

## Лабораторная работа 2-1. Моделирование сети с топологией звезда на базе концентратора

*Звезда* — базовая топология компьютерной сети, в которой все компьютеры сети присоединены к центральному узлу, образуя физический сегмент сети. Центральным узлом выступает концентратор, коммутатор или ПК. Рабочая станция, с которой необходимо передать данные, отправляет их на концентратор. В определённый момент времени только одна машина в сети может пересылать данные, если на концентратор одновременно приходят два пакета, обе посылки оказываются не принятыми и отправителям нужно будет подождать случайный промежуток времени, чтобы возобновить передачу данных. Этот недостаток отсутствует на сетевом устройстве более высокого уровня — коммутаторе, который, в отличие от концентратора, подающего пакет на все порты, подает лишь на определенный порт — получателю. Одновременно может быть передано несколько пакетов. Сколько — зависит от коммутатора. Достоинства звезды: *выход* из строя одной рабочей станции не отражается на работе всей сети в целом; лёгкий *поиск* неисправностей и обрывов в сети; высокая *производительность* сети (при условии правильного проектирования); гибкие возможности администрирования. Недостатки звезды: *выход* из строя центрального концентратора обернётся неработоспособностью сети (или сегмента сети) в целом; для прокладки сети зачастую требуется больше кабеля, чем для большинства других топологий; число рабочих станций в сети (или сегменте сети) ограничено количеством портов в центральном концентраторе.

В данном примере мы с помощью программного симулятора *Packet Tracer* построим сеть с топологией *Звезда* на базе концентратора (рис. 2.1) и изучим ряд новых приемов работы в этой программе.

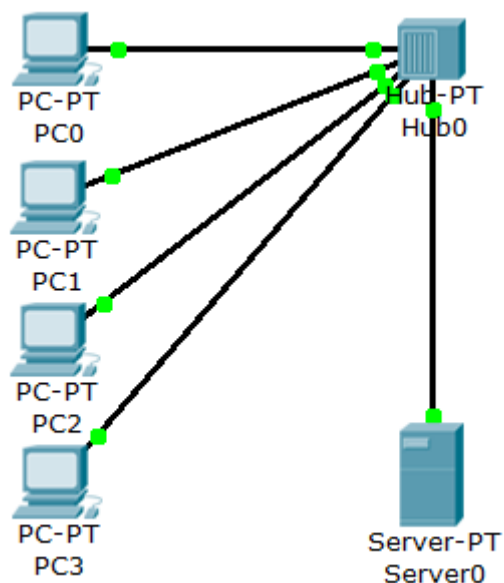


Рис. 2.1. Моделирование сети с топологией звезда на базе концентратора

### В рабочей области komponуем узлы сети

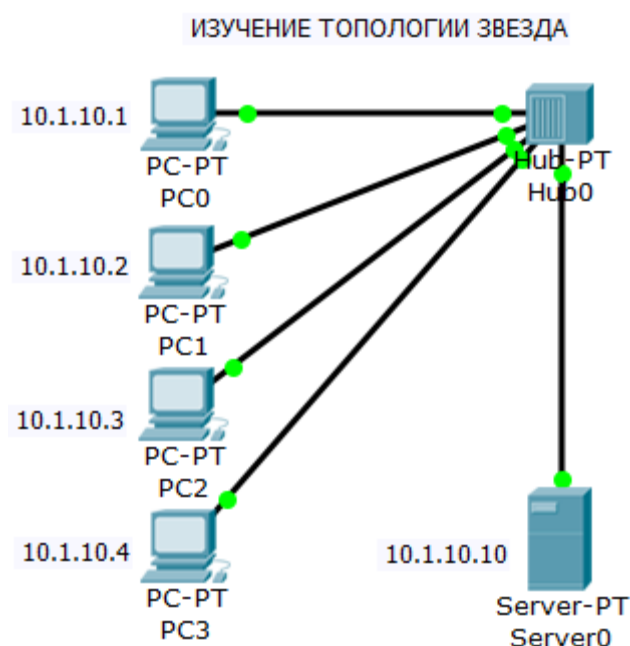
Выбираем тип оборудования **Hub's** (Концентраторы). В меню "список устройств данного типа оборудования" выбираем конкретный концентратор - *Hub-PT* и перетаскиваем его мышью в рабочую область программы. Далее выбираем тип устройства **End Devices** (Конечные устройства) и в дополнительном меню выбираем настольный компьютер *PC-PT* и перетаскиваем его мышью в рабочую область программы. Таким образом, устанавливаем ещё три компьютера и один сервер. Для подключения компьютеров и сервера к концентратору выбираем новый тип устройств **Connections** (Соединения), далее

