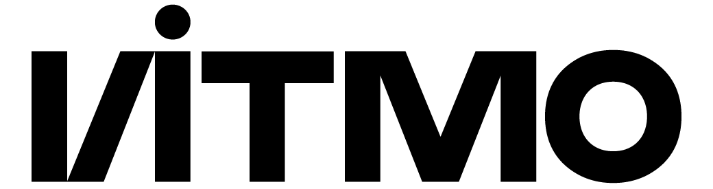
****

**Лабораторная работа №3   
«Компьютерные сети с маршрутизаторами»**

*по дисциплине: Компьютерные сети*

Выполнил: Неграш Андрей, P33301

Преподаватель: Алиев Тауфик Измайлович

Санкт-Петербург, 2023

Оглавление

[1. Цель работы 3](#_Toc133207020)

[2. Постановка задачи 3](#_Toc133207021)

[3. Сеть с одним маршрутизатором 4](#_Toc133207022)

[3.1. Построение и настройка сети с маршрутизатором 4](#_Toc133207023)

[3.2. Тестирование сети 5](#_Toc133207024)

[3.2.1. Передача сообщений с помощью UDP 5](#_Toc133207025)

[3.2.2. Передача сообщений с помощью TCP 6](#_Toc133207026)

[4. Сеть с двумя маршрутизаторами 7](#_Toc133207027)

[4.1. Построение и настройка сети с двумя маршрутизаторами 7](#_Toc133207028)

[4.2. Тестирование сети 8](#_Toc133207029)

[4.2.1. Передача сообщений с помощью UDP 8](#_Toc133207030)

[4.2.2. Передача сообщений с помощью TCP 9](#_Toc133207031)

[5. Сеть с тремя маршрутизаторами 10](#_Toc133207032)

[5.1. Построение и настройка сети с тремя маршрутизаторами 10](#_Toc133207033)

[5.2. Тестирование сети 12](#_Toc133207034)

[5.2.1. Передача сообщений по протоколу UDP 12](#_Toc133207035)

[5.3. Настройка динамической маршрутизации по протоколу RIP 14](#_Toc133207036)

[5.4. Настройка автоматического получения сетевых настроек по протоколу DHCP 15](#_Toc133207037)

[Вывод 17](#_Toc133207038)

# Цель работы

В процессе выполнения данной лабораторной работы необходимо:

* построить модели компьютерных сетей, представляющих собой несколько подсетей, объединенных в одну автономную сеть, в соответствии с заданными вариантами топологий;
* выполнить настройку сети при статической маршрутизации, заключающуюся в присвоении IP-адресов интерфейсам сети и ручном заполнении таблиц маршрутизации;
* промоделировать работу сети при использовании динамической маршрутизации на основе протокола RIP и при автоматическом распределении IP-адресов на основе протокола DHCP;
* выполнить тестирование построенных сетей путем проведения экспериментов по передаче данных на основе протоколов UDP и TCP;
* проанализировать результаты тестирования и сформулировать выводы об эффективности сетей с разными топологиями;
* сохранить разработанные модели локальных сетей для демонстрации процессов передачи данных при защите лабораторной работы.

# Постановка задачи

При выполнении данной лабораторной работы были использованы сети из предыдущей работы, а именно сеть №1 из 4 компьютеров, соединённых концентратором, сеть №2 из 2 компьютеров, соединённых коммутатором и сеть №3 из 3 компьютеров, соединённых коммутатором.

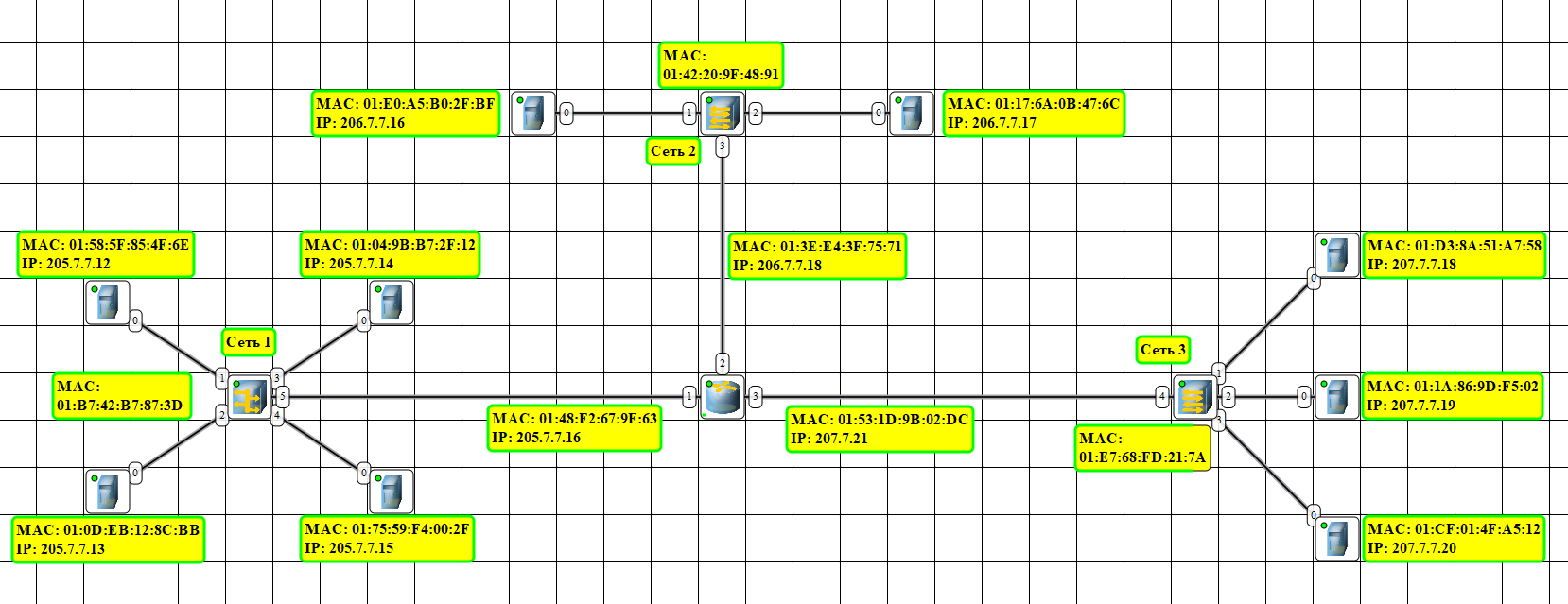
Пул IP-адресов для разных сетей:

Сеть №1: 205.7.7.12 – 205.7.7.16  
Сеть №2: 206.7.7.16 – 205.7.7.18  
Сеть №3: 207.7.7.18 – 207.7.7.21

# Сеть с одним маршрутизатором

## Построение и настройка сети с маршрутизатором

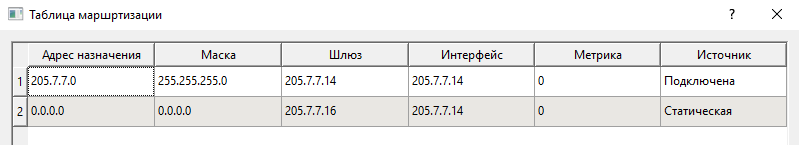
Свяжем созданные сети при помощи одного маршрутизатора по схеме «Звезда», назначим IP-адреса из заданного множества, а также выведем актуальные IP- и MAC-адреса на схему для наглядности:



Все IP-адреса в сетях у нас имеют класс C, поэтому маски адресов одинаковы и равны *255.255.255.0*.

