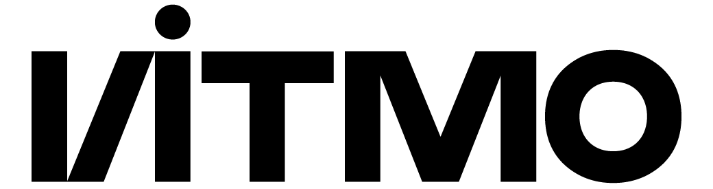
****

**Лабораторная работа №2 «Локальные сети»**

*по дисциплине: Компьютерные сети*

Выполнил: Неграш Андрей, P33301

Преподаватель: Алиев Тауфик Измайлович

Санкт-Петербург, 2023

Оглавление

[Цель работы 3](#_Toc131953025)

[Вычисление варианта 3](#_Toc131953026)

[Сеть 1. Локальная сеть с концентратором 4](#_Toc131953027)

[Построение 4](#_Toc131953028)

[Настройка 4](#_Toc131953029)

[Анализ таблиц 5](#_Toc131953030)

[Тестирование сети 6](#_Toc131953031)

[UDP 6](#_Toc131953032)

[TCP 6](#_Toc131953033)

[Сеть 2. Локальная сеть с коммутатором 8](#_Toc131953034)

[Построение сети и анализ таблиц 8](#_Toc131953035)

[Тестирование сети 9](#_Toc131953036)

[UDP 9](#_Toc131953037)

[TCP 10](#_Toc131953038)

[Многосегментная локальная сеть 11](#_Toc131953039)

[Построение 11](#_Toc131953040)

[Анализ лучшей топологии 12](#_Toc131953041)

[Топология «Звезда» с коммутатором 12](#_Toc131953042)

[Топология «Звезда» с концентратором 12](#_Toc131953043)

[Топология «Кольцо» 13](#_Toc131953044)

[Топология «Последовательное соединение» 13](#_Toc131953045)

[Тестирование сети 14](#_Toc131953046)

[UDP 14](#_Toc131953047)

[TCP 15](#_Toc131953048)

[Вывод 16](#_Toc131953049)

# Цель работы

Изучение принципов настройки и функционирования локальных сетей, построенных с использованием концентраторов и коммутаторов, а также процессов передачи данных на основе стека протоколов TCP/IP, с использованием программы моделирования компьютерных сетей NetEmul.

В процессе выполнения лабораторной работы (ЛР) необходимо:

* построить три модели локальной сети: с использованием концентратора, коммутатора и многосегментную сеть;
* выполнить настройку сети, заключающуюся в присвоении IP-адресов интерфейсам сети;
* выполнить тестирование разработанных сетей путем проведения экспериментов по передаче данных (пакетов и кадров) на основе протоколов UDP и TCP;
* проанализировать результаты тестирования и сформулировать выводы об эффективности смоделированных вариантов построения локальных сетей;
* сохранить разработанные модели локальных сетей для демонстрации процессов передачи данных при защите лабораторной работы.

# Вычисление варианта

Студент группы P33301 Неграш Андрей Владимирович (Ф=6, И=6, О=12, Н=1)  
Номер в группе по [списку](https://docs.google.com/spreadsheets/d/16fs6sPWdmyumKAR_xnn_9_pwsXmsu25m/edit#gid=130251719): 14

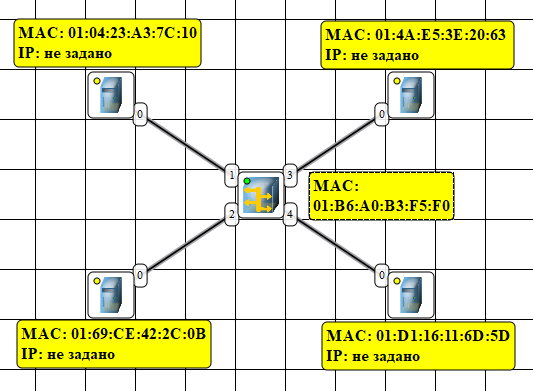
|  |  |  |  |
| --- | --- | --- | --- |
| **Количество компьютеров в …** | | | **Класс IP-адресов** |
| **сети 1** (*N1*) | **сети 2** (*N2*) | **сети 3** (*N3*) |
| 4 | 2 | 3 | C |

Класс C: (192+1+12).(6+1).(6+1).(6+6) = 205.7.7.12

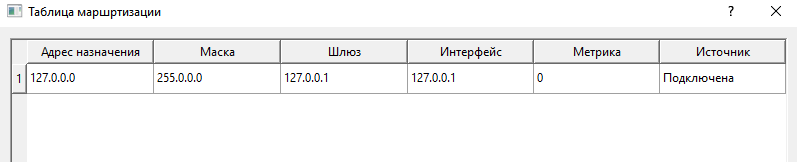
# Сеть 1. Локальная сеть с концентратором

## Построение

Сеть должна состоять из 4 компьютеров, объединённых в одну сеть с помощью концентратора. Получившаяся сеть представлена ниже:

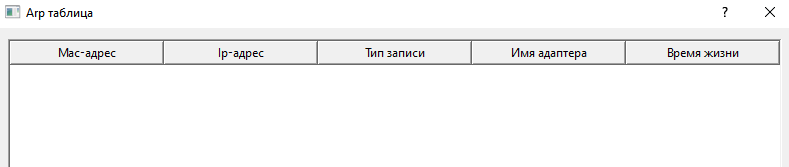


Таблицы маршрутизации всех 4 компьютеров выглядят так:



Это значит, что каждое из устройств изначально знает только свой IP на основе loopback, то есть имеет доступ только к внутреннему локальному адресу.

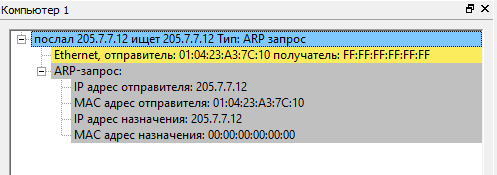
ARP-таблицы на компьютерах пока что пусты:



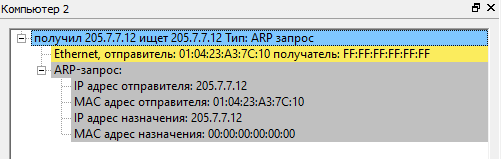
Это происходит из-за того, что мы пока что не настроили подключение и компьютер «не знает» о наличии сети и других компьютеров в ней, а значит не может предполагать ни IP, ни MAC-адреса.

