

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

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

Выполнил:

Студент 3 курса группы Р3331  
Дворкин Борис Александрович

Преподаватель:

Алиев Тауфик Измайлович

г. Санкт-Петербург

2024 г.

# Содержание

<b>Содержание.....</b>	<b>2</b>
<b>1 Введение.....</b>	<b>3</b>
<b>2 Вариант лабораторной работы.....</b>	<b>3</b>
<b>3 Этап 1. Локальная сеть с концентратором.....</b>	<b>3</b>
3.1 Построение сети.....	3
3.2 Анализ таблиц.....	4
ARP-таблица.....	4
Таблица маршрутизации.....	5
3.3 Тестирование сети (отправка пакетов по UDP).....	6
3.3.1. Какие пакеты и кадры передаются в сети?.....	7
3.3.2. Как происходит передача, что содержится в пакетах?.....	8
3.3.3. Появились ли изменения в таблицах?.....	8
3.4 Тестирование сети (отправка пакетов по TCP).....	9
1. ARP-обмен (определение MAC-адресов).....	10
2. Установление TCP-соединения (трехэтапное рукопожатие).....	10
3. Передача данных.....	11
4. Завершение соединения (четырёхэтапное рукопожатие).....	12
<b>4 Этап 2. Локальная сеть с коммутатором.....</b>	<b>13</b>
4.1 Построение сети.....	13
4.2 Анализ таблиц.....	13
4.3 Тестирование сети (отправка пакетов).....	14
<b>5 Этап 3. Многосегментная локальная сеть.....</b>	<b>15</b>
5.1 Построение сети.....	15
5.2 Тестирование сети (отправка пакетов), анализ таблиц.....	15
<b>6 Выводы.....</b>	<b>16</b>

# 1 Введение

Целью работы является изучение принципов конфигурирования и процессов функционирования компьютерных сетей, представляющих собой несколько подсетей, связанных с помощью маршрутизаторов, процессов автоматического распределения сетевых адресов, принципов статической маршрутизации и динамической маршрутизации, а также передачи данных на основе стека протоколов TCP/IP (UDP и TCP), с использованием программы моделирования компьютерных сетей NetEmul.

## 2 Вариант лабораторной работы

Ф = 7 (Дворкин); И = 5 (Борис); О = 13 (Александрович); Н = 31 (Р3331)

