

Санкт-Петербургский Национальный
Исследовательский Университет Информационных
технологий, механики и оптики

Лабораторная работа 2
Моделирование сети с топологией звезда

Выполнил: Фисенко
Максим Вячеславович
Шкода Глеб Ярославович
Группа № К34211
Проверила: Казанова
Полина Петровна

Санкт-Петербург
2024

Цель работы:

Смоделировать сеть с топологией «Звезда».

Задачи:

1. Создать сеть.
2. Сформировать трафик.
3. Заменить в сети концентраторы на коммутаторы.
4. Создать статическую маршрутизацию.
5. Настроить динамическую маршрутизацию.

Ход работы:

1. Создание сети

В ходе выполнения данной лабораторной работы первым делом было необходимо создать сеть. В программе **Cisco Packet Tracer** были добавлены 4 концентратора и 10 клиентских компьютеров таким образом, чтобы получилась сеть топологии звезда. При этом за концентраторами Hub3 и Hub4 было установлено по 1 клиентскому компьютеру, что соответствует последней цифре в табельном номере ИСУ (336751). Устройства в сети также были соединены (рисунок 1).

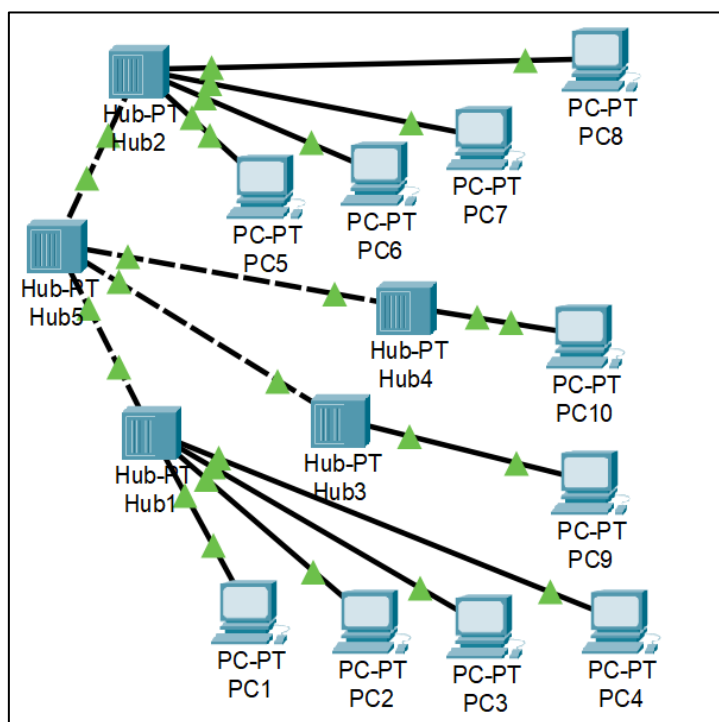


Рисунок 1 - Получившаяся сеть топологии звезда

Далее с помощью опции **IP Configuration** всем рабочим станциям (клиентским компьютерам) были даны ip-адреса из сети 192.168.24.0/24. Устройства получили статические ip-адреса 192.168.24.1, 192.168.24.2 и так далее (рисунок 2).