## Настройка

В процессе настройки я присваиваю адреса с 205.7.7.12 по 205.7.7.15 всем компьютерам. При добавлении IP-адреса немедленно отправляется широковещательный ARP-запрос следующего вида:



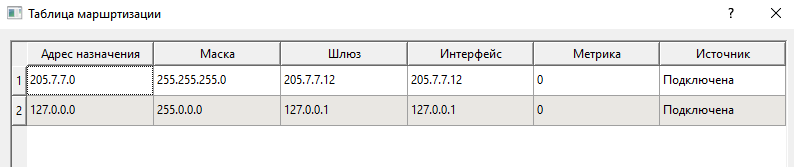
С помощью этого ARP-запроса компьютер хочет удостовериться, что в данной сети ещё нет устройства с таким же IP-адресом.  
Каждый из подключённых компьютеров получает этот ARP-запрос, но поскольку никто из них не имеет такого же адреса – ответ отправлен не будет.



## Анализ таблиц

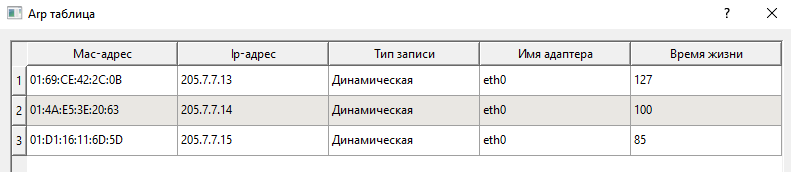
И в таблицах маршрутизации, и в ARP-таблицах произошли изменения.

Таблицы маршрутизации теперь имеют следующий вид:



Добавилась ещё одна запись, описывающая шлюз локальной сети, в которой находится компьютер.

ARP-таблицы перестали быть пустыми:

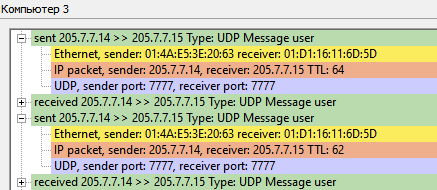


Теперь в них содержатся связки MAC-IP для всех соседей, то есть каждый компьютер знает адреса всех остальных своих «соседей» по сети. Однако со временем эти записи пропадают.

## Тестирование сети

### UDP

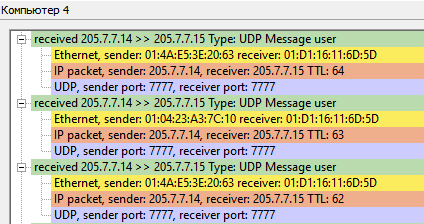
Передаваемые пакеты на устройстве-отправителе выглядят следующим образом:



Благодаря записям в ARP-таблицах заполнено поле Ethernet с MAC-адресами отправителя и получателя. Также указываются IP-адреса отправителя и получателя и их порты для протокола UDP.

Отдельно отмечу, почему после каждого отправленного пакета (sent) идёт полученный (received). Всё дело в концентраторе, работа которого сводится к отправлению полученного пакета всем подключенным устройствам, в том числе и обратно отправителю.

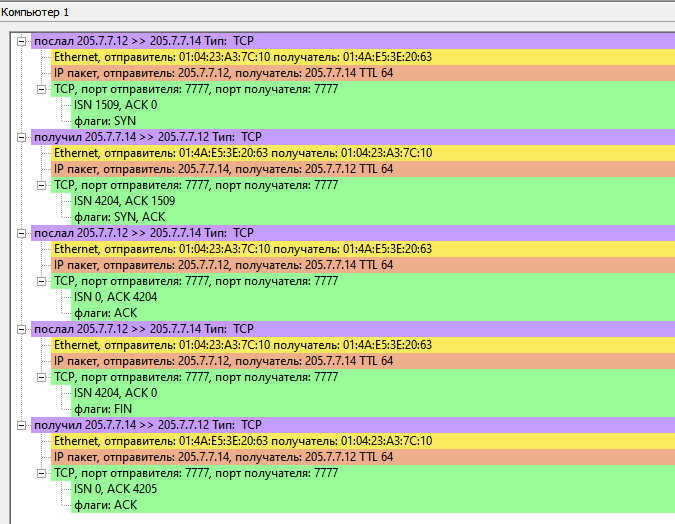
Получаемые пакеты выглядят так:



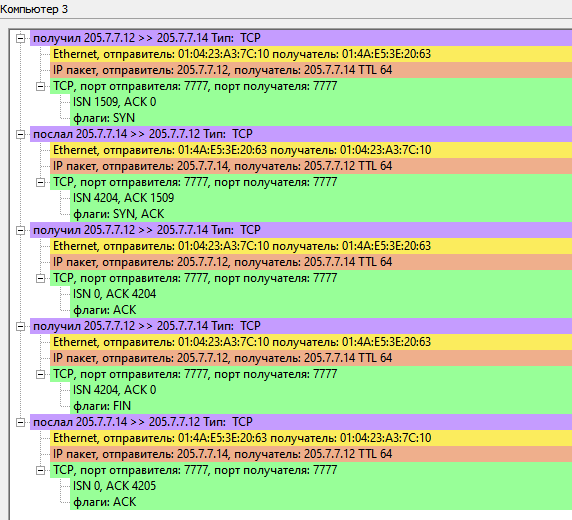
Их структура аналогична отправляемым, так что останавливаться отдельно не буду.

### TCP

Передаваемые пакеты на устройстве-отправителе выглядят следующим образом:



Ethernet-адреса и IP-адреса идентичны протоколу UDP. Секция протокола TCP имеет следующие отличия: появились поля «Порядковый номер» (ISN) и «Номер подтверждения» (ACK), а также флаги (SYN – синхронизация номеров последовательности, ACK – задействовано поле «Номер подтверждения», FIN – завершение соединения). Получаемые пакеты выглядят так:



Структура пакетов описана выше.

Основное отличие протокола TCP от UDP заключается в наличии подтверждения получения каждого пакета данных, и за счёт этого увеличение необходимых полей и количества посылок.

