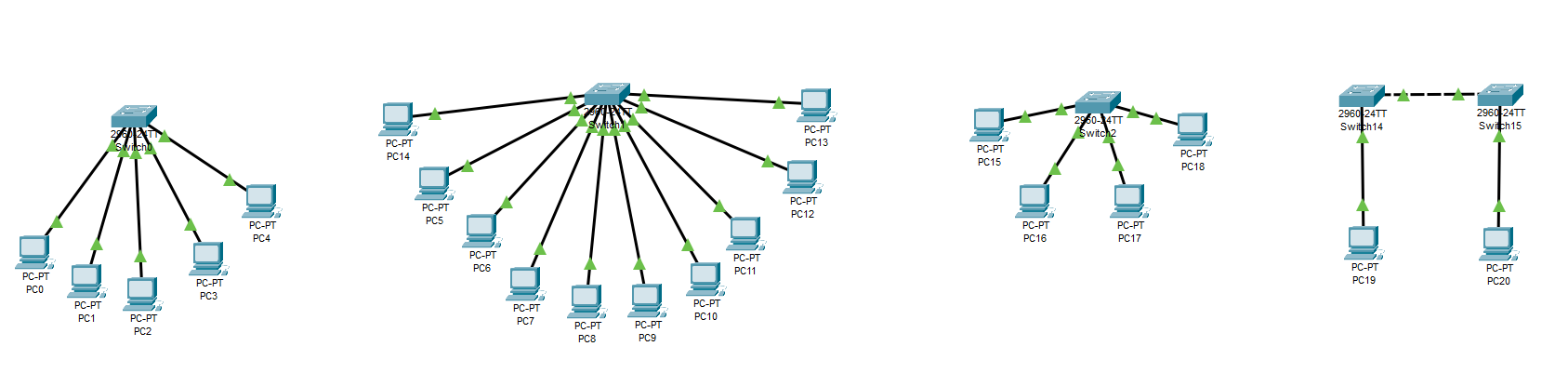


Рисунок 1 - Подсети

Теперь попробуем пингануть компьютеры между собой, которые находится в одной сети (см. Рисунок 2).