=> N1 = 4; N2 = 6; N3 = 4; Класс IPv4-адресов: В, 164.44.38.12

## 3 Этап 1. Сеть с одним маршрутизатором

### 3.1 Построение сети

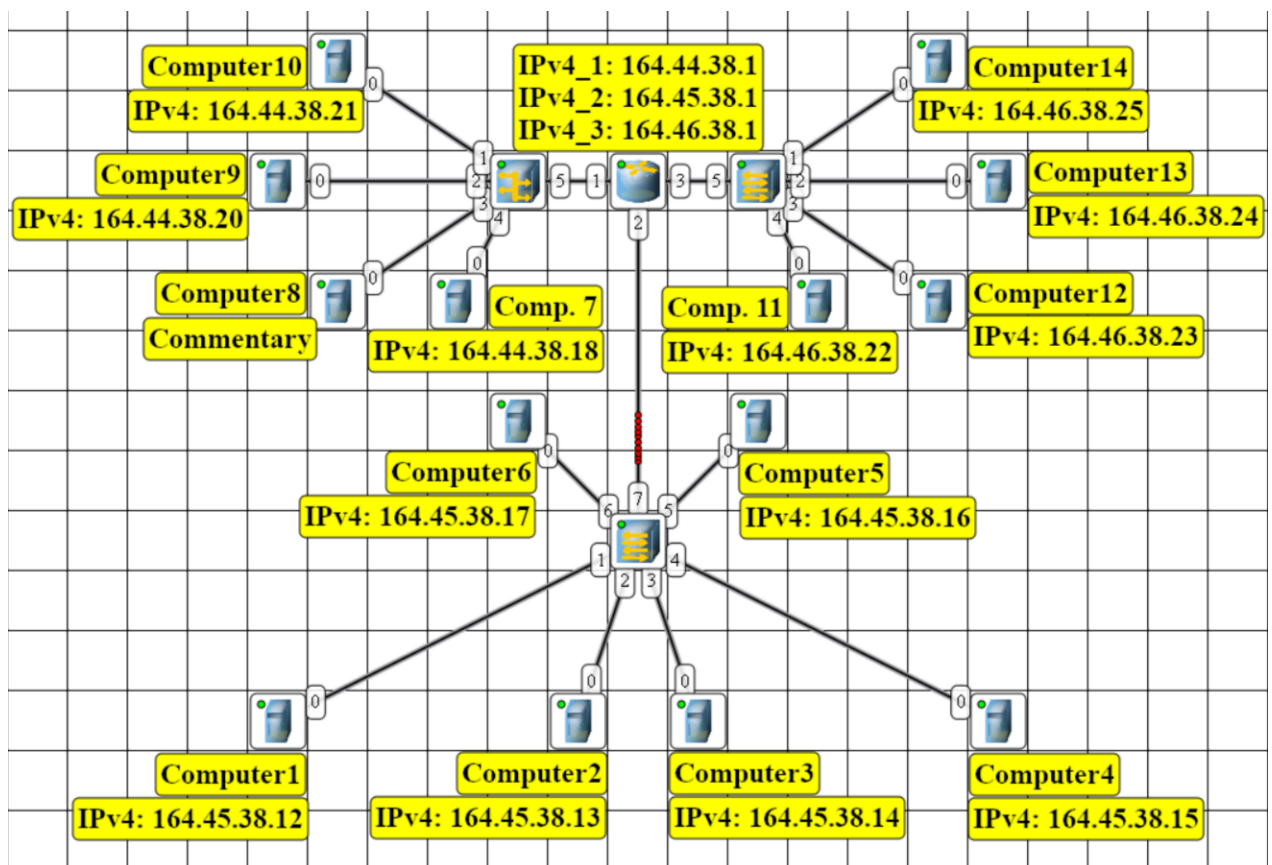


Рис.1: Модель сети с одним маршрутизатором

Построенная ранее сеть компьютеров 1-14 была разбита на подсети класса B: **164.44.0.0/16**, **164.45.0.0/16**, **164.46.0.0/16**

Подсети соединены между собой одним маршрутизатором, топология "звезда".

А для передачи пакетов между сетями, каждому компьютеру назначен "default gateway" в виде соответствующего интерфейса маршрутизатора.

То есть если мы хотим отправить пакет компьютеру с неизвестным нашей подсети IP-адресом, то мы отправляем пакет на определенный гейтвей маршрутизатора и он переправляет пакет в нужную подсеть.

## 3.2 Анализ таблиц маршрутизации

Таблица маршрутизации Computer 10:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.21	164.44.38.21	0	Connected
2	0.0.0.0	0.0.0.0	164.44.38.1	164.44.38.21	0	Static

Таблица маршрутизации Computer 2:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.45.0.0	255.255.0.0	164.45.38.13	164.45.38.13	0	Connected
2	0.0.0.0	0.0.0.0	164.45.38.1	164.45.38.13	0	Static

Таблица маршрутизации Computer 14:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.46.0.0	255.255.0.0	164.46.38.25	164.46.38.25	0	Connected
2	0.0.0.0	0.0.0.0	164.46.38.1	164.46.38.25	0	Static

Из данных таблиц можно сделать вывод, что помимо базовой записи с loopback адресом также появляется запись с адресом 0.0.0.0 - это тот самый "default gateway", который я описывал выше - это такой адрес, на который по умолчанию отправляется пакет данных, если адрес назначения (получателя) не принадлежит нашей подсети.

То есть, если например компьютер с IPv4 адресом 164.44.38.21 хочет отправить пакет на адрес 164.45.38.22 - это получается другая

подсеть 164.45.0.0/16, следовательно этот пакет отправляется на гейтвей маршрутизатора 164.46.38.1.

Таблица маршрутизатора:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.1	164.44.38.1	0	Connected
2	164.45.0.0	255.255.0.0	164.45.38.1	164.45.38.1	0	Connected
3	164.46.0.0	255.255.0.0	164.46.38.1	164.46.38.1	0	Connected

Вот тут как раз те самые записи о интерфейсах для взаимодействия с подсетями. Для сети 164.44.0.0/16 - свой интерфейс 164.44.38.1, для остальных - аналогично.

### 3.3 Тестирование сети (отправка пакетов по UDP)

Обычная отправка по UDP. Сначала ARP запросы, затем пакеты. Интересно тут то, что сначала мы определяем MAC адрес интерфейса LAN1 маршрутизатора:

Computer10 logs:

- sent 164.44.38.21 search 164.44.38.1 Type: ARP request
- received 164.44.38.21 found 164.44.38.1 Type: ARP response
- sent 164.44.38.21 >> 164.46.38.25 Type: UDP Message user

Router logs:

- received 164.44.38.21 search 164.44.38.1 Type: ARP request
- sent 164.44.38.21 found 164.44.38.1 Type: ARP response
- received 164.44.38.21 >> 164.46.38.25 Type: UDP Message user

Handwritten note: "Router LAN1" with an arrow pointing to the MAC address 01:28:AD:A2:2B:87 in the Router's ARP response log.