# Сеть 2. Локальная сеть с коммутатором

## Построение сети и анализ таблиц

Необходимо построить сеть из 2 компьютеров, объединённых в локальную сеть с помощью коммутатора (свитча):

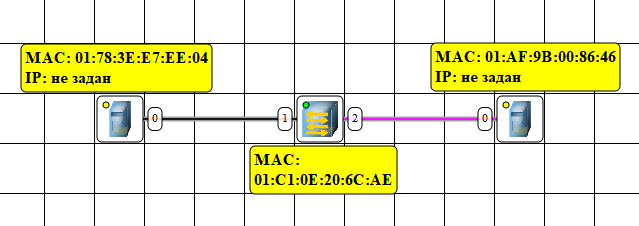
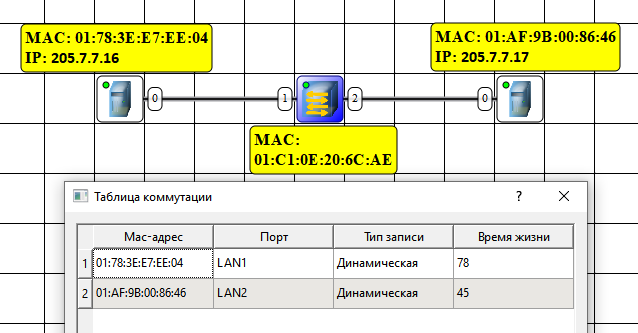


Таблица коммутации после соединения пустая:



Однако присвоив IP-адреса 205.7.7.16 и 205.7.7.17 компьютерам, таблица коммутации будет заполнена благодаря отправляемым ARP-запросам:



Как мы видим, таблица коммутации содержит 4 поля: MAC-адрес подключенного устройства, Порт коммутатора, в который подключено данное устройство, Тип записи, динамическая или статическая и Время жизни записи, которое измеряется в секундах. Максимальное значение «Времени жизни» составляет 5 минут (300 секунд).

Попробуем передать данные между компьютерами при пустой таблице коммутации.

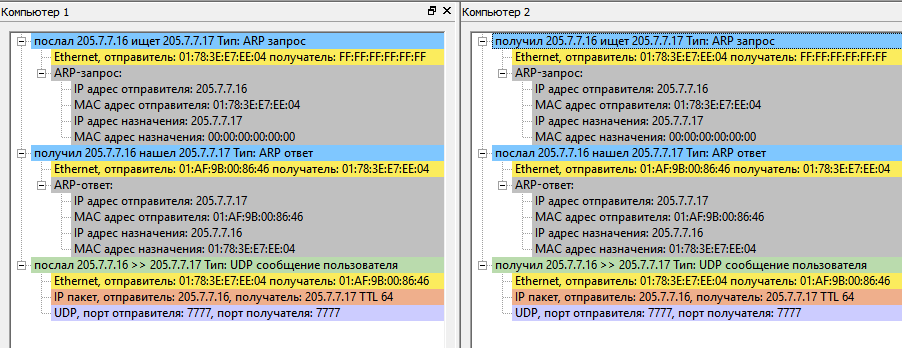
Заполнение таблицы коммутации происходит последовательно и на основании ARP-запроса и ARP-ответа. Сначала вносится запись о компьютере-отправителе, т.к. он посылает ARP-запрос, а затем данные компьютера, который отправил ARP-ответ.

В моей сети, состоящей из всего двух компьютеров, полное заполнение таблицы коммутации достигается отправкой одного ARP-запроса и получения ответа на него. Однако для исследования я добавил в сеть дополнительный третий компьютер, и понял, что для полного заполнения таблицы коммутации необходимо получить ARP-запрос или ARP-ответ от всех устройств.

Максимальное количество записей в таблице коммутации равно количеству портов в данном коммутаторе.

## Тестирование сети

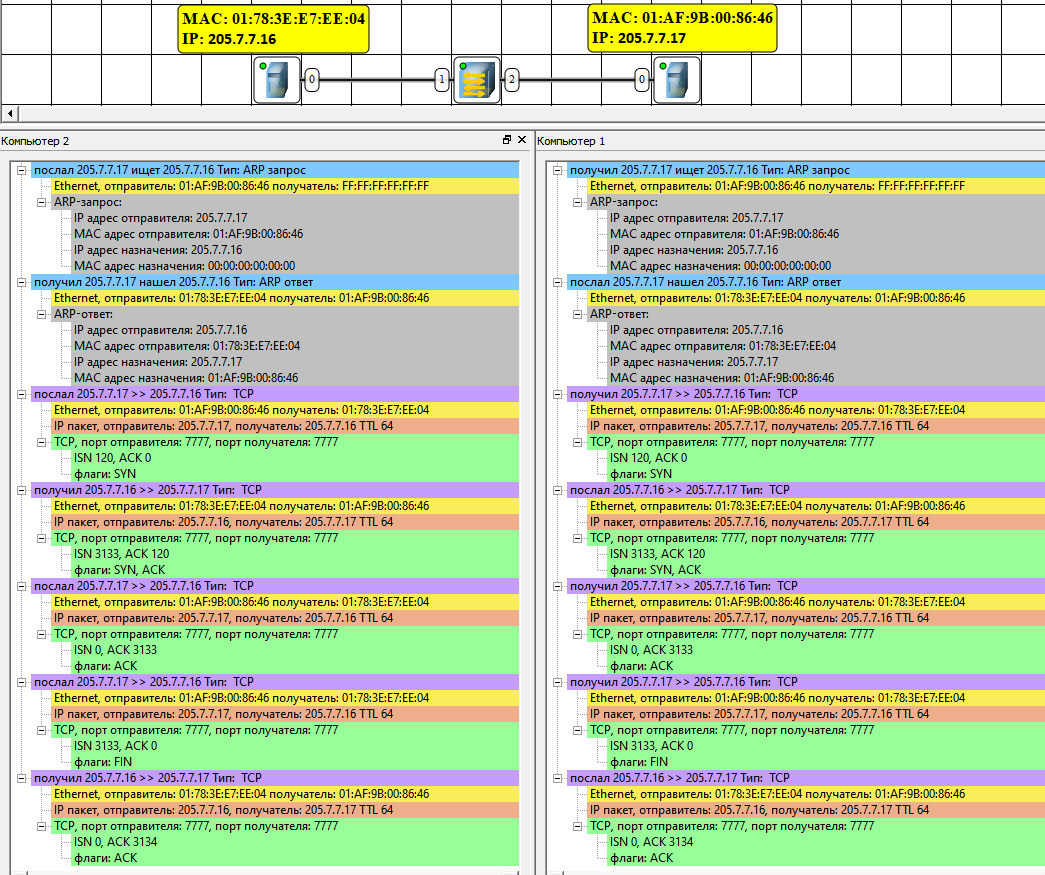
### UDP



Если в ARP-таблице компьютера-отправителя нет нужной записи, то будет сформирован ARP-запрос и ожидаться ARP-ответ. После получения которого будет отправлен сам пакет UDP-сообщения.