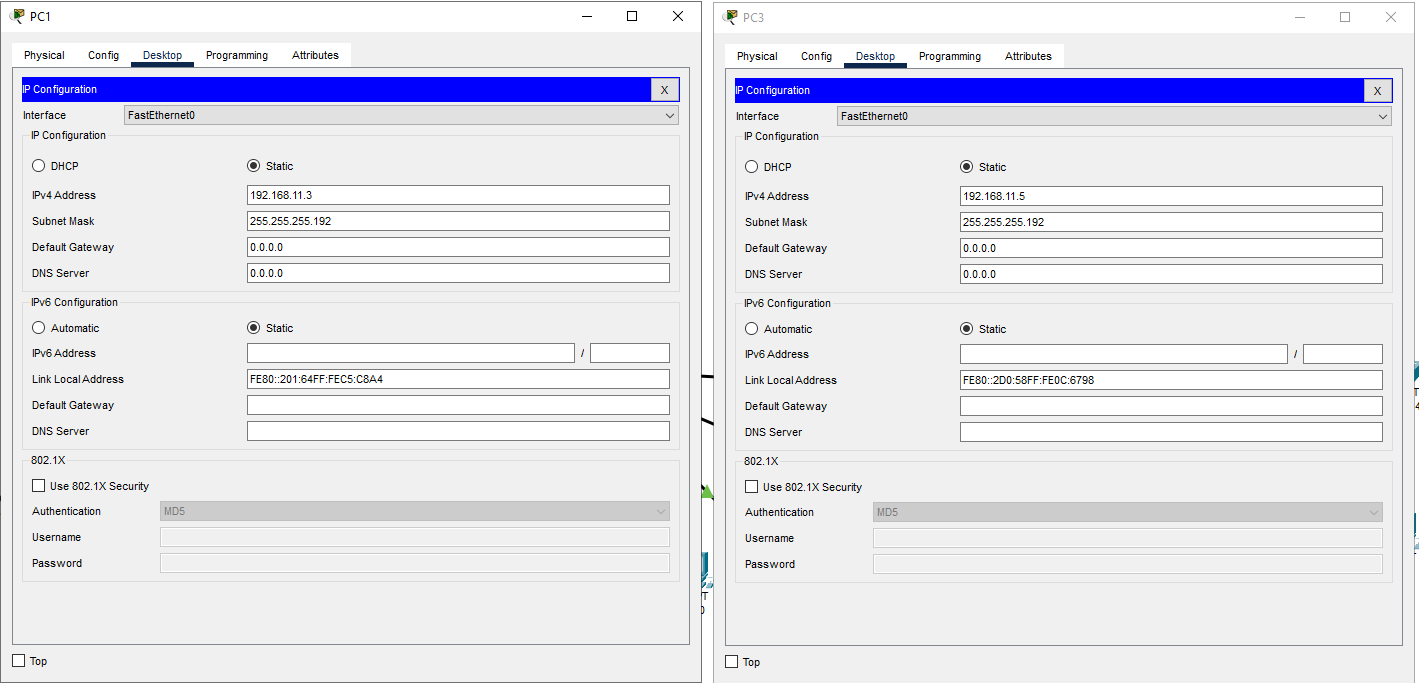


Рисунок 2 – PC1 и PC3

Как можно заметить на Рисунке 3 на компьютеры одной сети обмениваются пакетами ping.