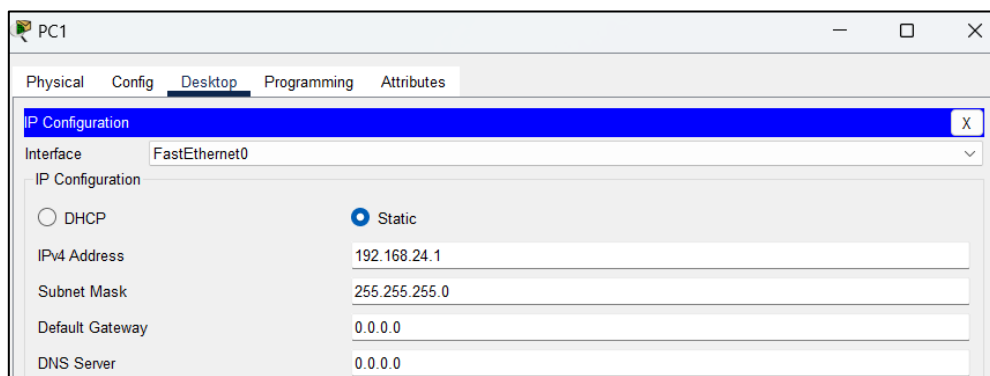


Рисунок 2 - Настроенный вручную ip-адрес устройства PC1

Затем на каждом из конечных устройств была запущена команда **ipconfig /all**, отображающая информацию о конфигурации IP на данном устройстве. С помощью этой команды было подтверждено, что IP-адреса на хостах были установлены успешно (рисунок 3).

```
C:\>ipconfig /all

FastEthernet0 Connection: (default port)

    Connection-specific DNS Suffix...:
    Physical Address.....: 0003.E44E.9137
    Link-local IPv6 Address.....: FE80::203:E4FF:FE4E:9137
    IPv6 Address.....: ::
    IPv4 Address.....: 192.168.24.1
    Subnet Mask.....: 255.255.255.0
    Default Gateway.....: ::
```

Рисунок 3 - Информация о конфигурации сети устройства PC1

После этого была осуществлена проверка доступности узлов в построенной сети. Между некоторыми узлами была запущена утилита **ping**, позволяющая проверить качество соединения в сети. Утилита была запущена между следующими парами устройств: PC1 и PC2, PC1 и PC8, PC1 и PC9 (рисунок 4), PC5 и PC10 (рисунок 5). Во всех четырех случаях все пакеты между узлами были успешно доставлены, что говорит о правильной настройке сети.

```

C:\>ping 192.168.24.8

Pinging 192.168.24.8 with 32 bytes of data:

Reply from 192.168.24.8: bytes=32 time=11ms TTL=128
Reply from 192.168.24.8: bytes=32 time=9ms TTL=128
Reply from 192.168.24.8: bytes=32 time<1ms TTL=128
Reply from 192.168.24.8: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.24.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 11ms, Average = 5ms

C:\>ping 192.168.24.2

Pinging 192.168.24.2 with 32 bytes of data:

Reply from 192.168.24.2: bytes=32 time<1ms TTL=128
Reply from 192.168.24.2: bytes=32 time<1ms TTL=128
Reply from 192.168.24.2: bytes=32 time<1ms TTL=128
Reply from 192.168.24.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.24.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.24.9

Pinging 192.168.24.9 with 32 bytes of data:

Reply from 192.168.24.9: bytes=32 time<1ms TTL=128
Reply from 192.168.24.9: bytes=32 time<1ms TTL=128
Reply from 192.168.24.9: bytes=32 time<1ms TTL=128
Reply from 192.168.24.9: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.24.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Рисунок 4 - Успешный запуск ping между PC1 и PC2, PC8, PC9

```

C:\>ping 192.168.24.10

Pinging 192.168.24.10 with 32 bytes of data:

Reply from 192.168.24.10: bytes=32 time<1ms TTL=128
Reply from 192.168.24.10: bytes=32 time<1ms TTL=128
Reply from 192.168.24.10: bytes=32 time<1ms TTL=128
Reply from 192.168.24.10: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.24.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Рисунок 5 - Успешный запуск ping между PC5 и PC10

2. Формирование трафика

Первым делом на устройстве PC2 была открыта утилита **Traffic Generator**, позволяющая формировать трафик между двумя узлами. В поля утилиты были вбиты необходимые данные в соответствии с вариантом 4 таблицы 2 в инструкции к лабораторной работе (рисунок 6). Аналогичные действия, но с другими данными, были выполнены для второй пары устройств.

Source Settings

Source Device: PC2

Outgoing Port: FastEthernet0 ☐ Auto Select Port

PDU Settings

Select Application: PING

Destination IP Address: 192.168.24.7

Source IP Address: 192.168.24.2

TTL: 32

TOS: 0

Sequence Number:

Size: 320

Simulation Settings

☐ Single Shot

☒ Periodic

Interval: 0.035 Seconds

Рисунок 6 - заполненные данные утилиты Traffic Generator

Затем была нажата кнопка «**Send**», вследствие чего появился следующий сетевой трафик: PC2 – PC7, Traffic Generator, $N = 320$, $T = 0.035$. Во время отправки пакетов по этому трафику, на устройстве PC1 была запущена утилита **ping**, которая пыталась отправить 100 пакетов на устройство PC8. В результате из 100 пакетов было потеряно 2, то есть коэффициент потери пакетов составил 2%.