В ARP-таблице устройств, участвующих в передаче данных, добавляются записи друг о друге (или обнуляется время жизни записи). Таблицы маршрутизации на компьютерах остаются неизменными, таблица коммутации изменяется, как описано в предыдущем пункте.

### TCP



Если в ARP-таблице компьютера-отправителя нет нужной записи, то будет сформирован ARP-запрос и ожидаться ARP-ответ. После получения которого будут отправлены сами пакеты TCP-сообщения.

Передача данных со стороны компьютера-отправителя:

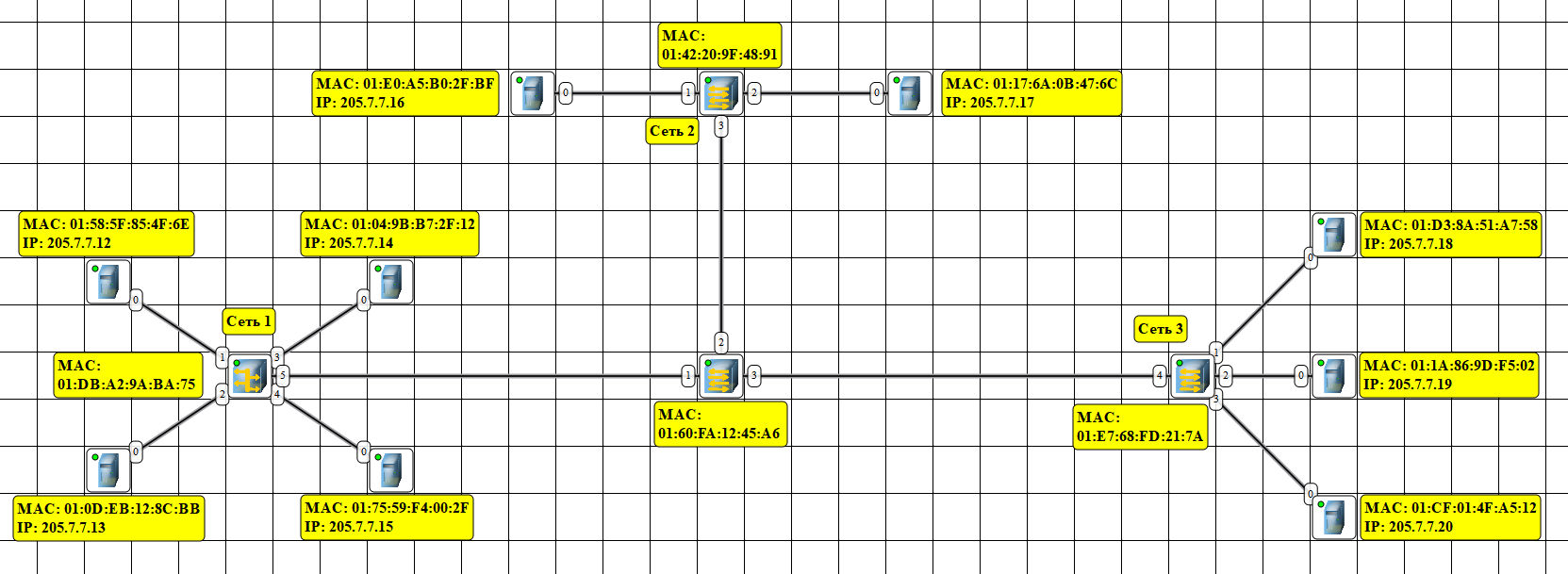
1. Отправлено служебное TCP-сообщение с флагом синхронизации и порядковым номером
2. Ответ приходит с флагом синхронизации и номером подтверждения
3. Отправка пакета-подтверждения с флагом номера подтверждения
4. Отправка пакета с данными и флагом финальной посылки (поскольку размер данных был 1 килобайт, что равно 1 сообщению)
5. Получение ответа с флагом номера подтверждения

Между TCP и UDP видна разница в количестве отправляемых пакетов для передачи сообщения одинакового размера. В сети с 2 компьютерами сложно увидеть разницу в работе коммутатора и концентратора, однако первый отправляет данные только необходимому адресату, а второй рассылает по всем подключенным абонентам.

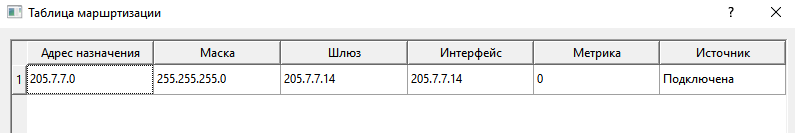
# Многосегментная локальная сеть

## Построение

Данная многосегментная локальная сеть состоит из 3 сегментов, 2 из которых были построены в предыдущих пунктах. Добавим третью сеть-сегмент согласно варианту и соединим сегменты при помощи коммутатора (топология «Звезда»):



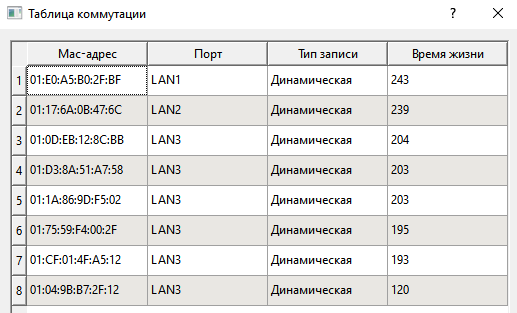
В таблицах маршрутизации всех компьютеров содержится по одной записи:



В ARP-таблицах каждого компьютера попадают записи обо всех устройствах во всех сегментах сети (если не произошло коллизий в концентраторе):



Таблицы коммутации выглядят так:



## Анализ лучшей топологии

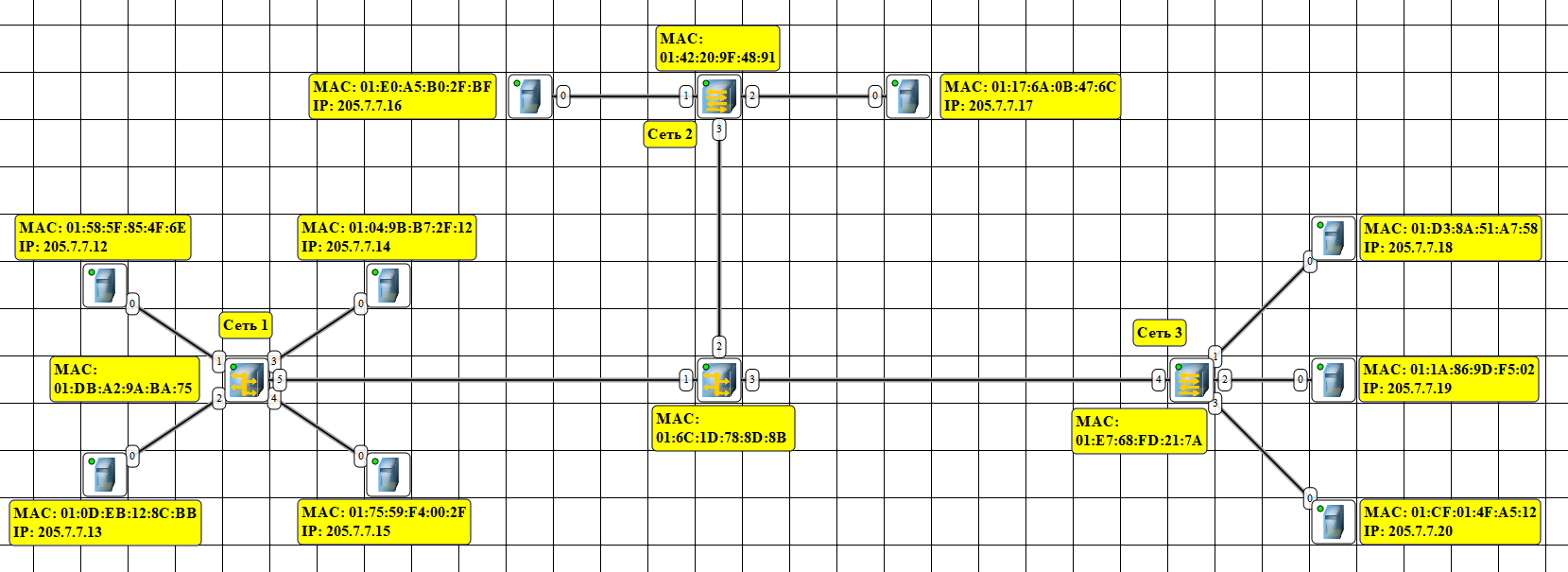
### Топология «Звезда» с коммутатором

Изображение выглядит как стол

Автоматически созданное описание

Изначальный вариант. Работает при пустых ARP-таблицах и таблицах коммутации. Могут возникать ошибки при повышенном трафике и передаче по TCP. Если заменить и концентратор в сегменте 1 на коммутатор, то ошибки исчезнут.

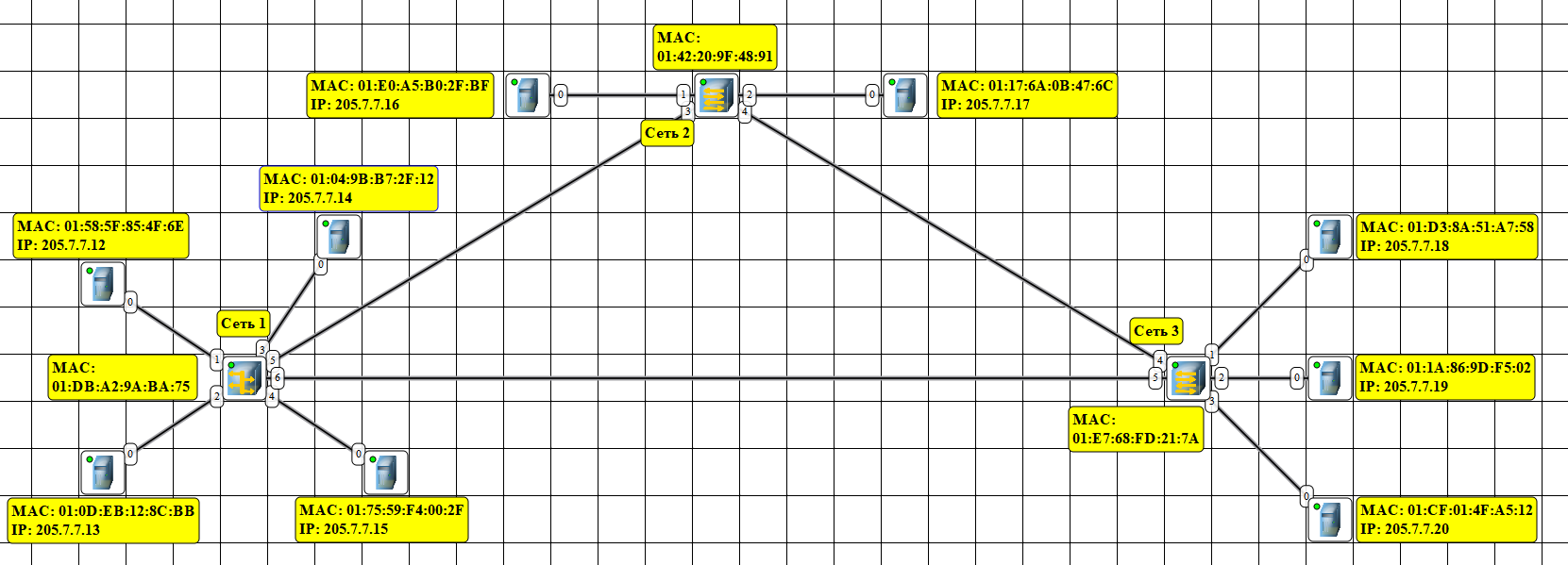
### Топология «Звезда» с концентратором



Работоспособна только если заполнены ARP-таблицы, иначе передача данных между сегментами заканчивается ошибкой TCP: transmission error, таблицы коммутации также должны быть полностью заполнены.