выбираем **Copper Straight-Through** (Медный прямой) тип кабеля. Чтобы соединить сетевую карту

компьютера с портом *Hub-a*, необходимо щелкнуть левой клавишей мыши по нужному компьютеру. В открывшемся графическом *меню* выбрать *порт* FastEthernet0 и протянуть *кабель* от ПК к концентратору, где в аналогичном *меню* выбрать любой свободный *порт Fast Ethernet* концентратора. При этом желательно всегда придерживаться следующего правила: для сервера выбираем 0-й *порт*, для PC1 - 1й *порт*, для PC2 - 2й *порт* и так далее. Назначаем узлам сети *IP* адреса и маску. Для этого двойным щелчком открываем нужный *компьютер*, далее Config (Конфигурация)- **Interface** (Интерфейс)- **FastEthernet0**. В группе параметров **IP Configuration** (Настройка *IP*) должен быть активирован *переключатель* **Static** (Статический) в *поле* **IP Address** необходимо ввести *IP-адрес* компьютера, *маска* появится автоматически. **Port status** (Состояние порта) – **On** (Вкл).

## Инструмент создания заметок Place Note

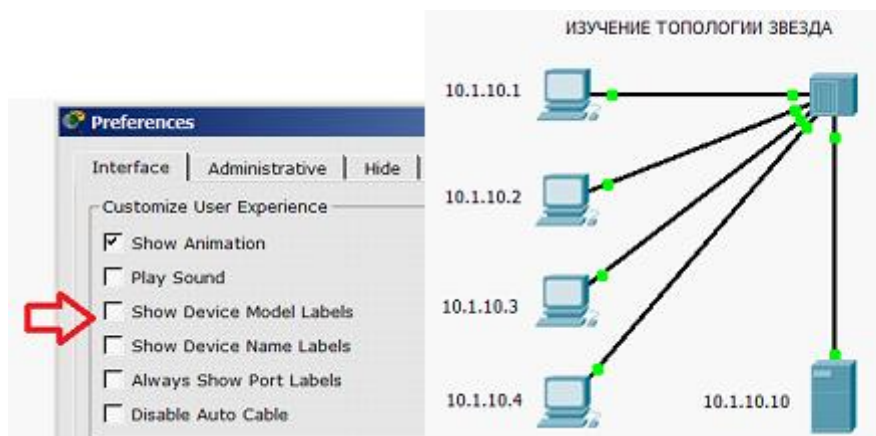
Используя инструмент создания заметок **Place Note** (клавиша N), подписываем все *IP* устройств, а вверху рабочей области создаем заголовок нашего проекта "**Изучение топологии звезда**" - рис. 2.2.



**Рис. 2.2.** Используем инструмент Place Note (Заметка)  
**Совет**

*IP* адреса следует скопировать из окна **Config** (Конфигурация). При этом активируйте инструмент **Place Note** (Заметка).

С целью исключения нагромождения рабочей области надписями, уберем надписи (метки) типов устройств: откроем *меню* **Options** (Опции) в верхней части окна *Packet Tracer*, затем в ниспадающем списке выберем *пункт* **Preferens** (Настройки), а в диалоговом окне снимем флажок **Show device model labels** (Показать модели устройств) - рис. 2.2.



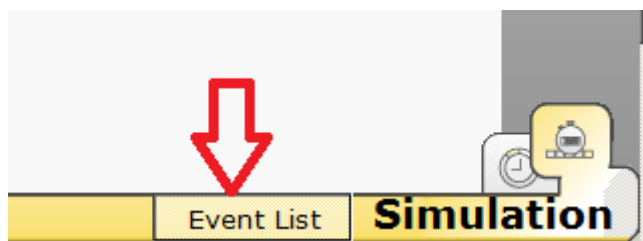
**Рис. 2.2.** Дезактивируем флажок Show device model labels

Для проверки работоспособности сети отправим с компьютера на другой ПК тестовый сигнал *ping* и переключимся в режим **Simulation** (Симуляция). В окне **Event list** (Список событий), с помощью кнопки **Edit filters** (Изменить фильтры), сначала очистите фильтры от всех типов сигнала, а затем установим тип контроля сигнала: только *ICMP*.