После этого период повторения пакетов был уменьшен до минимально возможного ($T = 0.55$). После этого была также запущена утилита **ping**: на этот раз коэффициент потери пакетов равнялся 3%.

После этого были также проведены испытания со следующим сетевым трафиком: PC3 – PC7, Traffic Generator, $N = 870$, $T = 0.55$, был потерян только 1 пакет. После этого период повторения был уменьшен до $T = 0.1$, а коэффициент потери пакетов увеличился до 6% (рисунок 7).

```
Ping statistics for 192.168.24.8:
  Packets: Sent = 100, Received = 94, Lost = 6 (6% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 9ms, Average = 0ms
```

Рисунок 7 - Результат работы утилиты ping после уменьшения периода отправки пакетов

Наконец, в последнем, пятом, испытании были включены оба сетевых трафика. Коэффициент потери пакетов в таком случае составил уже 7%. Результаты всех испытаний представлены ниже в таблице 1.

Таблица 1 - Результаты формирования трафика сети

№ испытания	Сетевой трафик	Состоян ие потока	Кол-во потерянн ых пакетов	Коэффициент потери пакетов, %
1	PC1 – PC8, ping, n = 100	+	2	2
	PC2 – PC7, Traffic Generator, N = 320, T = 0.035	+		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.55	-		
2	PC1 – PC8, ping, n = 100	+	3	3
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	+		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.55	-		
3	PC1 – PC8, ping, n = 100	+	1	1
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	-		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.55	+		
4	PC1 – PC8, ping, n = 100	+	6	6
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	-		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.1	+		
5	PC1 – PC8, ping, n = 100	+	7	7
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	+		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.1	+		

3. Замена концентраторов коммутаторами

На данном этапе концентраторы Hub1, Hub2 и Hub5 были заменены на коммутаторы 2960, сеть же осталась в рабочем состоянии (рисунок 8).

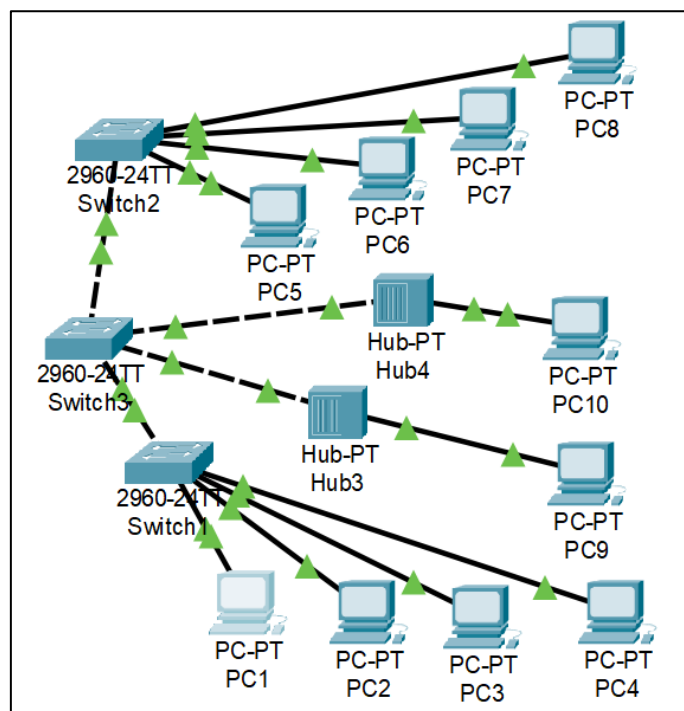


Рисунок 8 - Сеть после замены концентраторов коммутаторами

После этого был сформирован трафик и были проведены такие же испытания, как и в сети с концентраторами до этого. Результаты испытаний представлены ниже в таблице 2.