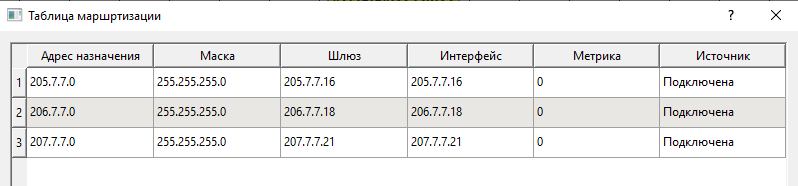
После настройки всех адресов проанализируем содержимое таблиц маршрутизации.

Таблица маршрутизации компьютеров в любой сети выглядит следующим образом (на скриншоте таблица компьютера 3 в сети №1):



В таблице присутствует две записи, в которой содержится IP-адрес подсети назначения, маска сети, шлюз, интерфейс (локальный адрес, по которому достигается шлюз), метрика и текущий статус.

Таблица маршрутизации на маршрутизаторе выглядит так:

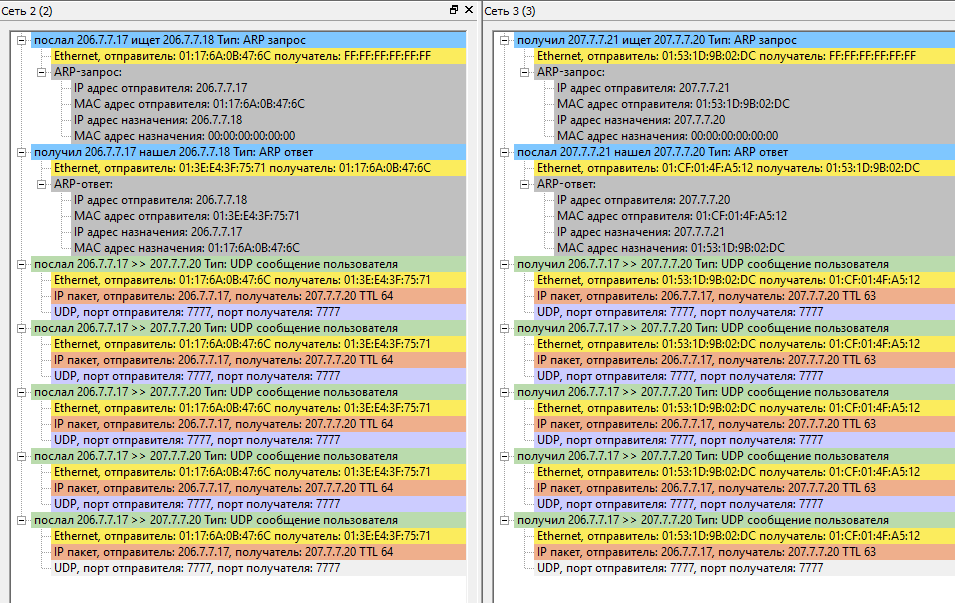


В данной таблице присутствуют записи с 3 IP-адресами подсетей, которые подключены к данному маршрутизатору.

## Тестирование сети

### Передача сообщений с помощью UDP

Для тестирования протокола UDP мы отправим сообщение размером 5 килобайт с компьютера 2 в сети №2 на компьютер 3 в сети №3. Журнал передачи выглядит следующим образом:

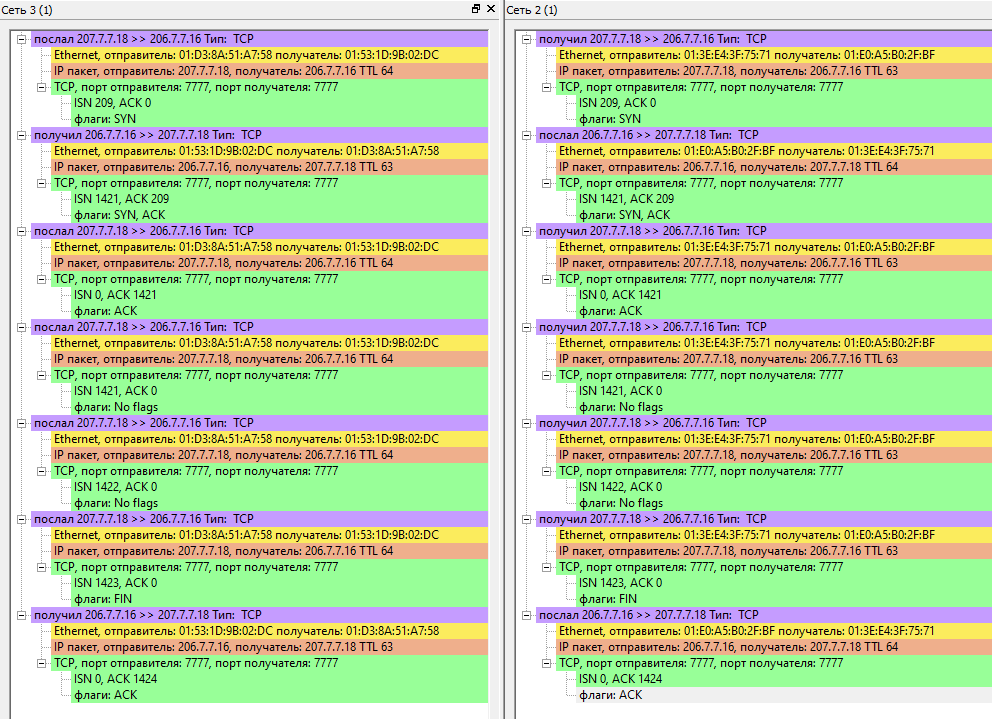


Алгоритм передачи выглядит следующим образом:

1. MAC-адрес получателя в ARP-таблице отправителя не найден, поэтому отправляется ARP-запрос, который ищет маршрутизатор, т.к. IP-адрес получателя находится в другой сети и значит нужно передать данные через шлюз по умолчанию, которым является маршрутизатор.
2. Маршрутизатор посылает положительный ответ отправителю со своим MAC-адресом, и отправитель сразу же отправляет 5 пакетов сообщения, каждый из которых несёт 1 килобайт сообщения. На этом работа отправителя завершена.
3. Когда маршрутизатор получил пакеты с сообщением, он посылает широковещательный ARP-запрос в сеть №3, и компьютер 3, который имеет искомый IP-адрес, отправляет положительный ARP-ответ маршрутизатору.
4. После получения ARP-ответа маршрутизатор посылает 5 пакетов с сообщениями конечному получателю и таким образом сообщение оказывается у получателя.