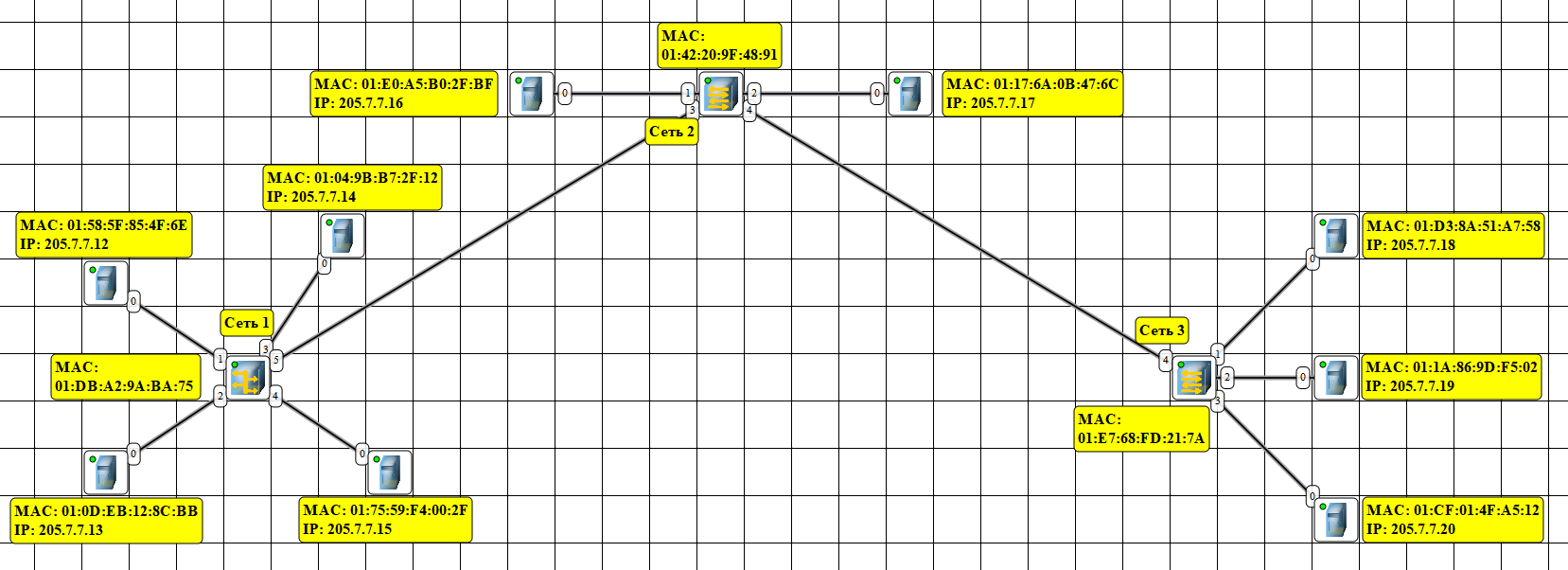
Таким образом звезду с концентратором можно использовать только внося статические записи в ARP-таблицы и таблицы коммутаций. Однако возникновение коллизий всё ещё делает этот вариант топологии неудачным.

### Топология «Кольцо»

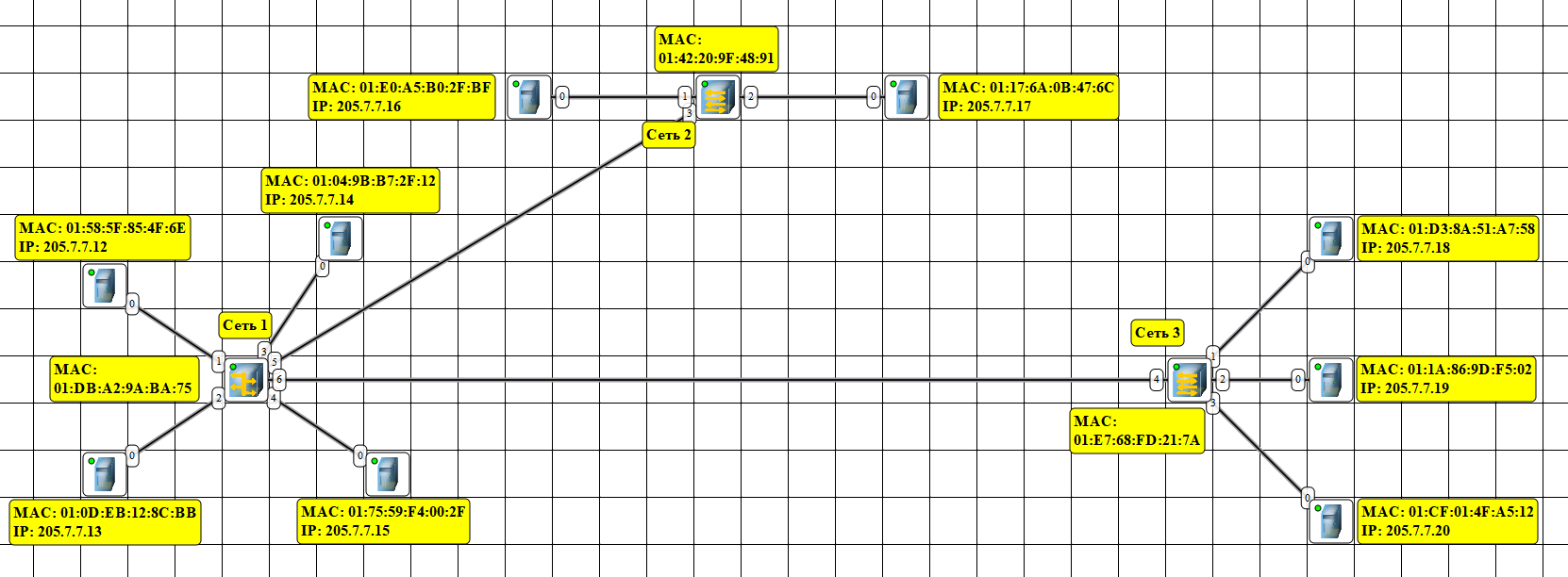


Неработоспособна из-за возникновения коллизий в концентраторе. Если заменить концентратор на коммутатор, то работать будет, но иногда возникают ошибки при передаче данных между сегментами.