Затем отправляем туда данные и определяем MAC адрес итогового компьютера:

Switch46 logs:

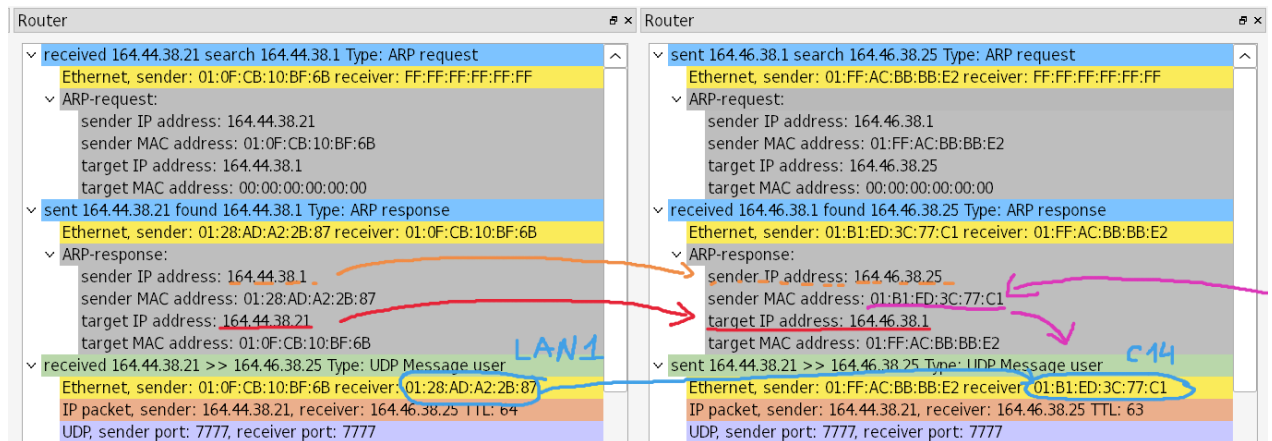
- received 164.46.38.1 search 164.46.38.25 Type: ARP request
- sent 164.46.38.1 found 164.46.38.25 Type: ARP response
- received 164.44.38.21 >> 164.46.38.25 Type: UDP Message user

Computer14 logs:

- received 164.46.38.1 search 164.46.38.25 Type: ARP request
- sent 164.46.38.1 found 164.46.38.25 Type: ARP response
- received 164.44.38.21 >> 164.46.38.25 Type: UDP Message user

Handwritten note: "Computer14" with an arrow pointing to the MAC address 01:B1:ED:3C:77:C1 in the Switch46's ARP response log.

И, наконец, с помощью таблицы маршрутизации и ARP-таблицы маршрутизатора пересылаем пакет между подсетями.



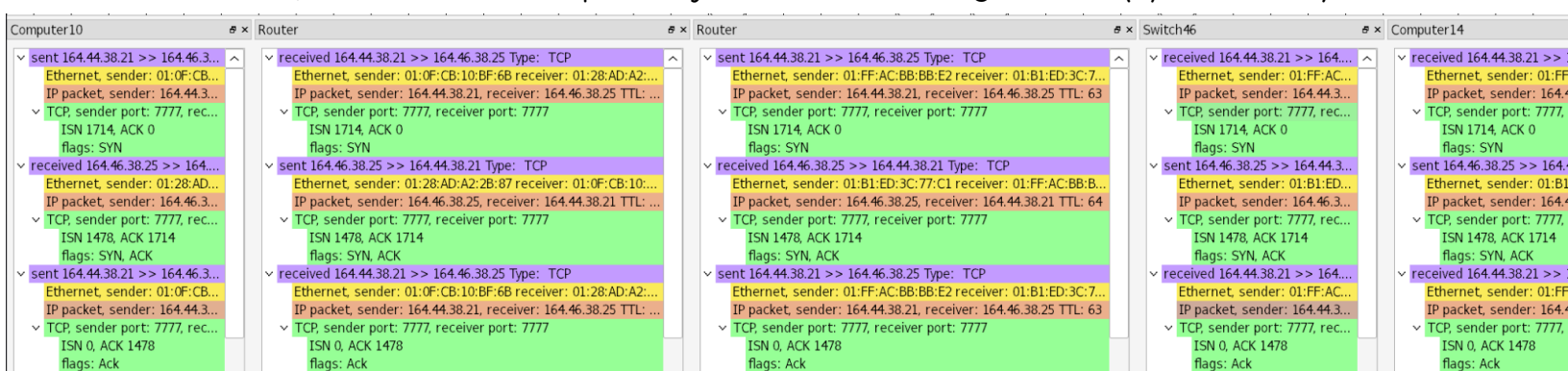
Композиция рис.2: Журналы сообщений сетевых устройств

Содержимое в пакетах при отправке по UDP обыденное - сначала Ethernet пакет с mac-адресами отправителя и получаемого, затем IP пакет с IP-адресами отправителя и получателя, и замыкает UDP-сегмент с информацией о портах отправителя и получателя.

### 3.4 Тестирование сети (отправка пакетов по TCP)

Механизм передачи по TCP был также крайне подробно описан мною в предыдущей лабораторной.

Первый этап - установление соединения. В отличие от UDP, где мы просто сразу же посылаем данные, в TCP мы посылаем пакет с флагом SYN, обозначая намерение установить соединение (synchronize).