### Передача сообщений с помощью TCP

Для тестирования протокола TCP мы отправим сообщение размером 3 килобайта с компьютера 1 в сети №3 на компьютер 1 в сети №2. Журнал передачи выглядит следующим образом:



Алгоритм передачи выглядит следующим образом:

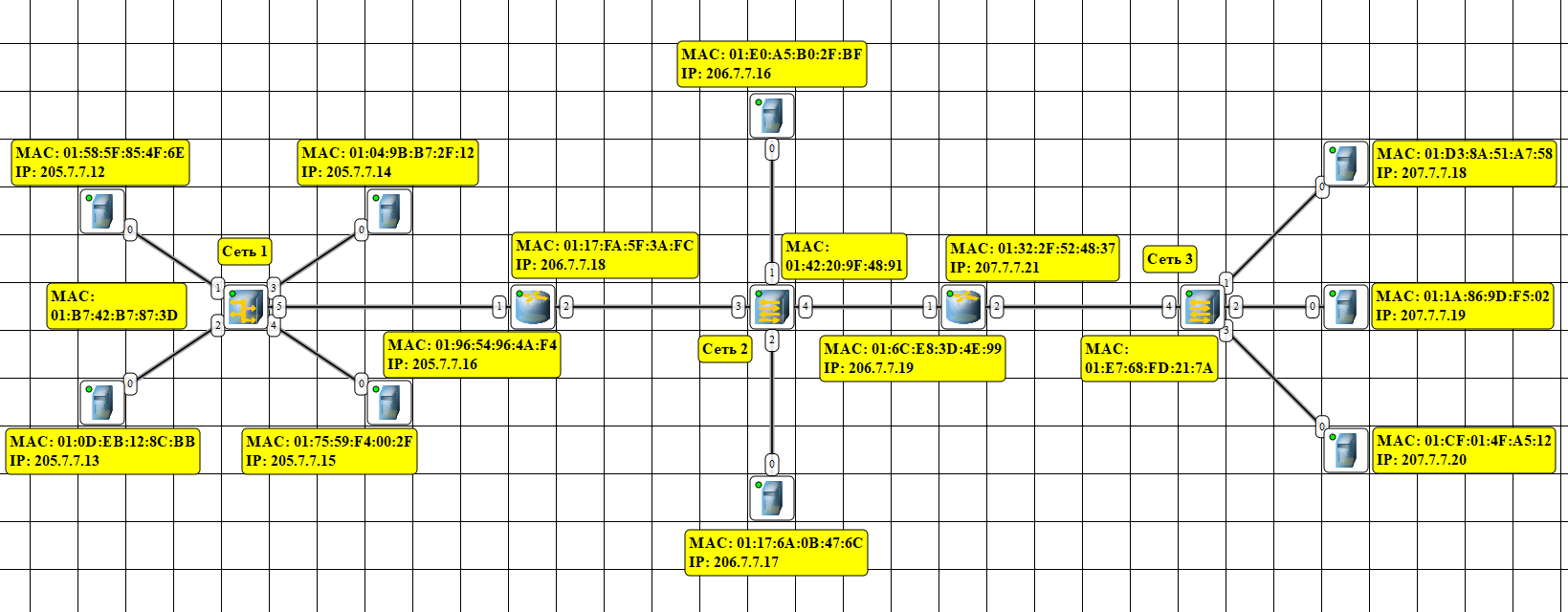
1. Если в ARP-таблице отправителя нет необходимой записи, то он отправит ARP-запрос. Однако у нас ARP-таблица заполнена нужными данными для более детального рассмотрения самой передачи по TCP.
2. Отправитель посылает на маршрутизатор TCP-пакет с флагом синхронизации и порядковым номером (стартовым битом). Маршрутизатор в свою очередь пересылает пакет данных с нужным MAC-адресом получателю.
3. Получатель принимает пакет с флагом синхронизации и посылает ответ с номером подтверждения получения кадра.
4. Отправитель через маршрутизатор получил подтверждение синхронизации и считает надёжным установленное соединение. Поэтому он отправляет 1 пакет с подтверждением получения подтверждения и 3 пакета с сообщением, последний из которых будет иметь флаг окончания сообщения (прекращения передачи).
5. Получатель получает информацию о получении подтверждения, получает в целости 3 пакета с сообщением и отправляет подтверждение получения финального пакета, передав его номер в качестве номера подтверждения.
6. Отправитель получает подтверждение успешной передачи данных и на этом процесс передачи заканчивается.

Основным отличием передачи данных при помощи протоколов UDP и TCP по сравнению с сетями без маршрутизаторов заключается в том, что в заголовке пакетов отправитель указывает MAC-адрес не конечного получателя, а MAC-адрес маршрутизатора, который в свою очередь осуществит пересылку пакета данных в нужную сеть получателю с нужным MAC-адресом.

# Сеть с двумя маршрутизаторами

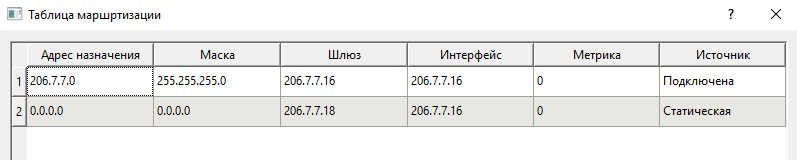
## Построение и настройка сети с двумя маршрутизаторами

Свяжем созданные сети при помощи двух маршрутизатора по схеме «Последовательное соединение» и обновим данные IP- и MAC-адресов для новых маршрутизаторов:



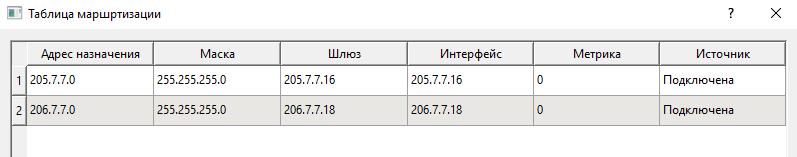
После настройки всех адресов проанализируем содержимое таблиц маршрутизации.

Таблица маршрутизации компьютеров в любой сети выглядит следующим образом (на скриншоте таблица компьютера 1 в сети №2):



В таблице (как и в предыдущем пункте) присутствует две записи, в которой содержится IP-адрес подсети назначения, маска сети, шлюз, интерфейс (локальный адрес, по которому достигается шлюз), метрика и текущий статус.

Таблица маршрутизации каждого из маршрутизаторов выглядит следующим образом:



В таблице содержится по 2 записи, поскольку каждый маршрутизатор подключен одновременно к двум сетям.

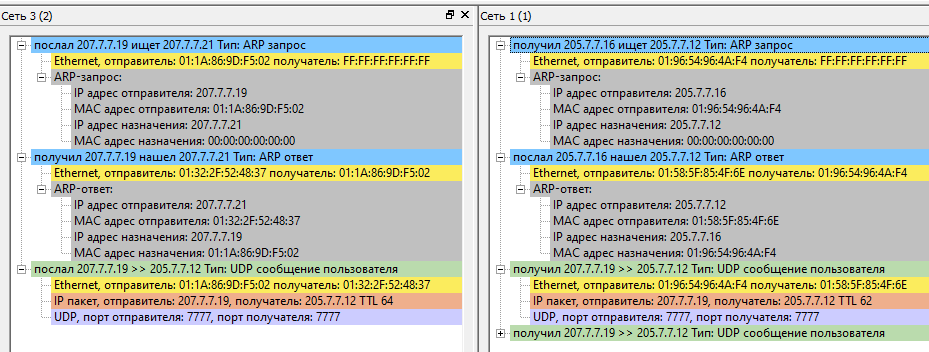
## Тестирование сети

### Передача сообщений с помощью UDP

Для тестирования протокола UDP мы отправим сообщение размером 1 килобайт с компьютера 2 в сети №3 на компьютер 1 в сети №1.