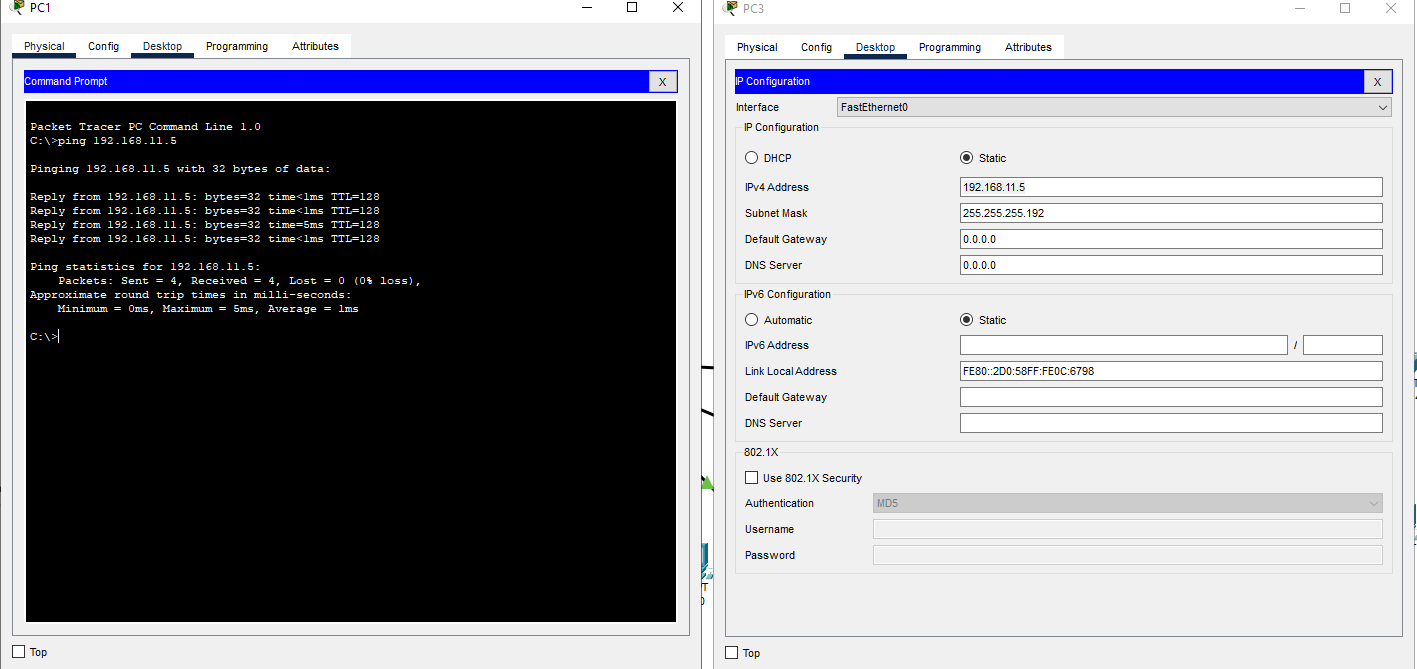


Рисунок 3 – Ping

Далее соединим коммутаторы каждой сети с роутером (см. Рисунок 4).

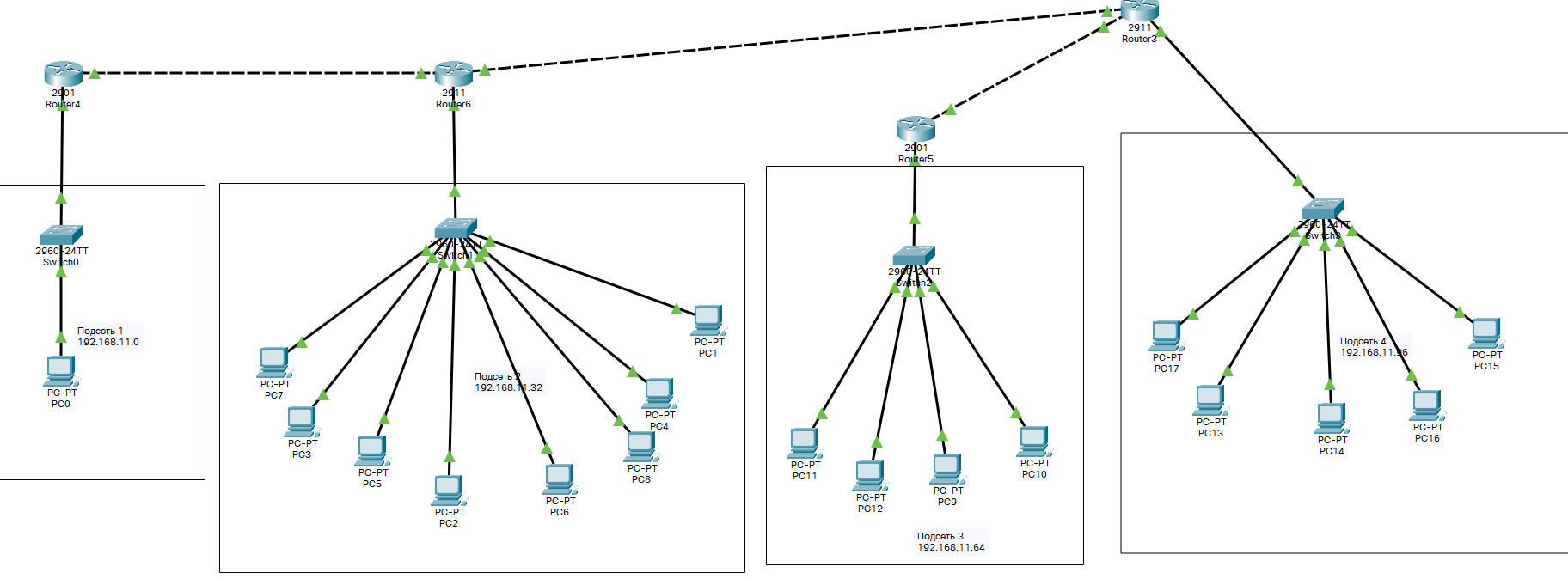


Рисунок 4 - Добавление роутеров

Объединим роутеры между собой и настроим статическую маршрутизацию. Таким образом, полностью готовая модель (см. Рисунок 5).