При этом, в TCP пакет помещается ISN - Initial Sequence Number. Это произвольное 32-битное число, нужное чтобы различать пакеты между собой, нумеровать их. Если проще - мы передаём байты данных по протоколу и их нужно как-то нумеровать. Первый пакет посылается как ISN, а сами данные начинаются с ISN+1 (корректнее - с ISN+N, где N - количество байт), т.к. SYN пакеты "съели" один ISN.



Таким образом, Computer 14 получает SYN пакет от Computer 10, генерирует свой ISN, устанавливает дуплексную связь и отправляет в качестве ответа свой ISN, показывая что он увидел пришедший пакет (для этого добавляя флаг ACK - Acknowledge). А Computer 10 в свою очередь в ответ на пришедший пакет с флагами SYN, ACK от Computer 14, понимает что дуплексная связь установлена, видит ISN Computer 14, отправляет сообщение о том что он увидел (опять ACK) обратно и начинает передачу данных начиная с этого ISN+1. Аналогично если бы Computer 14 хотел передавать данные, он бы начинал их с ISN+1 компьютера 10, таким образом и получается дуплексная связь.

```

✓ sent 164.44.38.21 >> 164.46.38.25 Type: TCP
  Ethernet, sender: 01:0F:CB:10:BF:6B receiver: 01:28:AD:A2:2B:87
  IP packet, sender: 164.44.38.21, receiver: 164.46.38.25 TTL: 64
✓ TCP, sender port: 7777, receiver port: 7777
  ISN 1478, ACK 0
  flags: No flags
✓ sent 164.44.38.21 >> 164.46.38.25 Type: TCP
  Ethernet, sender: 01:0F:CB:10:BF:6B receiver: 01:28:AD:A2:2B:87
  IP packet, sender: 164.44.38.21, receiver: 164.46.38.25 TTL: 64
✓ TCP, sender port: 7777, receiver port: 7777
  ISN 1479, ACK 0
  flags: No flags
✓ sent 164.44.38.21 >> 164.46.38.25 Type: TCP
  Ethernet, sender: 01:0F:CB:10:BF:6B receiver: 01:28:AD:A2:2B:87
  IP packet, sender: 164.44.38.21, receiver: 164.46.38.25 TTL: 64
✓ TCP, sender port: 7777, receiver port: 7777
  ISN 1480, ACK 0
  flags: No flags

```

В самих пакетах данных ничего особенного нету - никаких флагов, просто - TCP-сегменты с указанием портов отправителя и получателя, увеличивающийся ISN на 1 (а по-хорошему, на количество байтов) и всё:

Как мы отправили все пакеты данных надо инициализировать закрытие соединения. Для этого служит флаг Fin - Finish. И аналогично Computer 14 отвечает Computer 10 флагом ACK, чтобы показать что он увидел закрытие им соединения со своей стороны. Таким образом совершается определенное "рукопожатие". В реальном мире должно быть 4х этапное рукопожатие, но в нашем случае, это NetEmul, поэтому имеем что имеем:

<pre> ✓ sent 164.44.38.21 &gt;&gt; 164.46.38.25 Type: TCP   Ethernet, sender: 01:0F:CB:10:BF:6B receiver: 01:28:AD:A2:2B:87   IP packet, sender: 164.44.38.21, receiver: 164.46.38.25 TTL: 64 ✓ TCP, sender port: 7777, receiver port: 7777   ISN 1482, ACK 0   flags: Fin ✓ received 164.46.38.25 &gt;&gt; 164.44.38.21 Type: TCP   Ethernet, sender: 01:28:AD:A2:2B:87 receiver: 01:0F:CB:10:BF:6B   IP packet, sender: 164.46.38.25, receiver: 164.44.38.21 TTL: 63 ✓ TCP, sender port: 7777, receiver port: 7777   ISN 0, ACK 1483   flags: Ack </pre>	<pre> ✓ received 164.44.38.21 &gt;&gt; 164.46.38.25 Type: TCP   Ethernet, sender: 01:FF:AC:BB:BB:E2 receiver: 01:B1:ED:3C:77:C1   IP packet, sender: 164.44.38.21, receiver: 164.46.38.25 TTL: 63 ✓ TCP, sender port: 7777, receiver port: 7777   ISN 1482, ACK 0   flags: Fin ✓ sent 164.46.38.25 &gt;&gt; 164.44.38.21 Type: TCP   Ethernet, sender: 01:B1:ED:3C:77:C1 receiver: 01:FF:AC:BB:BB:E2   IP packet, sender: 164.46.38.25, receiver: 164.44.38.21 TTL: 64 ✓ TCP, sender port: 7777, receiver port: 7777   ISN 0, ACK 1483   flags: Ack </pre>
--	--