Таблица 2 - результаты формирования трафика сети с коммутаторами

№ испытания	Сетевой трафик	Состояние потока	Кол-во потерянных пакетов	Коэффициент потери пакетов, %
1	PC1 – PC8, ping, n = 100	+	0	0
	PC2 – PC7, Traffic Generator, N = 320, T = 0.035	+		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.55	-		
2	PC1 – PC8, ping, n = 100	+	0	0
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	+		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.55	-		
3	PC1 – PC8, ping, n = 100	+	0	0
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	-		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.55	+		
4	PC1 – PC8, ping, n = 100	+	0	0

	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	-		
	PC3 – PC7, Traffic Generator, N = 870, T = 0.1	+		
5	PC1 – PC8, ping, n = 100	+	0	0
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	+		
	PC2 – PC7, Traffic Generator, N = 320, T = 0.001	+		

4. Создание статической маршрутизации

Для выполнения данной задачи в **Cisco Packet Tracer** была создана новая сеть, включающая в себя маршрутизаторы, коммутаторы, а также рабочие станции (рисунок 9). Все устройства были переименованы в соответствии с заданием.

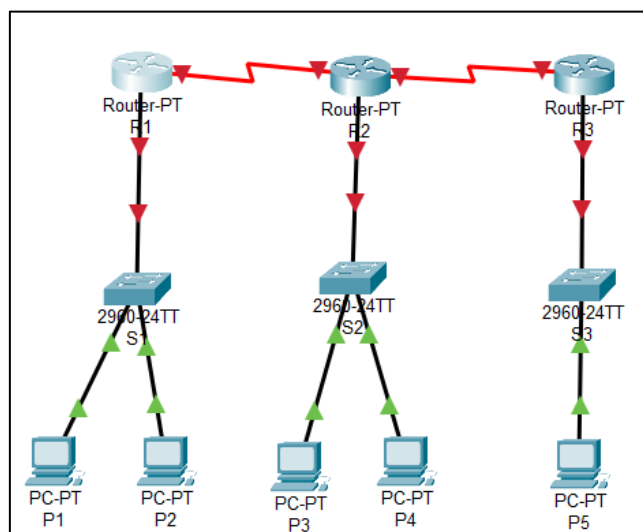


Рисунок 9 - новая созданная сеть

Первым делом было необходимо задать IP адреса сетевым интерфейсам маршрутизаторов. Это было сделано с помощью ввода команд в CLI, в частности с помощью команды **ip address** (рисунок 10).


```

R2(config)#interface Serial2/0
R2(config-if)#ip address 172.16.1.2 255.255.0.0
R2(config-if)#no shutdown

R2(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

R2(config-if)#exit
R2(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

R2(config)#interface fa0/0
R2(config)#
R2(config)#interface Serial3/0
R2(config-if)#interface fa0/0
R2(config-if)#ip address 192.168.2.1 255.255.255.0
R2(config-if)#no shutdown

R2(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

```

Рисунок 10 - настройка IP адресов интерфейсов роутера R2

Затем то же самое было сделано и на коммутаторах, но уже с помощью команды **ip vlan** (рисунок 11).

```

S1>enable
S1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
S1(config)#interface vlan 1
S1(config-if)#ip address 192.168.1.254 255.255.255.0
S1(config-if)#no shutdown exit

S1(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up

S1(config-if)#exit

```

Рисунок 11 - настройка ip адреса интерфейса управления коммутатора S1

Затем IP-адреса также были даны и клиентским компьютерам, однако это уже было сделано с помощью графического интерфейса **Cisco Packet Tracer**. Таким образом, после настройки ip-адресов интерфейсов всех устройств, была установлена связь между соседними маршрутизаторами на физическом и канальном уровнях по последовательному сетевому интерфейсу. Чтобы проверить это, была использована утилита **ping**. Как видно (рисунок 12), маршрутизатор R2 получает пакеты от маршрутизатора R1.

```

R1>ping 172.16.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/23/29 ms

```

Рисунок 12 - успешная доставка пакетов от R1 к R2

При этом устройства также могут также пересылать данные соседним объектам своей сети. Например, с маршрутизатора R1 успешно доходят пакеты до хоста P1 (рисунок 13).

```
R1>ping 192.168.1.11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.11, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Рисунок 13 - доставка данных между объектами одной сети

Далее с помощью команды `ip route` были настроены статические маршруты на маршрутизаторы R2 к сетям локальных компьютеров P1 и P3 (рисунок 14).