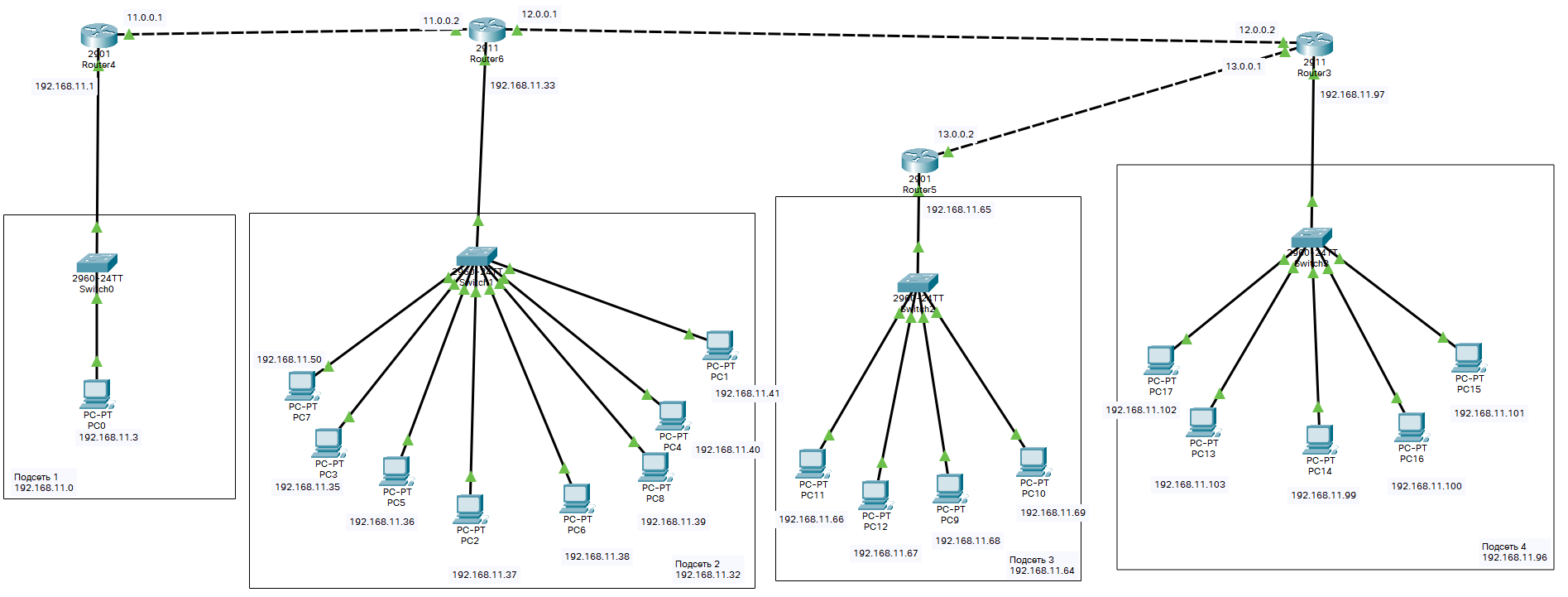


Рисунок 5 - Готовая модель

Проверим связь между компьютерами.

Используя ping и tracert проверим связь между PC0 и PC7 (см. Рисунок 6).