## 4 Этап 2. Сеть с двумя маршрутизаторами

### 4.1 Построение сети

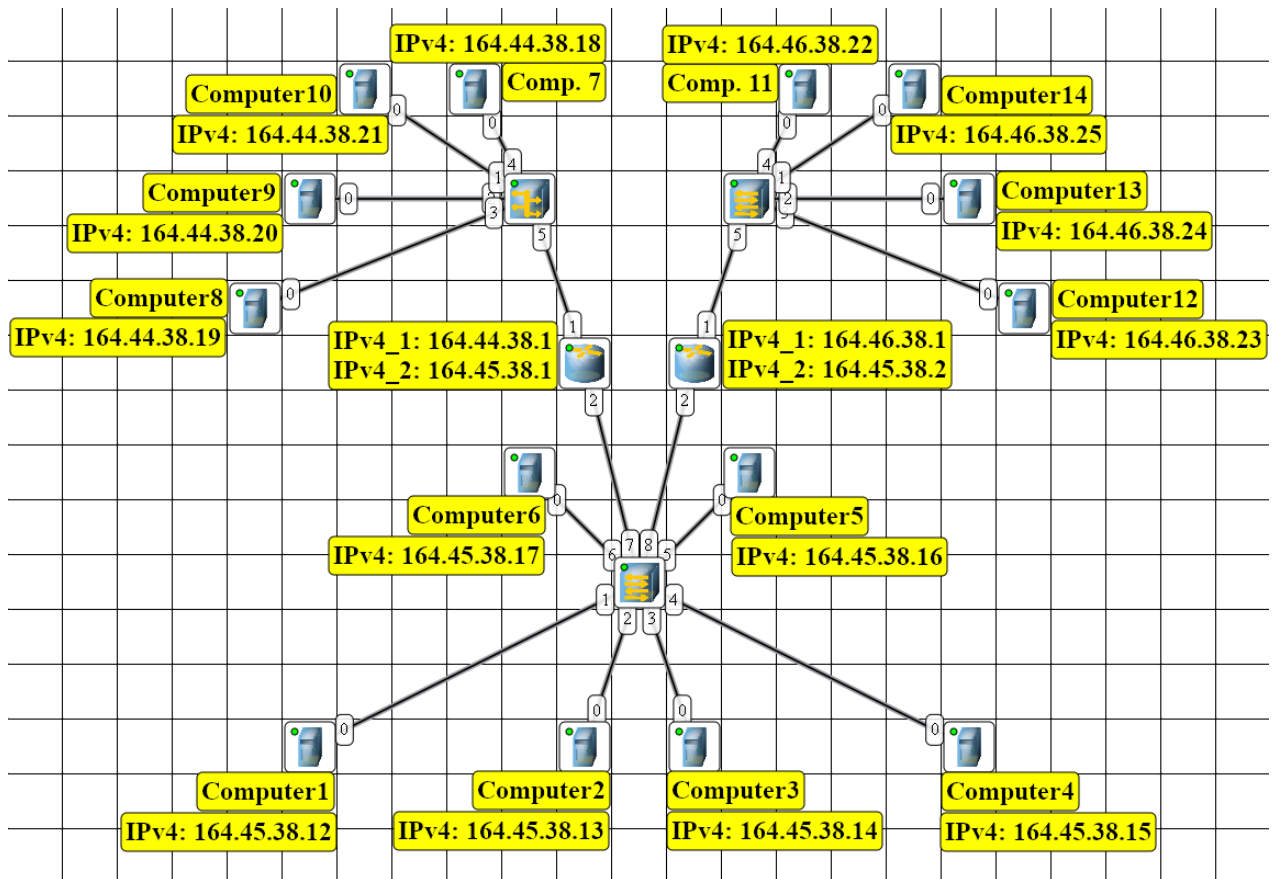


Рис.4: Модель сети с двумя маршрутизаторами

Крайне интересная оказалась ситуация с маршрутизаторами. А долго не мог понять что не так и как разрешить проблему - ведь для подсети 164.45.0.0/16 - получается два дефолтных гейтвея, оба вот эти маршрутизаторы, как понять какой из них выбрать? Да, я могу поменять на одном из них IPv4 адрес, чтобы они не были одинаковыми, но что дальше? Решение пришло не сразу, но довольно очевидное - надо просто добавить loopback каждому из маршрутизаторов, чтоб их гейтвеи перенаправляли друг на друга. В остальном - такая топология выглядит куда более надежной чем первичная, хоть и вышла не такой лаконичной и звездообразной)

### 4.2 Анализ таблиц

Тут будет целесообразно проанализировать лишь таблицы маршрутизаторов, т.к. таблицы компьютеров не изменились.



Таблица маршрутизатора 1:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.45.0.0	255.255.0.0	164.45.38.2	164.45.38.2	0	Connected
2	164.46.0.0	255.255.0.0	164.46.38.1	164.46.38.1	0	Connected
3	0.0.0.0	0.0.0.0	164.45.38.1	164.45.38.2	0	Static

Таблица маршрутизатора 2:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.1	164.44.38.1	0	Connected
2	164.45.0.0	255.255.0.0	164.45.38.1	164.45.38.1	0	Connected
3	0.0.0.0	0.0.0.0	164.45.38.2	164.45.38.1	0	Static

Исходя из таблиц наглядно видны новые записи с loopback адресом, ссылающиеся друг на друга, которые я добавил для обеспечения работоспособности сети.