Для корректной передачи при такой схеме (когда сеть №3 и сеть №1 не пересекаются в одном маршрутизаторе), необходимо настроить таблицу маршрутизации на каждом из маршрутизаторов, добавив запись с необходимыми данными. Так в маршрутизаторе 1 (соединяющем сеть №1 и сеть №2, скриншот в предыдущем пункте) должна появиться третья запись с адресом назначения 207.7.7.0, маской 255.255.255.0, шлюзом 206.7.7.19 и интерфейсом 206.7.7.18. После добавления этих записей в таблице маршрутизации осуществим передачу.

Журнал передачи выглядит следующим образом:



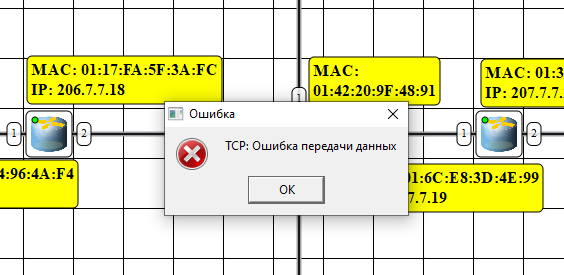
Алгоритм передачи выглядит следующим образом:

1. Отправитель в сети №3 посылает широковещательный ARP-запрос в надежде найти маршрутизатор. Маршрутизатор отправляет положительный ответ со своим MAC-адресом.
2. Отправитель в сети №3 отправляет пакет с данными на маршрутизатор.
3. После получения пакета с данными первый маршрутизатор смотрит в таблицу маршрутизации и отправляет широковещательный ARP-запрос в сети №2, желая узнать MAC-адрес второго маршрутизатора в этой сети, так как согласно таблице маршрутизации, сеть с нужным IP находится у него. Второй маршрутизатор отправляет свой MAC-адрес вместе с положительным ARP-ответом.
4. После получения искомого MAC-адреса осуществляется пересылка пакета данных от одного маршрутизатора другому.
5. Второй маршрутизатор уже обращается к сети №1 с ARP-запросом, пытаясь найти конечного получателя. Компьютер 1, которому предназначается сообщение, высылает ARP-ответ.
6. Получив ARP-ответ и MAC-адрес искомого компьютера, маршрутизатор посылает ему пакет с сообщением. Однако из-за работы концентратора (рассылки пакетов всем подключенным устройствам) происходит зацикливание и данное UDP-сообщение приходит на Компьютер 1 повторно (и ещё многократно).

### Передача сообщений с помощью TCP

Для тестирования протокола TCP мы отправим сообщение размером 3 килобайта с компьютера 2 в сети №3 на компьютер 1 в сети №1.

При попытке передачи данных, когда пакет с флагом синхронизации доходит до сети №1, а если быть точнее, то до концентратора в этой сети – количество передаваемых пакетов внутри сети начинает кратно увеличиваться и пересылаться внутри сети, из-за чего возникают коллизии. Это приводит к тому, что до маршрутизатора не доходит ответ Компьютера 1 об установке надёжного соединения и возникает ошибка передачи данных:



Данную проблему полностью могла бы решить замена концентратора на коммутатор, однако согласно заданию необходимо использовать именно концентратор. Поэтому констатирую, что передача данных в сеть №1 по протоколу TCP невозможна.

Отличия протоколов UDP и TCP на данном примере особенно разительны. Несмотря на одинаково возникающую в процессе передачи данных коллизию и зацикливание в сети с концентратором, протокол UDP смог доставить полезные данные с сообщением конечному получателю. Протокол TCP же не смог установить надёжное соединение, посылая лишь служебные пакеты, и таким образом не смог доставить ни одного байта полезных данных.

# Сеть с тремя маршрутизаторами

## Построение и настройка сети с тремя маршрутизаторами

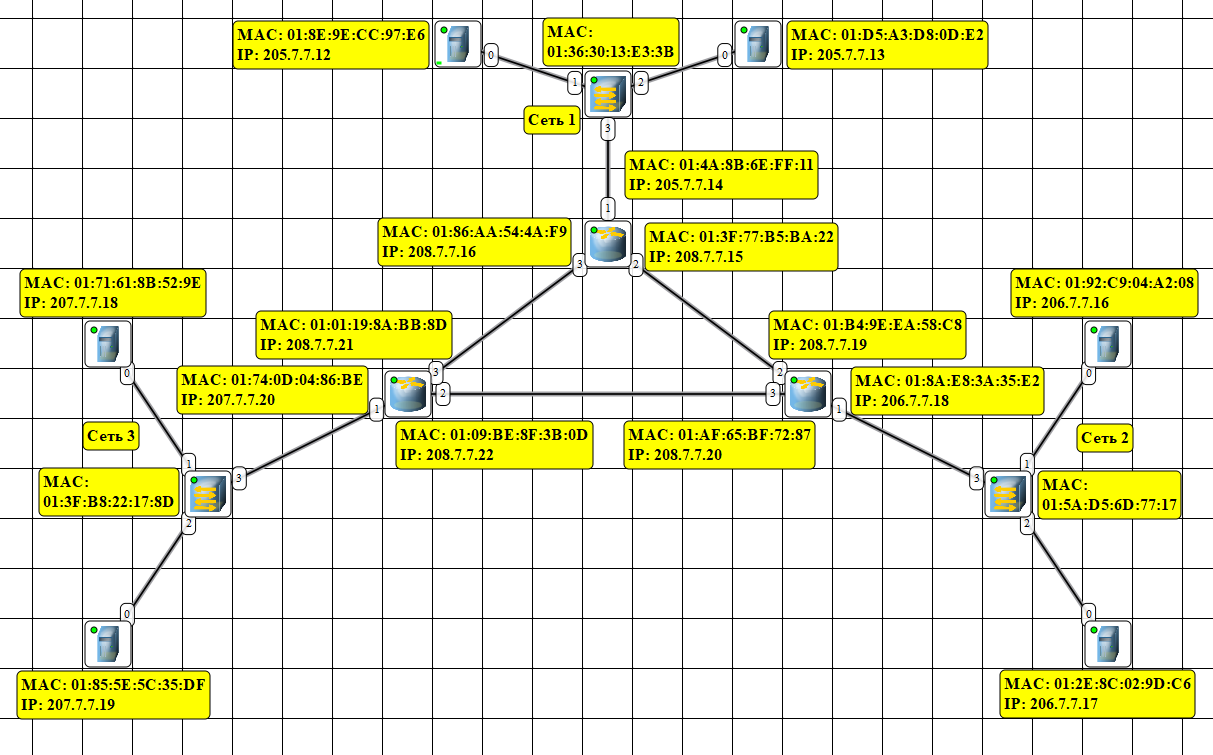
Для начала рассмотрим все предложенные схемы в виде таблицы-сравнения достоинств и недостатков каждой схемы:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Схема** | **Достоинства** | **Недостатки** |
| В3 |  | + есть 1 альтернативный путь у каждой сети  + лёгкая настройка | - мало альтернативных путей  - маршрутизатор содержит данные о двух сетях сразу |
| В4 |  | + есть 2 альтернативных пути у каждой сети  + каждый маршрутизатор содержит информацию только об одной сети и других маршрутизаторах | - трудная настройка |
| В5 |  | + есть минимум 4 альтернативных пути у каждой сети  + сеть №2 и сеть №3 могут совершать обмен без маршрутизатора | - трудная настройка (труднее, чем в предыдущем варианте) |
| В6 |  | + есть минимум 4 альтернативных пути у каждой сети | - самая сложная настройка  - маршрутизатор №3 является узким местом |