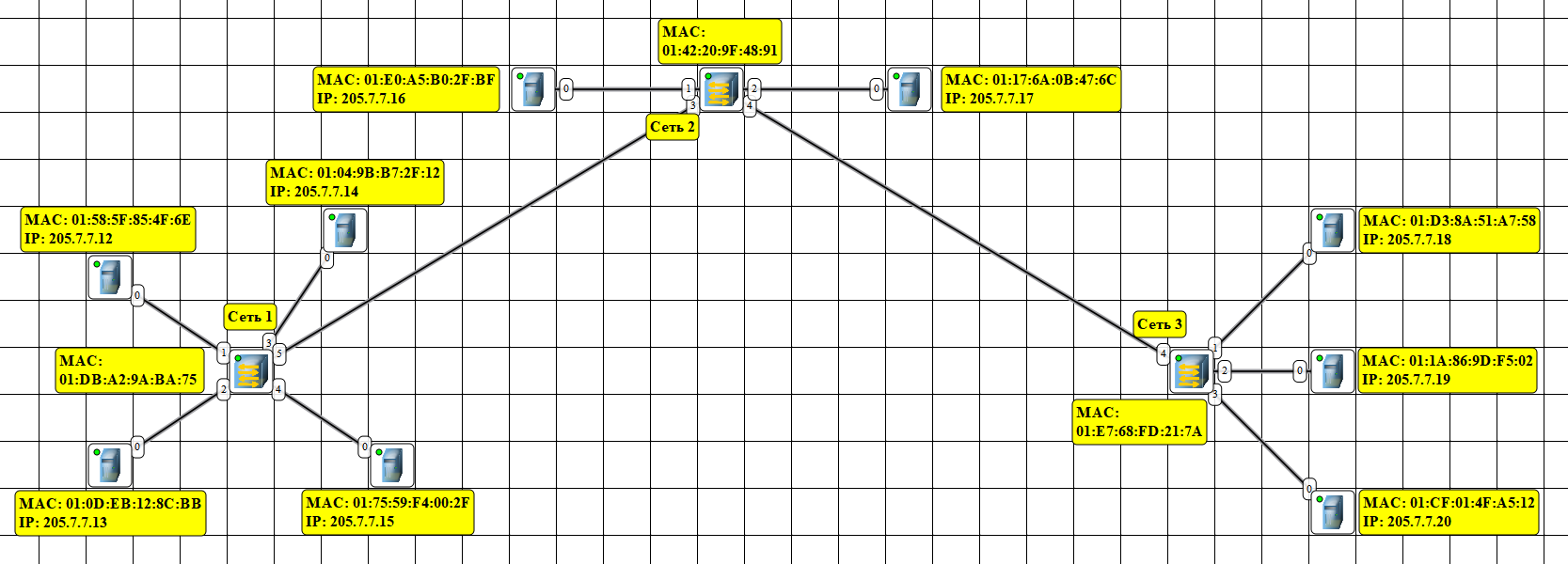
### Топология «Последовательное соединение»



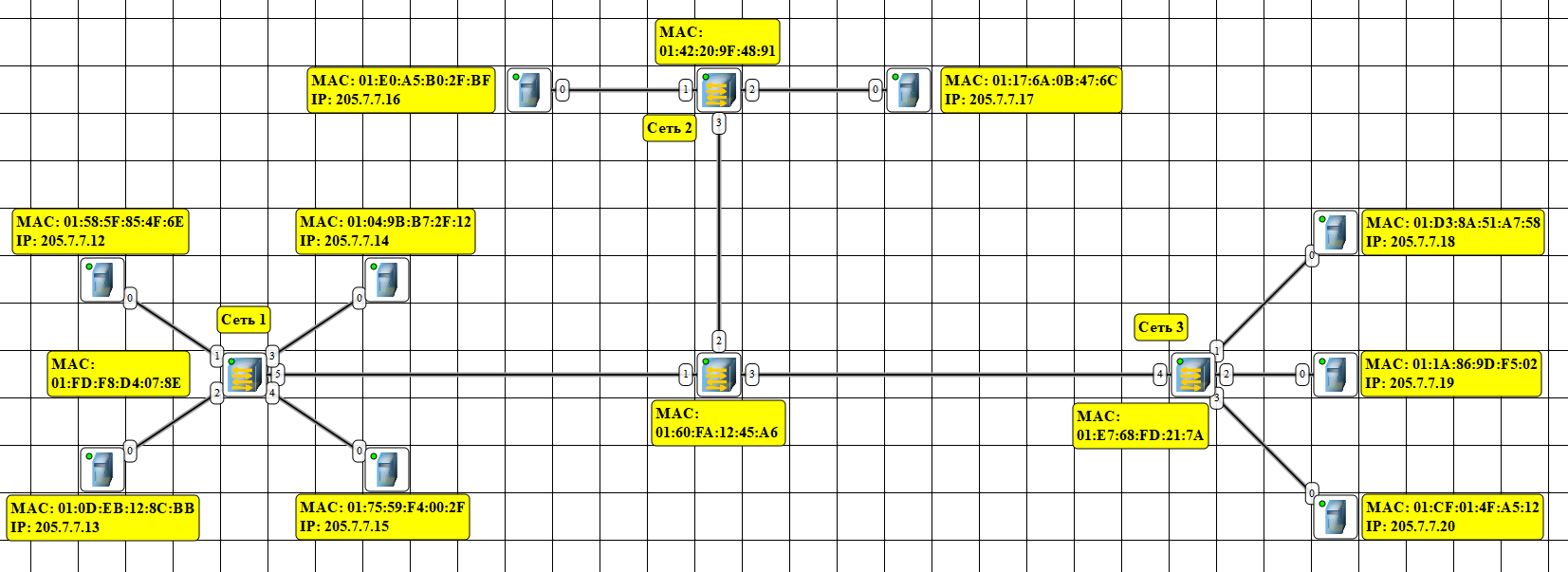
Реализуема, но могут появиться ошибки при передаче и UDP, и TCP пакетов. Если соединить немного иначе, то будут коллизии и возникнет ещё больше ошибок:



Если заменить концентратор на коммутатор и заполнить все ARP-таблицы, то проблем не будет возникать:



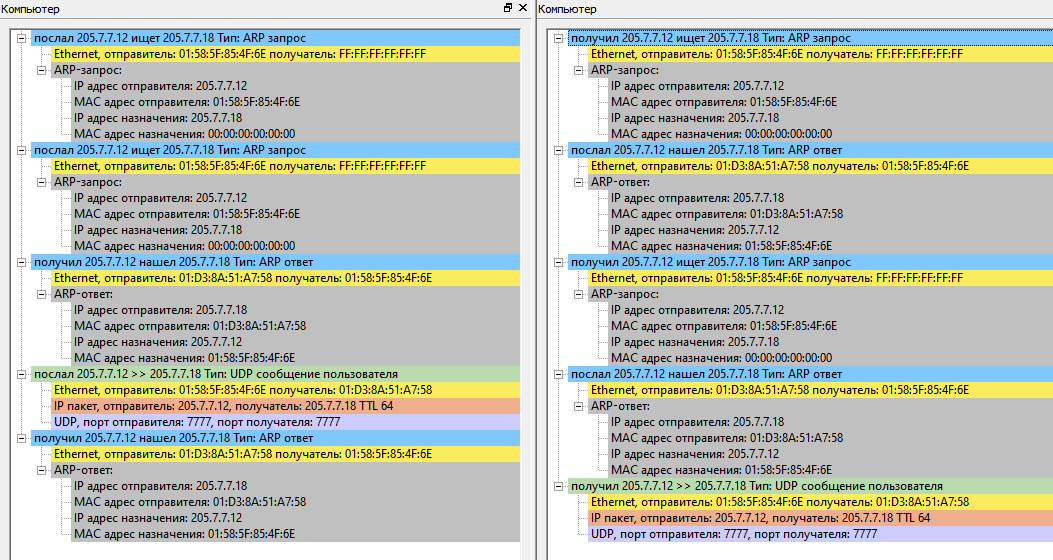
Таким образом делаем вывод, что лучше остальных себя проявила топология «Звезда» с коммутаторами на всех сегментах и в точке соединения. С ней и продолжим работать.



## Тестирование сети

### UDP

Отправим UDP-сообщение с данными объёмом 1 килобайт. Журнал компьютера-отправителя и компьютера-получателя представлен на скриншоте:



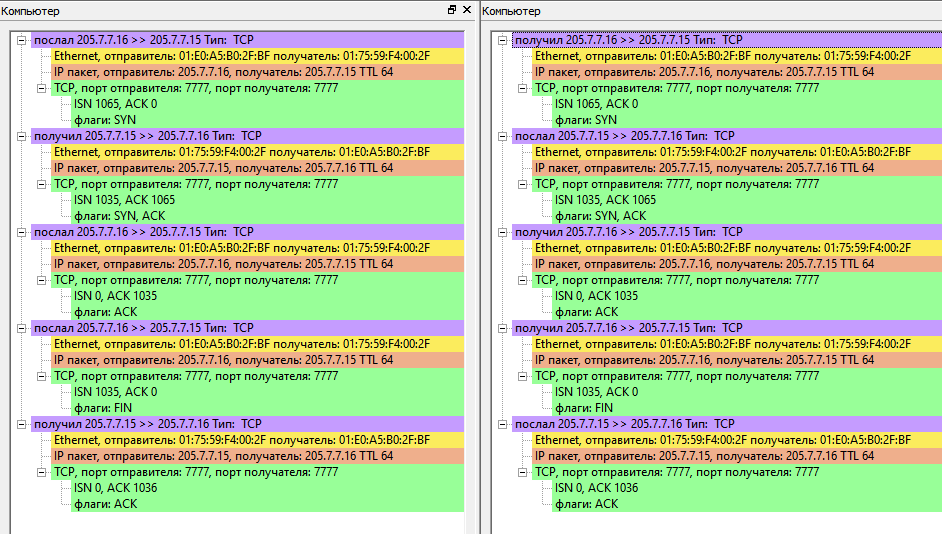
Передача осуществляется по следующему алгоритму:

1. Отправитель посылает ARP-запрос, который коммутаторами перенаправляется во все компьютеры и только итоговый получатель, который находится в сегменте 3, посылает ответ
2. Поскольку время пути до получателя и обратно вышло больше, чем тайм-аут ожидания, отправляющий компьютер посылает повторный ARP-запрос
3. Отправитель наконец получает положительный ARP-ответ о наличии нужного компьютера в сети и посылает UDP-сообщение с данными
4. Получателю доставлено UDP-сообщение
5. Отправителю приходит ARP-ответ на повторный запрос, но он на него никак не реагирует

Из-за того, что в сети у нас есть лишь коммутаторы, передача данных между любыми сегментами будут выглядеть идентично приведённому выше.

### TCP

Отправим TCP-сообщение объёмом 1 килобайт. Журналы отправителя и получателя представлены на скриншоте:



Передача осуществлялась по следующему алгоритму:

1. Отправка 2 ARP-сообщений и получение ответов на них так же как и в предыдущем пункте, здесь опущено, ARP-таблицы заполнены
2. Отправитель посылает TCP-запрос с флагом синхронизации и порядковым номером
3. Получатель в ответ отправляет служебный TCP-запрос с флагами синхронизации и номером подтверждения
4. Отправитель, удостоверившись в установке надёжного соединения, посылает сообщение с номером подтверждение и второе сообщение с непосредственно данными и флагом окончания сообщения, поскольку объём передаваемых данных равен 1 килобайту и для этого хватит одного сообщения
5. Получатель, получив пакет с данными и флагом окончания сообщения, отправляет ответное сообщение с флагом номера подтверждения
6. Отправитель получает сообщение о том, что данные успешно получены и значит передача данных может быть завершена

Из-за того, что в сети у нас есть лишь коммутаторы, передача данных между любыми сегментами будут выглядеть идентично приведённому выше.

# Вывод

Итак, в ходе данной лабораторной работы я построил в среде NetEmul несколько сетей, включая многосегментную, смоделировал передачи пакетов данных при помощи UDP и TCP в разных сетях. Также ознакомился с работой ARP-таблиц, таблиц коммутации и таблиц маршрутизации. Кроме того, я понял разницу между такими устройствами, как Концентратор (хаб) и Коммутатор (свитч).