### 4.3 Тестирование сети (отправка пакетов)

Всё аналогично предыдущим пунктам. Добраться можно из любой подсети в любую, все сетевые устройства функционируют как обычно, а повторяться по 10 раз я не буду.

## 5 Этап 3. Сеть с тремя маршрутизаторами

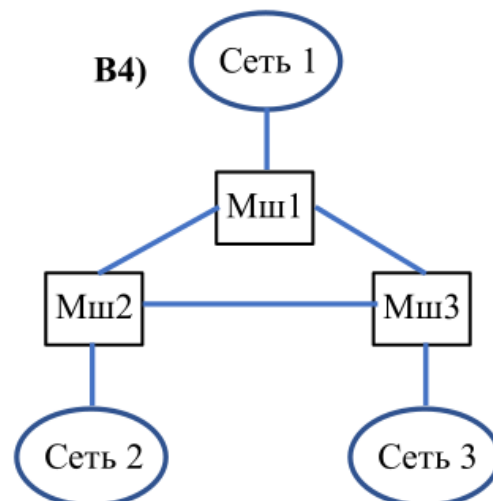
### 5.1 Построение сети

На этом этапе мне предстоит выбрать один из возможных вариантов, а выбрать я хочу, очевидно, наилучший по меркам достоинств и недостатков, поэтому объясню свой выбор по порядку.

Перед тем как выбирать топологию из предложенных, надо определить критерии выбора - пусть это будет отказоустойчивость - я считаю, что в сетевой инженерии это одна из самых важных характеристик. Например, если один из каналов связи внезапно выйдет из строя - всегда должен быть какой-то другой канал, который его заменит в случае чего, и пусть пакет пройдёт чуть большее расстояние. А также, не должно совершаться лишних roundtrip'ов из одной страны в другую просто потому что у нас нету посередине маршрутизатора)

Исходя из этого - я рассматриваю топологию В4. Несмотря на то, что при выходе Мш1-3, сеть становится полностью изолированной от других, в данной топологии нет никаких проблем с настройкой, как с топологией В3, (там просто невозможно корректно настроить без RIP'a), а также стоит учитывать что обычно из строя выходят каналы связи, а не само оборудование, поэтому надёжности данной топологии всё равно нам хватит с лихвой. Из плюсов - она также предоставляет более равномерное распределение трафика и не имеет избыточных соединений, что упрощает и конфигурацию, и администрирование.

Исходя из предоставленных аргументов, я считаю выбор данной топологии разумным.