```
R2>en
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#ip route 192.168.1.0 255.255.255.0 172.16.1.1
R2(config)#ip route 192.168.2.0 255.255.255.0 10.1.1.2
R2(config)#exit
```

Рисунок 14 - настройка статических маршрутов

Затем на маршрутизаторе R1 был задан маршрут по умолчанию к сетям других локальных компьютеров (рисунок 15). Трафик будет идти на маршрутизатор R2, где уже будет решаться, куда он пойдет дальше. То же самое было сделано и на маршрутизаторе R3.

```
R1>enable
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip route 0.0.0.0 0.0.0.0 172.16.1.2
R1(config)#exit
```

Рисунок 15 - настройка маршрута по умолчанию

После выполнения всех этих действия пересылка пакетов по протоколу IP стала доступно по всем объектам сети. Например, утилита `ping` успешно работает с компьютера P1 до компьютера P3, находящегося в другой сети (рисунок 16).

```
C:\>ping 192.168.2.12

Pinging 192.168.2.12 with 32 bytes of data:

Reply from 192.168.2.12: bytes=32 time=16ms TTL=126
Reply from 192.168.2.12: bytes=32 time=19ms TTL=126
Reply from 192.168.2.12: bytes=32 time=13ms TTL=126
Reply from 192.168.2.12: bytes=32 time=5ms TTL=126

Ping statistics for 192.168.2.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 19ms, Average = 13ms
```

Рисунок 16 - успешная доставка пакетов из одной сети в другую

Затем была выполнена команда **ping** между устройствами P1 и P5, а в режиме симуляции было рассмотрено, как происходит процесс обмена данными между узлами. Сначала пакет был отправлен на коммутатор S1, от него к маршрутизатору R1. После пакет пошел к R2, R3, коммутатору S3 и, наконец,

компьютеру P5. Затем ответ от устройства P5 был направлен устройству P1 по такому же маршруту.

В этом процессе важную роль сыграл протокол ARP. Когда компьютер (P1) пытается отправить ICMP-запрос на другой компьютер (P5), он должен знать MAC-адрес шлюза (например, R1). P1 отправляет ARP-запрос, чтобы его узнать. Шлюз получает этот запрос и отвечает своим MAC-адресом – это ARP-ответ. После этого P1 знает, куда отправлять ICMP-запрос. Он отправляет пакет на MAC-адрес шлюза. Шлюз передает этот пакет дальше к P5, который также использует ARP, чтобы узнать MAC-адрес своего шлюза.

Именно из-за протокола ARP при первой работе утилиты **ping** один или несколько запросов из 4 не проходят. При доставке первого запроса компьютер не знает MAC-адрес устройства, которому хочет послать пакет, поэтому вместо доставки пакета начинает действовать протокол ARP, пытаясь найти MAC-адрес нужного устройства.

5. Настройка динамической маршрутизации

Для выполнения данного задания первым делом было необходимо просмотреть существующие таблицы маршрутизации на маршрутизаторах. Это можно сделать, воспользовавшись CLI и вбив туда команду **show ip route**. Как видно из рисунка 17, для маршрутизатора R2 успешно вывелась таблица маршрутизации с указанными маршрутами, часть из которых была добавлена при выполнении предыдущего задания лабораторной работы.

```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, Serial3/0
C    172.16.0.0/16 is directly connected, Serial2/0
S    192.168.1.0/24 [1/0] via 172.16.1.1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
S    192.168.3.0/24 [1/0] via 10.1.1.2
```

Рисунок 17 - таблица маршрутизации R2

Далее не всех маршрутизаторах была включена поддержка протокола RIP с помощью команды **router rip**. С помощью команды **network** к протоколу были подключены все необходимые сети. После этого таблицы маршрутизации были обновлены. Например, на рисунке 18 видно, что в таблице маршрутизации

маршрутизатора R3 была добавлена запись с кодом R, что означает, что запись была получена в ходе динамической маршрутизации.