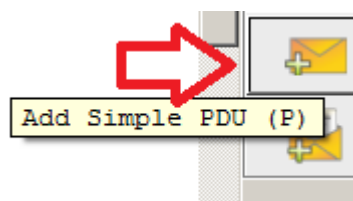
#### Примечание

Я не привожу иллюстраций для этих команд, поскольку они уже приводились в предыдущих главах курса.

Далее окно **Event list** (Список событий) закрываем ( рис. 2.4).



**Рис. 2.4.** Кнопка Event list (Список событий)



В правой части окна, в графическом *меню* выбираем щелчками мыши, устанавливаем его на ПК - выбираем источник сигнала (например, PC3) и, затем, на узле назначения (пусть это будет *сервер*). Нажимая на кнопку **Capture / Forward** (Захват/Вперед) наблюдаем пошаговое продвижение пакета *PDU* – рис. 2.5

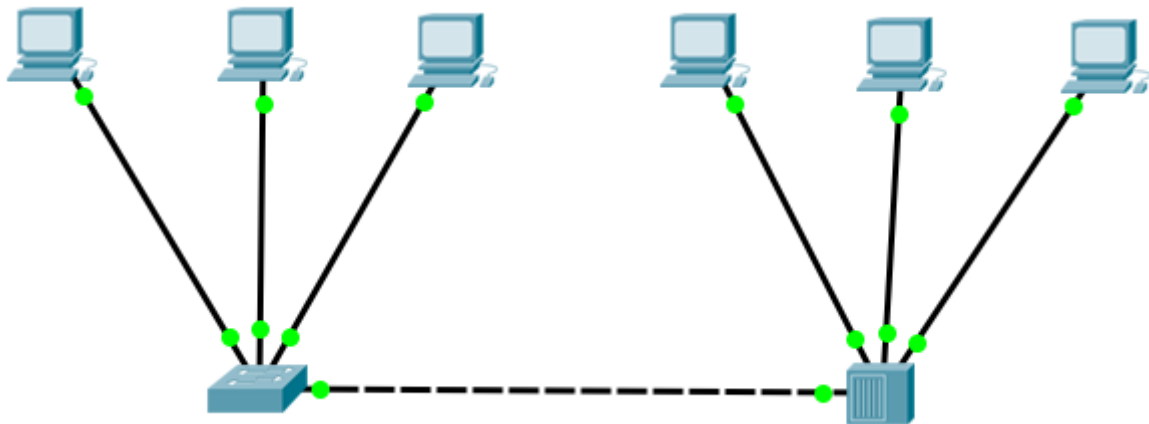
Fire	Last Status	Source	Destination	Type	Color	Time(se	Periodic	Num	Edit	Delete
	Successful	PC3	Server0	ICMP		0.000	N	0	(edit)	

**Рис. 2.5.** Успешное прохождение пакетов по сети  
**Новый термин**

*PDU* - обобщённое название фрагмента данных на разных уровнях Модели OSI: кадр Ethernet, ip-пакет, udp-датаграмма, tcp-сегмент и т. д.

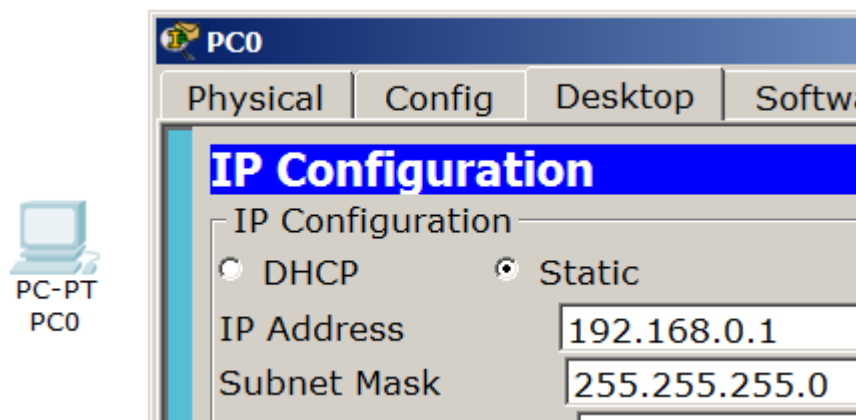
## Полезные приемы работы в СРТ

Предположим, что вам нужно спроектировать и настроить следующую *сеть* ( рис. 2.6). Рассмотрим, как можно ускорить и упростить этот процесс.