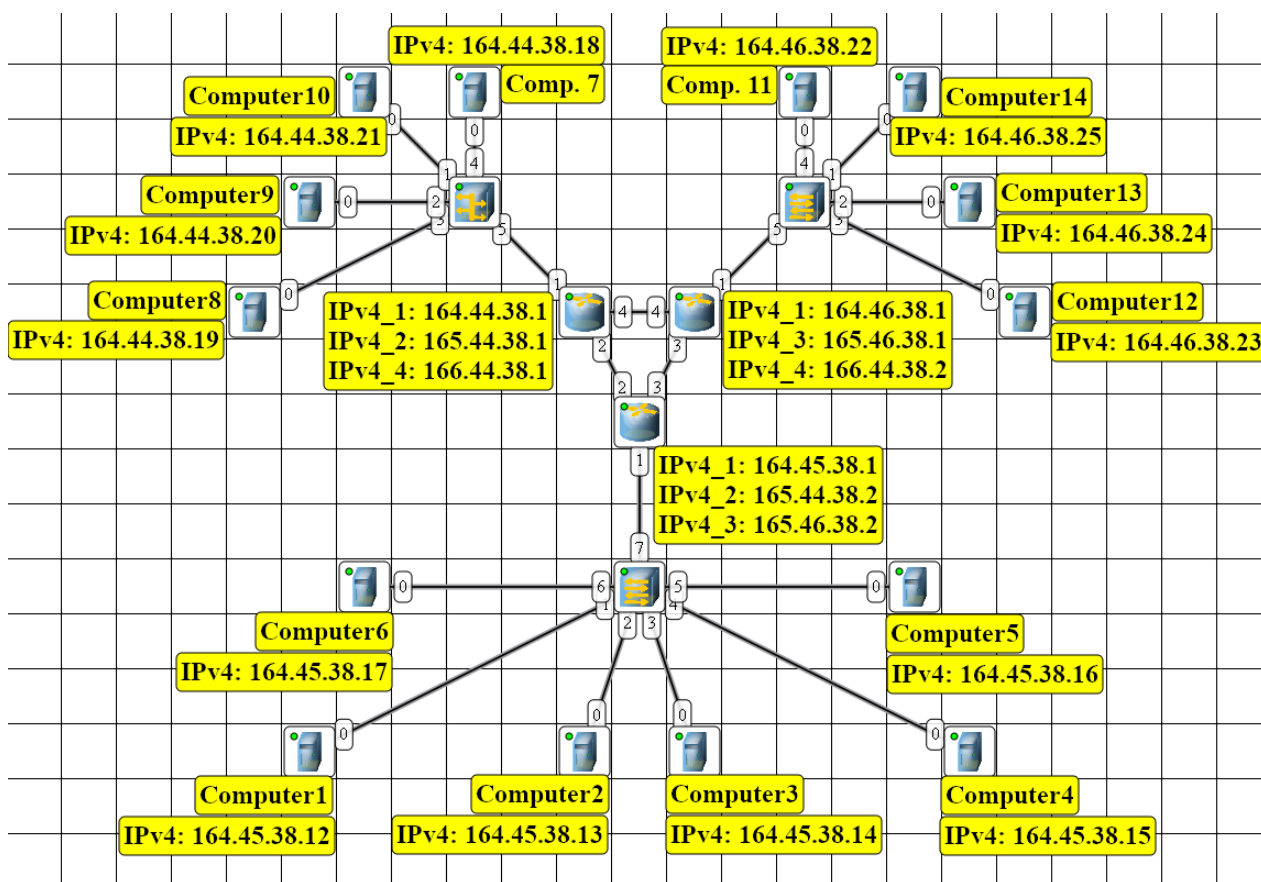


Рис.4: Модель сети с тремя маршрутизаторами

## 5.2 Тестирование сети (отправка пакетов), анализ таблиц

Никаких глобальных изменений не произошло - я лишь добавил три произвольные подсети между маршрутизаторами, и для этого назначил статические маршруты чтоб сообщить как из одной подсети попасть в другую.

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	165.44.38.1	165.44.38.2	0	Static
2	164.45.0.0	255.255.0.0	164.45.38.1	164.45.38.1	0	Connected
3	164.46.0.0	255.255.0.0	165.46.38.1	165.46.38.2	0	Static
4	165.44.0.0	255.255.0.0	165.44.38.2	165.44.38.2	0	Connected
5	165.46.0.0	255.255.0.0	165.46.38.2	165.46.38.2	0	Connected

Передача тоже никак не поменялась, просто теперь у нас добавился еще один узел при передаче в сеть 164.45.0.0/16, при этом передача из сети 164.44.0.0/16 в сеть 164.46.0.0/16 стала более логичной - так и должно происходить взаимодействие в сетях, без перегрузок узлов.

## 5.3 Настройка динамической маршрутизации по RIP

После настройки протокола RIP, маршрутизаторы начали слать RIP-запросы примерно каждые 13.5 секунд, для динамической маршрутизации, т.к. в таблицах маршрутизации появились новые записи для тех самых сетей между маршрутизаторами, а также подкорректировались метрики, если я всё правильно заметил:

	Destination	Mask	Gateway	Interface	Metric	Source
1	164.44.0.0	255.255.0.0	164.44.38.1	164.44.38.1	0	Connected
2	164.45.0.0	255.255.0.0	165.44.38.2	165.44.38.1	1	Static
3	164.46.0.0	255.255.0.0	166.44.38.2	166.44.38.1	1	Static
4	165.44.0.0	255.255.0.0	165.44.38.1	165.44.38.1	0	Connected
5	165.46.0.0	255.255.0.0	165.44.38.2	165.44.38.1	1	RIP
6	166.44.0.0	255.255.0.0	166.44.38.1	166.44.38.1	0	Connected

Router	# x	Router	# x	Router
> 09:47:55.739 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:48:09.240 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:48:22.739 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:48:36.238 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:48:49.788 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:49:03.338 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:49:16.889 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:49:30.387 sent 164.44.38.1 >> 164.44.255.255 Type: RIP > 09:49:43.888 sent 164.44.38.1 >> 164.44.255.255 Type: RIP ✓ 09:49:57.387 sent 164.44.38.1 >> 164.44.255.255 Type: RIP Ethernet, sender: 01:28:AD:A2:2B:87 receiver: FF:FF:FF:FF:FF:FF IP packet, sender: 164.44.38.1, receiver: 164.44.255.255 TTL: 64 UDP, sender port: 520, receiver port: 520		> 09:47:56.189 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:48:09.697 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:48:23.189 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:48:36.689 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:48:50.239 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:49:03.839 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:49:17.339 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:49:30.837 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:49:44.337 sent 164.46.38.1 >> 164.46.255.255 Type: RIP > 09:49:57.837 sent 164.46.38.1 >> 164.46.255.255 Type: RIP		> 09:48:00.699 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:48:14.189 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:48:27.688 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:48:41.240 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:48:54.788 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:49:08.338 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:49:21.838 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:49:35.338 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:49:48.839 sent 164.45.38.1 >> 164.45.255.255 Type: RIP > 09:50:02.336 sent 164.45.38.1 >> 164.45.255.255 Type: RIP