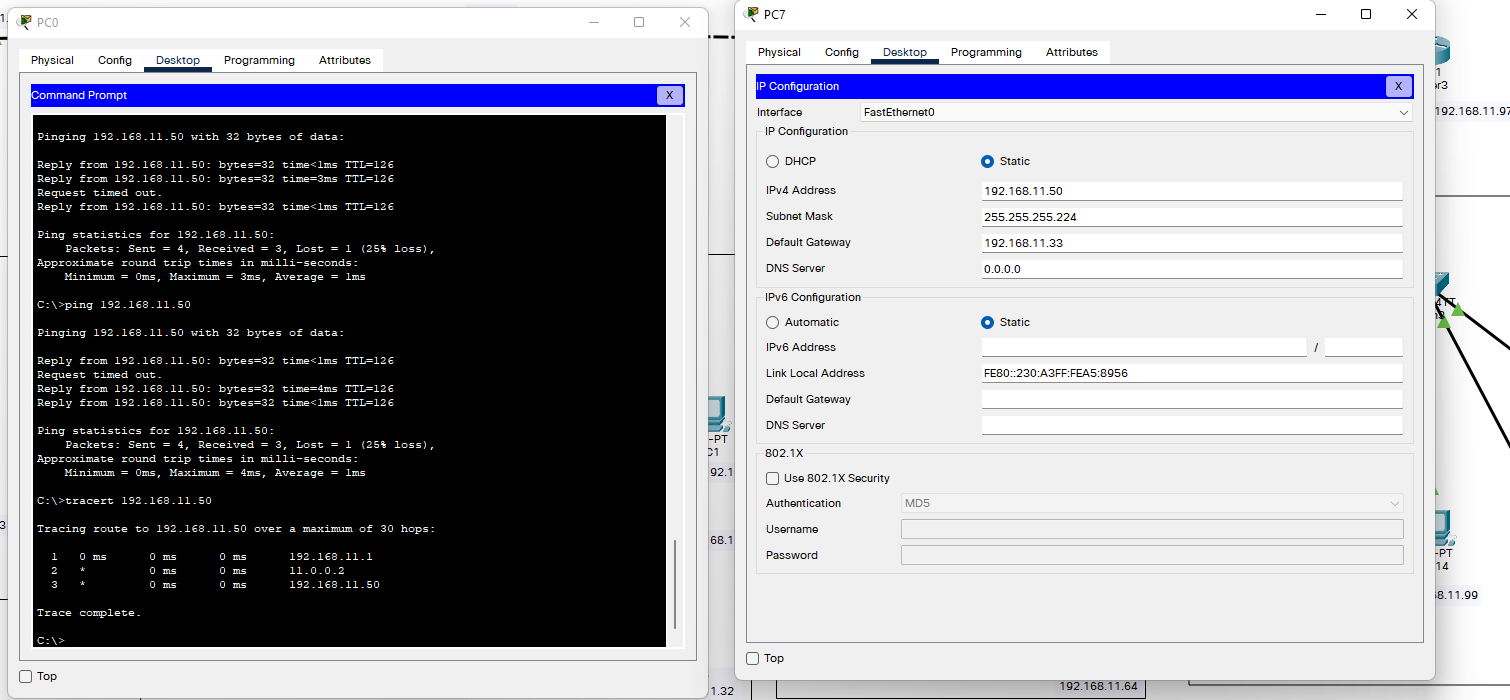


Рисунок 6 - Связь между PC0 и PC7

Теперь проверим PC0 и PC11 (см. Рисунок 7).