**Рис. 2.6.** Постановка задачи

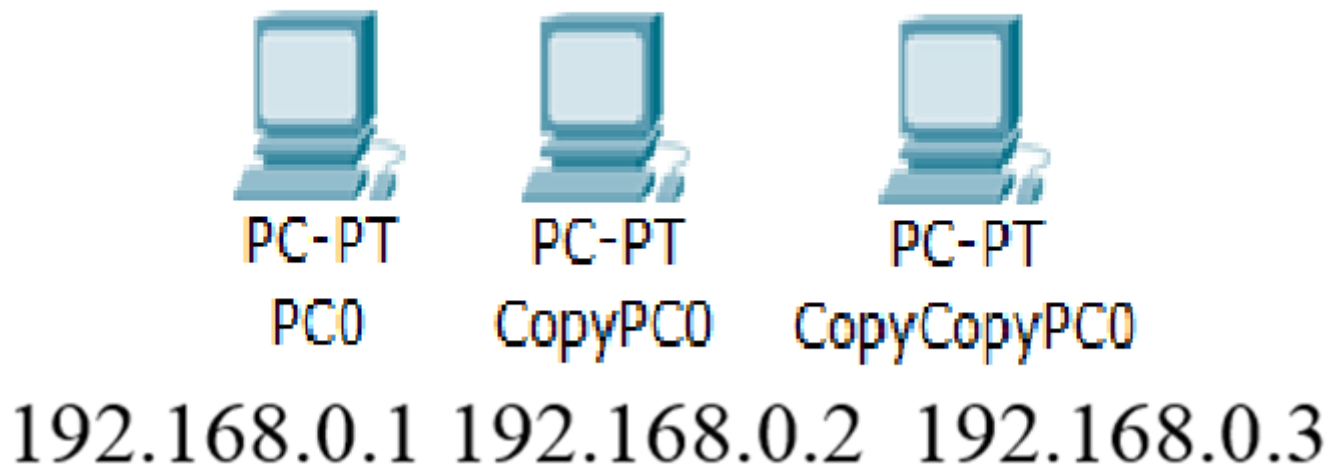
Поместите в рабочую область первый ПК (это будет *PC*) и настройте его ( рис. 2.7).



**Рис. 2.7.** Настраиваем PC0

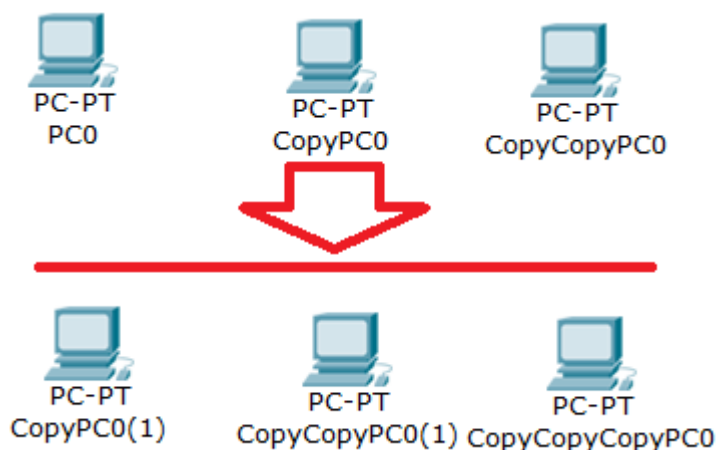
Удерживая клавишу Ctrl скопируйте этот ПК несколько раз и настройте остальные адреса ПК, меняя только последнюю цифру *IP* адреса ( рис. 2.8).

маска 255.255.255.0



**Рис. 2.8.** Быстрое создание и настройка трех ПК

Далее скопируйте, удерживая **Ctrl** сразу три ПК и настройте их также, меняя только последнюю цифру *IP* адреса ( рис. 2.9).



**Рис. 2.9.** Копируем все три ПК сразу

Добавление свитча и *хаб* делаем традиционно, а подключение кабеля - автоматическое.

Сначала немного теории. *Hub* работает на 1м уровне модели *OSI* и отправляет информацию во все порты, кроме порта – источника. *Switch* работает на 2м уровне *OSI* и отправляет информацию только в *порт* назначения за счет использования таблицы *MAC* адресов хостов. В сетях *IP* существует 3 основных способа передачи данных: *Unicast*, *Broadcast*, *Multicast*.

- *Unicast* (юникаст) – процесс отправки пакета от одного хоста к другому хосту.
- *Multicast* (мультикаст) – процесс отправки пакета от одного хоста к некоторой ограниченной группе хостов.
- *Broadcast* (бродкаст) – процесс отправки пакета от одного хоста ко всем хостам в сети.