```
C 10.0.0.0/8 is directly connected, Serial3/0
R 172.16.0.0/16 [120/1] via 10.1.1.1, 00:00:08, Serial3/0
S 192.168.1.0/24 [1/0] via 10.1.1.1
S 192.168.2.0/24 [1/0] via 10.1.1.1
C 192.168.3.0/24 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 10.1.1.1
```

Рисунок 18 - обновленная таблица маршрутизации R3

При этом пересылка данных по протоколу IP свободно возможно по всем объектам сети, что было проверено с помощью утилиты **ping**, результат работы которой представлен на рисунке 19.

```
C:\>ping 192.168.2.11

Pinging 192.168.2.11 with 32 bytes of data:

Reply from 192.168.2.11: bytes=32 time=12ms TTL=126
Reply from 192.168.2.11: bytes=32 time=8ms TTL=126
Reply from 192.168.2.11: bytes=32 time=7ms TTL=126
Reply from 192.168.2.11: bytes=32 time=2ms TTL=126

Ping statistics for 192.168.2.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 12ms, Average = 7ms
```

Рисунок 19 - проверка работы пересылки данных

Вывод:

В ходе выполнения лабораторной работы в программе Cisco Packet Tracer была смоделирована сеть с топологией звезда, а также было выполнены все задачи лабораторной работы: была создана сама сеть, в ней был сформирован трафик, а также в ней была произведена замена концентраторов на коммутаторы. Также была создана еще одна сеть, в которой была настроена статическая маршрутизация пакетов и динамическая маршрутизация по протоколу RIP. Помимо этого, были даны ответы на контрольные вопросы, представленные ниже.

Ответы на контрольные вопросы:

Вопросы 1

1. Как видно из результатов формирования трафика сети, в структурированной сети количество потерянных пакетов по сравнению с сетью общим доменом коллизий не только заметно уменьшилось, но и вовсе стало равняться нулю. Это связано с тем, что концентраторы, перенаправляющие полученные пакеты на все остальные порты, были заменены на коммутаторы, транслирующие

пакеты только на нужные порты, вследствие чего нагрузка на сеть заметно уменьшилась.

2. В замене концентраторов Hub3 и Hub4 на коммутаторы смысла нет, так как за ними пока что стоит только по одному устройству и, соответственно, заняты всего 2 порта. Получается, что данные концентраторы и так перенаправляют пакеты только на 1 оставшийся порт, и поэтому коммутаторы будут излишними. В замене будет смысл тогда, когда коммутаторы будут подсоединены к большому количеству устройств и нагрузка на сеть возрастет.

Вопросы 2

1. Основной принцип работы протоколов маршрутизации в сетях с коммутацией пакетов заключается в том, что маршрутизаторы обмениваются информацией о сетевых маршрутах друг с другом через протоколы маршрутизации, что позволяет им эффективно определять наилучший путь для передачи данных. Такие протоколы могут быть статическими или динамическими.

2. Метрика маршрута — это числовое значение, которое используется протоколами маршрутизации для оценки "стоимости" маршрута к определенной сети, которая помогает маршрутизаторам определять наилучший путь для передачи данных. Метрика отражает такие параметры маршрута, как дальность, надежность, пропускная способность и т. д. При этом разные протоколы маршрутизации используют разные методы для расчета метрик.

3. MPLS — это технология сетевой передачи, которая позволяет эффективно управлять трафиком и маршрутизацией в сетях с коммутацией пакетов. Она позволяет повысить скорость передачи данных и улучшить качество обслуживания в сети. Принцип работы такой передачи данных заключается в том, что при входе в MPLS-сеть каждому пакету присваивается числовая метка, в зависимости от которой определяется дальнейший маршрут пакета, и каждый последующий в сети маршрутизатор определяет следующий хоп, основываясь на метке. Этот процесс значительно ускоряет процесс маршрутизации в сети, ведь маршрутизаторы не смотрят на IP-адреса.