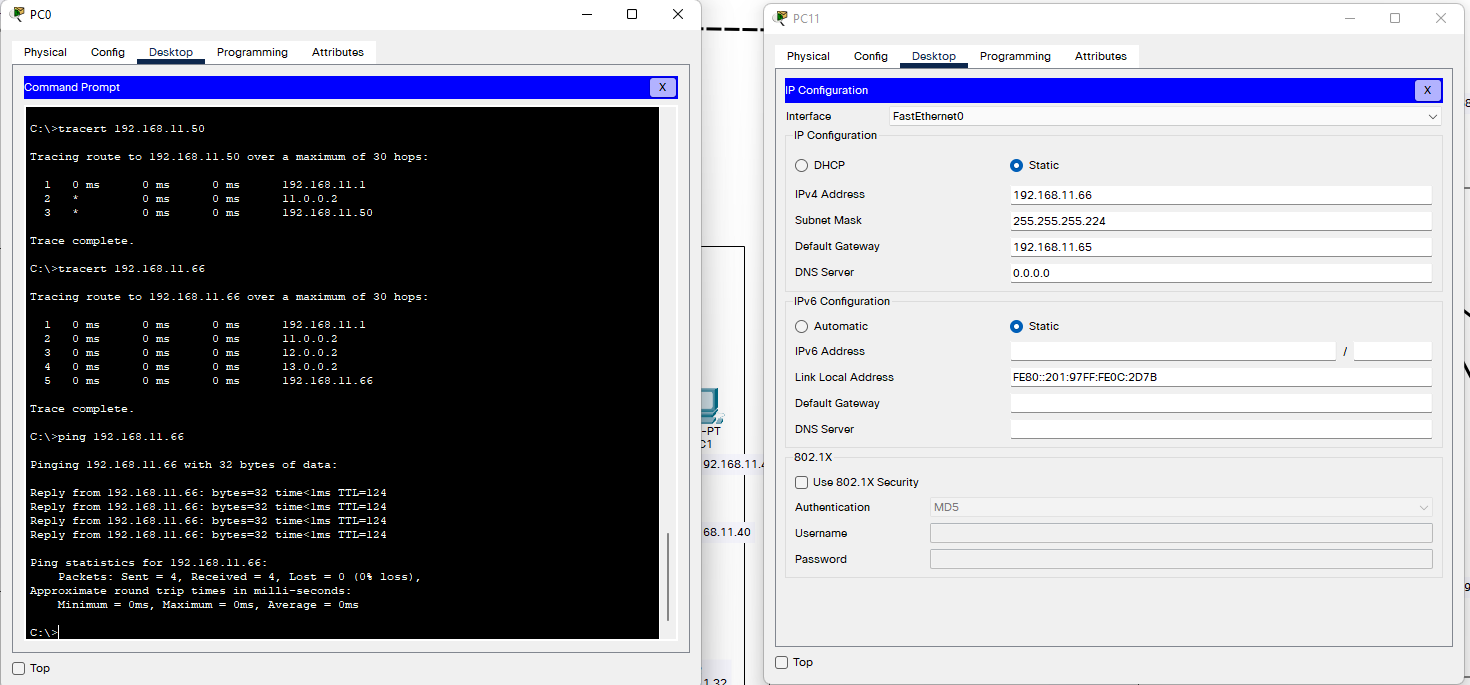


Рисунок 7 - Связь между PC0 и PC11

Теперь проверим хосты PC0 и PC17 (см. Рисунок 8).