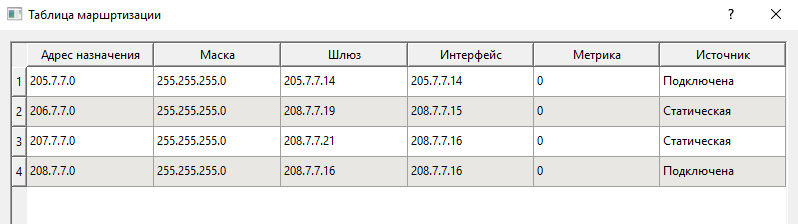
Выполним задание на основе схемы под номером В4, поскольку наличие 2 альтернативных путей и меньшая заполненность ARP-таблиц на маршрутизаторах нивелируют преимущество перед схемой В3 в сложности настройки.

В рамках выполнения данной части работы мы откажемся от использования концентратора и во всех сетях подключать к маршрутизаторам будем коммутаторы. Также каждая из сетей будет представлять собой 2 компьютера, соединённых 1 коммутатором для сокращения ненужных устройств на схеме, так как основной задачей является рассмотрение топологий.

Построенная схема выглядит следующим образом:



В таблицы маршрутизации добавим необходимые записи для корректной передачи данных между сетями. Таблицы после настройки выглядят таким образом:

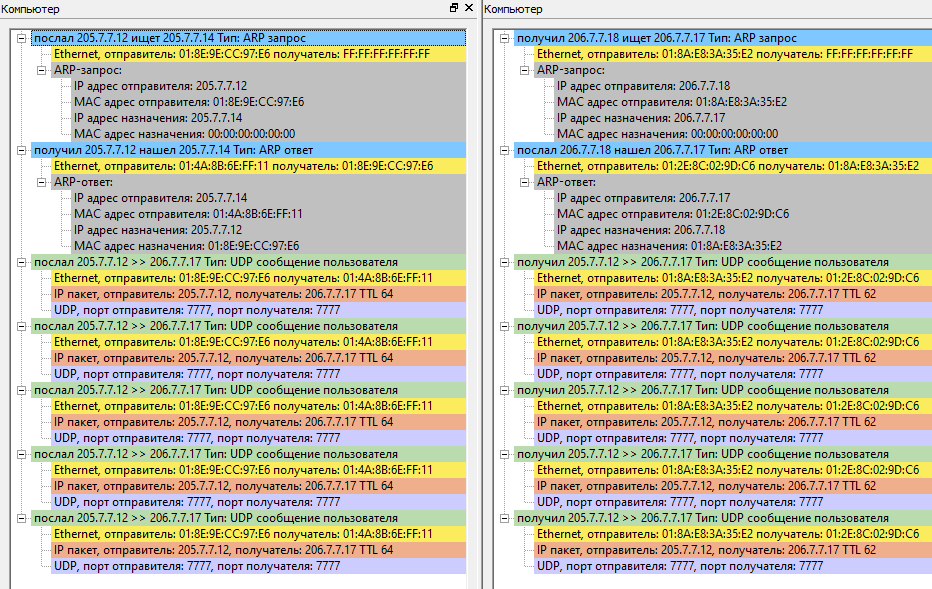


Были добавлены статические записи для пересылки данных между маршрутизаторами с использованием дополнительной сети с пулом IP-адресов 208.7.7.15 – 208.7.7.22.

## Тестирование сети

### Передача сообщений по протоколу UDP

Для тестирования протокола UDP мы отправим сообщение размером 5 килобайт с компьютера 1 в сети №1 на компьютер 2 в сети №2. Журнал передачи выглядит следующим образом:



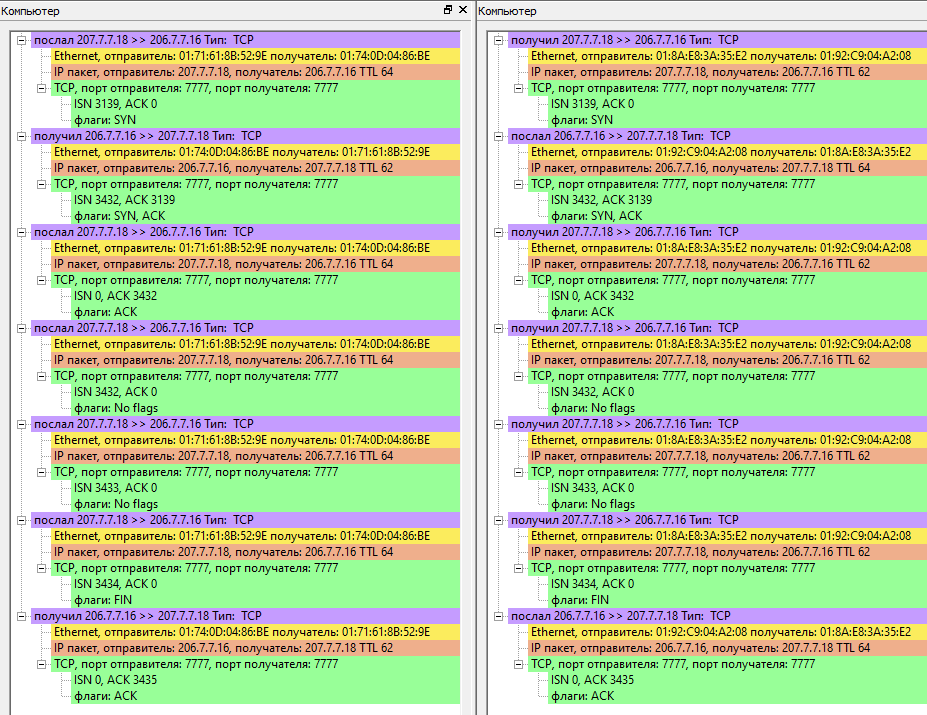
Алгоритм передачи выглядит следующим образом:

1. Компьютер 1 посылает ARP-запрос маршрутизатору и получает ARP-ответ.
2. Компьютер 1 отправляет сообщение из 5 кадров на маршрутизатор, на этом работа Компьютера 1 окончена.
3. Маршрутизатор после получения кадров сообщения посылает ARP-запрос среди маршрутизаторов, затем получает ответ и пересылает на нужный маршрутизатор сообщение.
4. Второй маршрутизатор после получения пакета с данными тоже посылает ARP-запрос, но уже во внутреннюю сеть.
5. Компьютер-адресат получает ARP-запрос, узнаёт в нём свой IP-адрес и посылает положительный ARP-ответ.
6. Маршрутизатор после получения положительного ARP-ответа отправляет сообщение из 5 кадров непосредственно компьютеру-получателю. На этом процесс передачи успешно завершается.