## 5.4 Настройка DHCP

Computer1
> sent 0.0.0.0 >> 255.255.255.255 Type: DHCP message Ethernet, sender: 01:68:AE:60:AD:C8 receiver: FF:FF:FF:FF:FF:FF IP packet, sender: 0.0.0.0, receiver: 255.255.255.255 TTL: 64 UDP, sender port: 67, receiver port: 68 DHCP Message, Type: DHCPDISCOVER > received 0.0.0.0 >> 255.255.255.255 Type: DHCP message > received 0.0.0.0 >> 255.255.255.255 Type: DHCP message > received 0.0.0.0 >> 255.255.255.255 Type: DHCP message > received 164.45.38.1 >> 255.255.255.255 Type: DHCP message > received 164.45.38.1 >> 255.255.255.255 Type: DHCP message > received 164.45.38.1 >> 255.255.255.255 Type: DHCP message > received 164.45.38.1 >> 255.255.255.255 Type: DHCP message ✓ sent 0.0.0.0 >> 255.255.255.255 Type: DHCP message Ethernet, sender: 01:68:AE:60:AD:C8 receiver: FF:FF:FF:FF:FF:FF IP packet, sender: 0.0.0.0, receiver: 255.255.255.255 TTL: 64 UDP, sender port: 67, receiver port: 68 DHCP Message, Type: DHCPREQUEST Xid: 2382, Yiaddr: 0.0.0.0 Siaddr: 164.45.38.1, Chaddr: 01:68:AE:60:AD:C8 > received 0.0.0.0 >> 255.255.255.255 Type: DHCP message > received 0.0.0.0 >> 255.255.255.255 Type: DHCP message > received 0.0.0.0 >> 255.255.255.255 Type: DHCP message ✓ received 164.45.38.1 >> 255.255.255.255 Type: DHCP message Ethernet, sender: 01:62:F9:51:E2:8D receiver: FF:FF:FF:FF:FF:FF IP packet, sender: 164.45.38.1, receiver: 255.255.255.255 TTL: 64 UDP, sender port: 68, receiver port: 67 DHCP Message, Type: DHCPACK Xid: 4755, Yiaddr: 164.45.38.12 Siaddr: 164.45.38.1, Chaddr: 01:4D:C9:0A:97:B1

Что сказать, работаем) Discover -> offer -> request -> ACK

## 6 Выводы

Выбор топологии в этой лабораторной работе вызвал неожиданные сложности, которые на первый взгляд не казались такими уж критичными. Пытаясь построить кольцевую топологию с тремя маршрутизаторами, я столкнулся с фундаментальной проблемой: концентратор и кольцо — вещи практически несовместимые. Концентратор, работая на физическом уровне, тупо дублирует сигналы на все порты, что приводит к бесконечным петлям и "размножению" пакетов в кольцевой схеме.

Статическая маршрутизация в такой топологии без RIP превращается в настоящий квест. Интересно было разобраться, почему при добавлении третьего маршрутизатора нельзя просто расширить схему, которая работала для двух. Оказалось, что loopback адреса и default gateway, спасавшие ситуацию в простой схеме, в кольце приводят к катастрофическим последствиям — пакеты начинают путешествовать вечно, а таблицы маршрутизации превращаются в свалку противоречий.

Протокол RIP в этом контексте выглядит как снизошедший до нас архангел — он автоматически строит оптимальные маршруты и справляется с проблемой петель. Но насколько ценно понимание, как настроить сеть "руками", без автоматики? Когда я разобрался в деталях процесса DHCP с его DISCOVER-OFFER-REQUEST-ACK, то понял, что настройка сетевых параметров вручную — это, конечно, полезно для понимания, но на этом мы закончим это упражнение)

Особенно мне понравилось чувствовать разницу между коммутаторами и концентраторами. Не так давно я менял у себя в комнате концентратор на коммутатор, и с пользовательской точки зрения буквально ничего не поменялось, кроме скорости соединения, но теперь я понимаю, что хоть коммутатор у меня и неуправляемый и я не настраиваю пути как сетевой инженер, под капотом оно работает по-другому и более эффективно 😊. А для более сложных топологий разница между физическим и канальным уровнем вообще критична для стабильной работы.

Забавно, что в реальности никто не будет строить кольцевую топологию на концентраторах, и знания о подобных ограничениях выглядят как чисто академические. Но благодаря этим "академическим" сложностям я глубже понял фундаментальные принципы маршрутизации, важность правильного выбора сетевых устройств и логику работы протоколов вроде DHCP и RIP.

В реальных проектах ценятся не заученные цифры, а понимание принципов и способность решать нестандартные проблемы, с

которыми не справляются автоматические системы. И на этом фронте я, кажется, неплохо продвинулся. Полезный курс, я считаю.