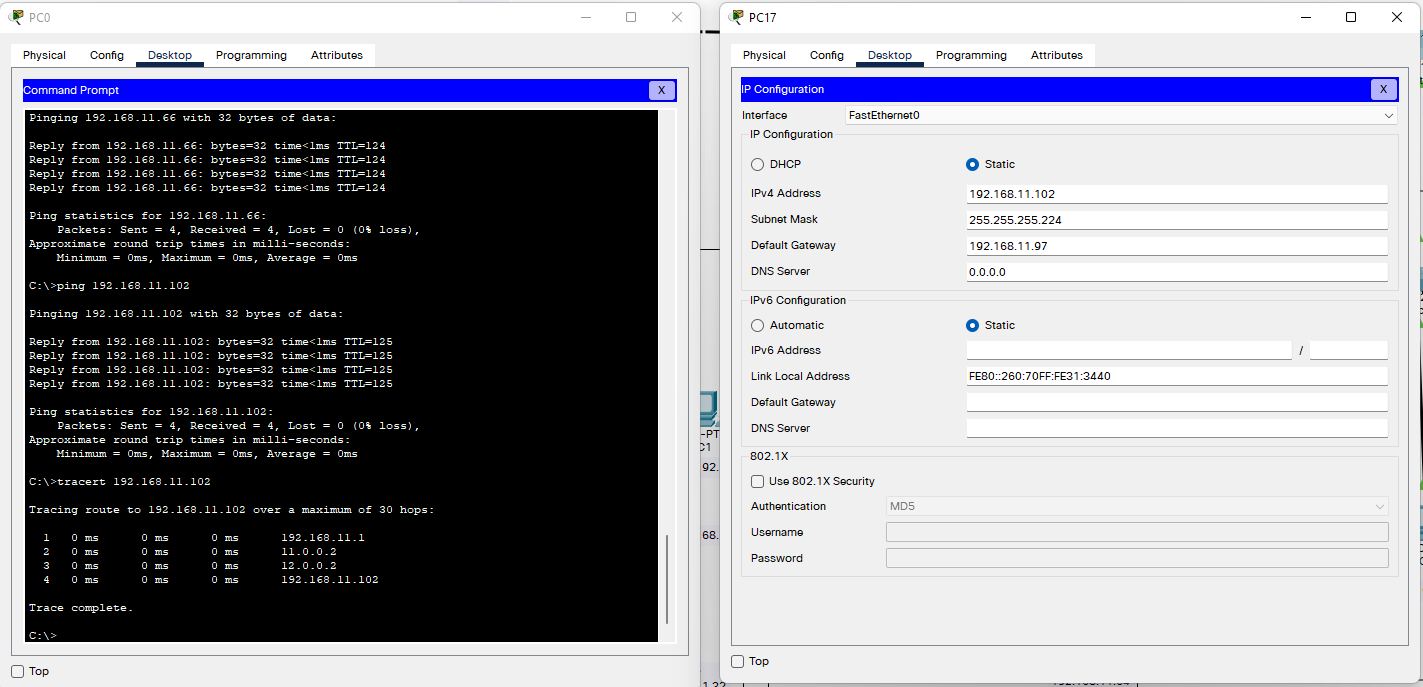


Рисунок 8 - Связь между PC0 и PC17

Между всеми хостами настроена связь.

Теоретический вопрос

HTTP (HyperText Transfer Protocol) — это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете. HTTP является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером (web-browser). Полученный итоговый документ будет (может) состоять из различных поддокументов, являющихся частью итогового документа: например, из отдельно полученного текста, описания структуры документа, изображений, видеофайлов, скриптов и многого другого.

Протокол HTTP предполагает использование клиент-серверной структуры передачи данных. Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом.

Каждый запрос (англ. request) отправляется серверу, который обрабатывает его и возвращает ответ (англ. response). Между этими запросами и ответами как правило существуют многочисленные посредники, называемые прокси, которые выполняют различные операции и работают как шлюзы или кэш, например.

Задача, которая традиционно решается с помощью протокола HTTP — обмен данными между пользовательским приложением, осуществляющим доступ к веб-ресурсам (обычно это веб-браузер) и веб-сервером. На данный момент именно благодаря протоколу HTTP обеспечивается работа Всемирной паутины.

Также HTTP часто используется как протокол передачи информации для других протоколов прикладного уровня, таких как SOAP, XML-RPC и WebDAV. В таком случае говорят, что протокол HTTP используется как «транспорт».

API многих программных продуктов также подразумевает использование HTTP для передачи данных — сами данные при этом могут иметь любой формат, например, XML или JSON.

Как правило, передача данных по протоколу HTTP осуществляется через TCP/IP-соединения. Серверное программное обеспечение при этом обычно использует TCP-порт 80, хотя может использовать и любой другой.

Список использованной литературы

1. Пример расчета количества хостов и подсетей на основе IP-адреса и маски [Электронный ресурс]. URL: https://tinyurl.com/284csetr (Дата обращения: 24.12.2021).
2. Протокол HTTP [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/HTTP (Дата обращения: 24.12.2021).
3. Количество хостов и подсетей [Электронный ресурс]. URL: https://tinyurl.com/bdja8utp (Дата обращения: 24.12.2021).