Для корректной передачи данных между маршрутизаторами была сформирована отдельная сеть, благодаря которой маршрутизация происходит по кратчайшему маршруту, ведь в заданной схеме каждый маршрутизатор соединён с каждым.

* + 1. Передача сообщений по протоколу TCP

Для тестирования протокола TCP мы отправим сообщение размером 3 килобайта с компьютера 1 в сети №3 на компьютер 1 в сети №2. Журнал передачи выглядит следующим образом:



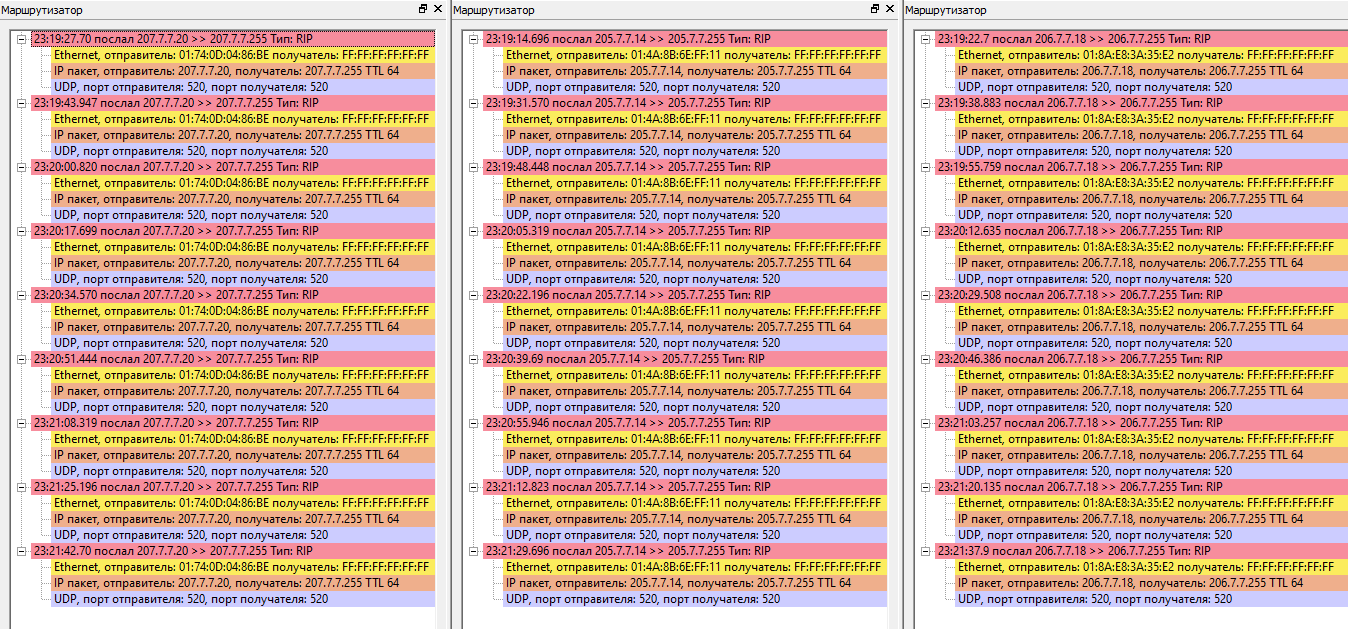
Для того, чтобы не возникало ошибки TCP, необходимо в настройках NetEmul или указать количество повторных отправок для TCP равному 2 или более, или там же время ожидания увеличить до 120 секунд. Данная проблема возникает из-за того, что на каждом этапе передачи при установке соединения отправляются ARP-запросы, которые тратят время и ответ не успевает прийти до истечения таймаута. В примере выше отправка идёт повторно и в ARP-таблицах уже содержатся необходимые записи. Их заполнение аналогично описанному в пункте про протокол UDP.

Алгоритм передачи:

1. Компьютер-отправитель посылает служебное сообщение об установке соединения на маршрутизатор, оттуда оно пересылается следующему маршрутизатору и доходит до адресата.
2. Получатель в ответ отправляет подтверждение соединения, которое проходит обратный маршрут через 2 маршрутизатора.
3. Отправитель после установки надёжного соединения посылает 3 кадра сообщения, которые также идут через 2 маршрутизатора и приходят к адресату.
4. Получатель видит флаг FIN и шлёт подтверждение получения всех кадров. На этом соединение разрывается.

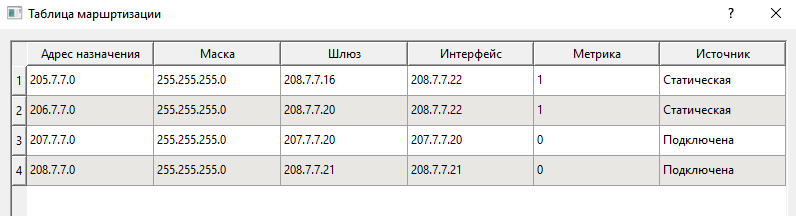
## Настройка динамической маршрутизации по протоколу RIP

Установим на все устройства, которые работают с сетью (компьютеры и маршрутизаторы), программу RIP. После выполнения этого шага откроем журналы всех трёх маршрутизаторов и отследим передачу пакетов. Выглядит она следующим образом:



На данном скриншоте мы видим, что протокол RIP отправляет UDP-запросы к известным ему адресам, а по получаемым данным на основе TTL выставляет значение метрики в таблице маршрутизации.

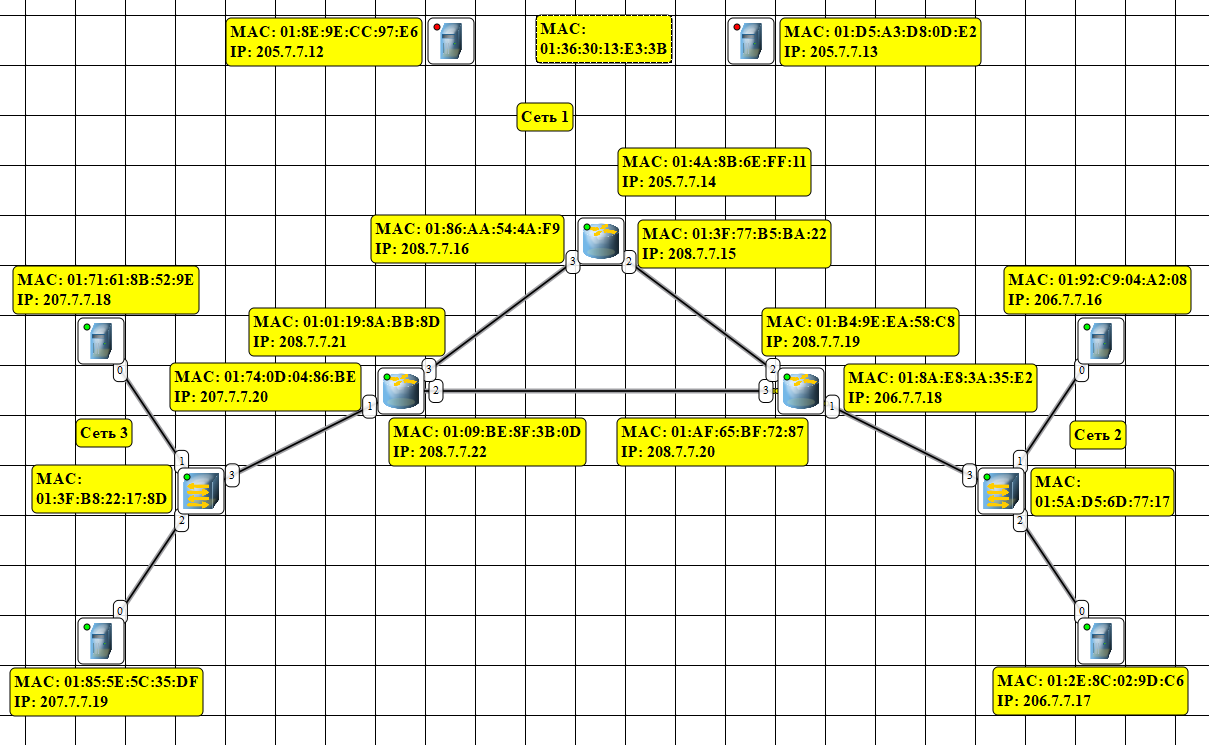
Таблицы маршрутизации заполнены следующим образом:



По стандарту протокол RIP отправляет данные каждые 30 секунд. Однако на нашем скриншоте это происходит каждые 16 секунд. Причина кроется в том, что я ускорил анимацию передачи в настройках NetEmul, и именно это повлияло на уменьшение интервала ожидания.

В таблице маршрутизации появились записи о подключенных маршрутизаторах с метриками-расстояниями до них.

Смоделируем ситуацию, при которой коммутатор в сети №1 вышел из строя:

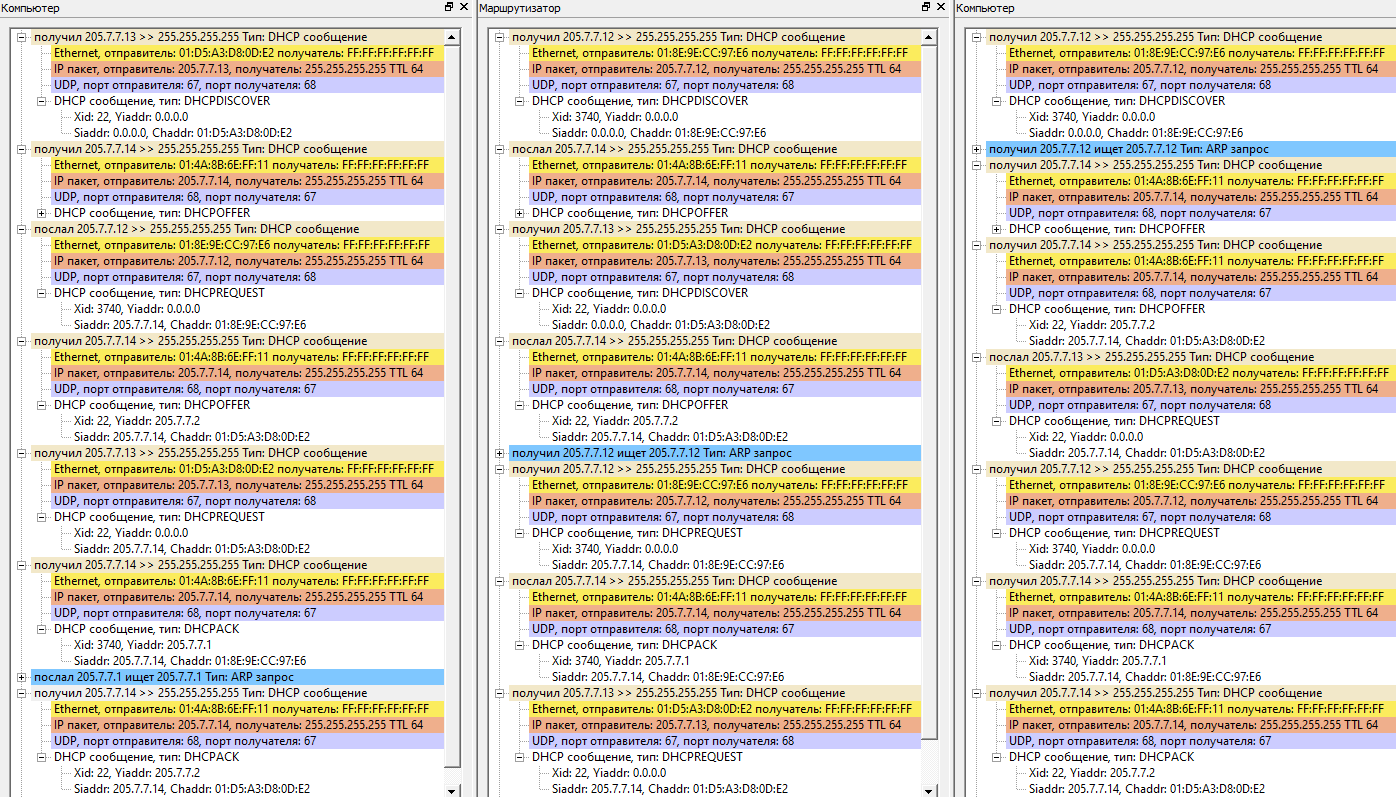


В процессе функционирования сети в таблице маршрутизации исчезла запись в маршрутизаторах о наличии подключения к этой сети. Передача данных в ту сеть завершается на любом из маршрутизаторов, куда приходит запрос от компьютера, так как он не знает о существовании данного диапазона IP-адресов.

## Настройка автоматического получения сетевых настроек по протоколу DHCP

В процессе настройки были установлены программы DHCP сервера на все маршрутизаторы и программы DHCP клиента на все компьютеры.

Запустим моделирование и отследим передачи сообщений настройки при помощи журналов. Скриншот журнала представлен здесь:

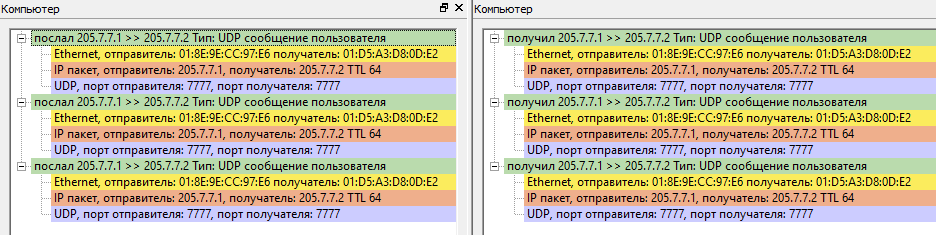


Протокол DHCP:

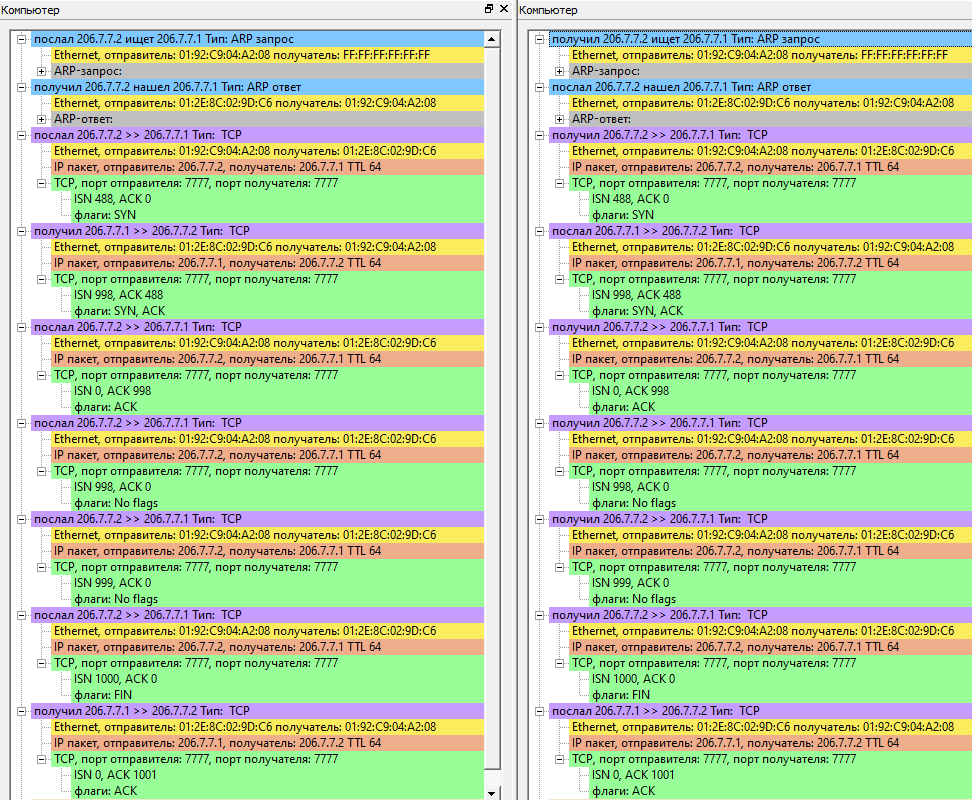
1. Компьютеры (DHCP-клиенты) отправляют сообщение DHCPDISCOVER в поисках DHCP-сервера (маршрутизатора)
2. Ответ с предлагаемым IP-адресом приходит сообщением DHCPOFFER. Это сообщение может быть отправлено сервером как адресно конкретному устройству, так и широковещательно.
3. После определения сервера клиентом, компьютер посылает DHCPREQUEST с подтверждением предложенного IP-адреса в DHCPOFFER. Примечательно, что DHCPREQUEST является широковещательным сообщением, с помощью него клиент сообщает о том, что данному MAC-адресу будет соответствовать данный IP.
4. Также есть сообщение DHCPACK, которым сервер отправляет уведомление о резервировании конкретного IP-адреса для определённого компьютера. Этот IP не должен расходиться с предложенным в рамках запроса DHCPOFFER.

После полной настройки сети попробуем передать пакеты данных между компьютерами и тем самым проверим работоспособность сети.

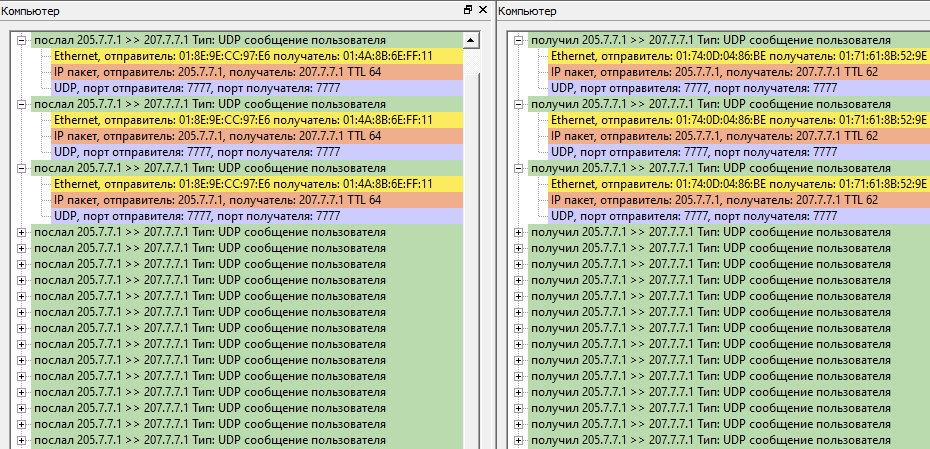
Тест UDP-протокола внутри одной подсети (сеть №1):



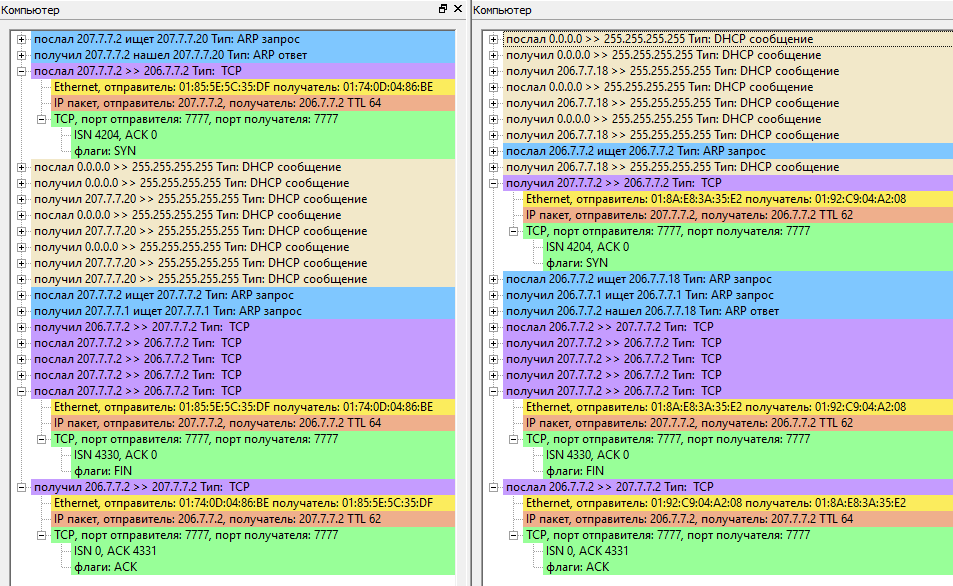
Тест TCP-протокола внутри одной подсети (сеть №2):



Тест UDP-протокола между компьютерами в сети №1 и сети №3:



Тест TCP-протокола между компьютерами в сети №3 и сети №2:



# Вывод

Итак, в процессе данной лабораторной работы я построил несколько видов компьютерных сетей с маршрутизатором, количество которых варьировалось в диапазоне от 1 до 3. Научился организовывать статическую маршрутизацию путём настройки таблиц маршрутизации и сетевого оборудования, а также рассмотрел работу протоколов динамической маршрутизации на примере протоколов RIP (Routing Information Protocol) и DHCP (Dynamic Host Configuration Protocol). Выполнение работы производилось в среде NetEmul на основе варианта к лабораторной работе №2.