В некоторых случаях *switch* может отправлять фреймы как *hub*, например, если *фрейм* бродкастовый (*broadcast* - *широковещание*) или unknown *unicast* (неизвестному единственному адресату).

## Лабораторная работа 2-2. Моделирование сети с топологией звезда на базе коммутатора

Работу сети с топологией *звезда* на базе концентратора мы уже изучили. Теперь рассмотрим аналогичную *сеть* на базе коммутатора ( рис. 4.1).

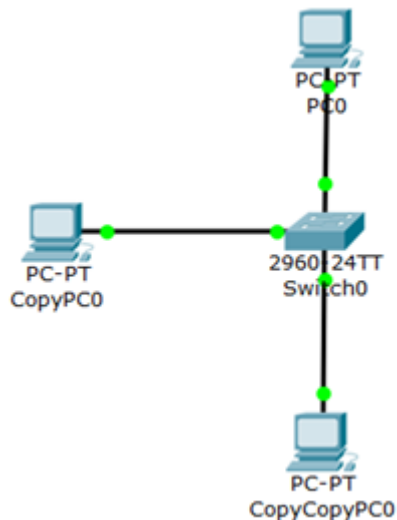


Рис. 4.1. Звезда на базе коммутатора модели 2960

На вкладке *Physical* вы можете посмотреть вид коммутатора, имеющего 24 порта *Fast Ethernet* и 2 порта *Gigabit Ethernet* ( рис. 4.2).

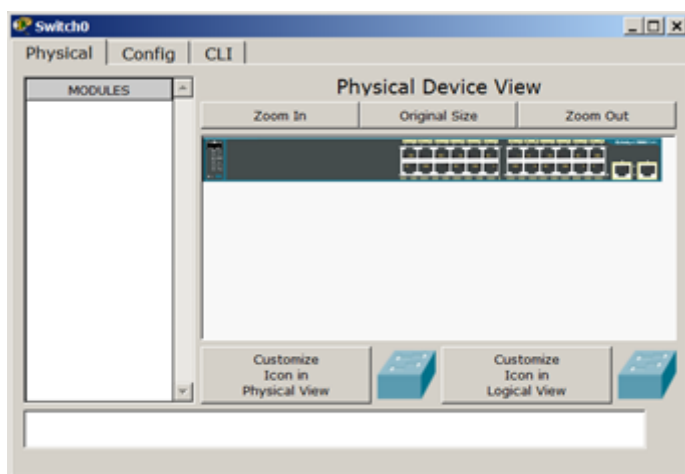



Рис. 4.2. Физический внешний вид коммутатора модели 2960

IPv4	IPv6	Misc
<input type="checkbox"/> ARP	<input type="checkbox"/> BGP	<input type="checkbox"/> DHCP
<input type="checkbox"/> DNS	<input type="checkbox"/> EIGRP	<input type="checkbox"/> HSRP
<input checked="" type="checkbox"/> ICMP	<input type="checkbox"/> OSPF	<input type="checkbox"/> RIP

В режиме *Simulation* настроим фильтры

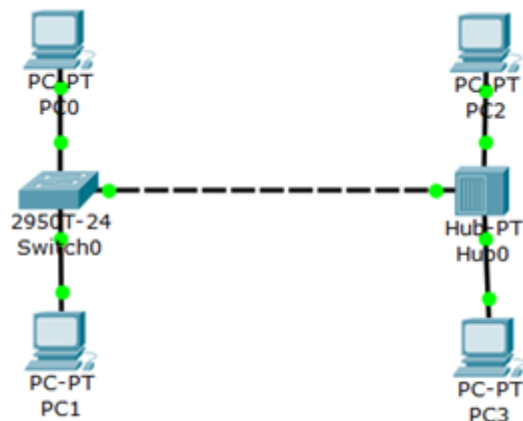
и с помощью

функции  просмотрим прохождение пакета между двумя ПК через *коммутатор*. Как видим, маршруты пакетов концентраторе и коммутаторе будут разными: как в прямом, так и в обратном направлении *хаб* отправляет всем, а *коммутатор* – только одному.

### Задание 4.1

Произведите проектирование локальной сети из хаба, коммутатора и 4х ПК

Сеть, которую необходимо спроектировать представлена на рис. 4.2.



**Рис. 4.2.** Проектируемая сеть

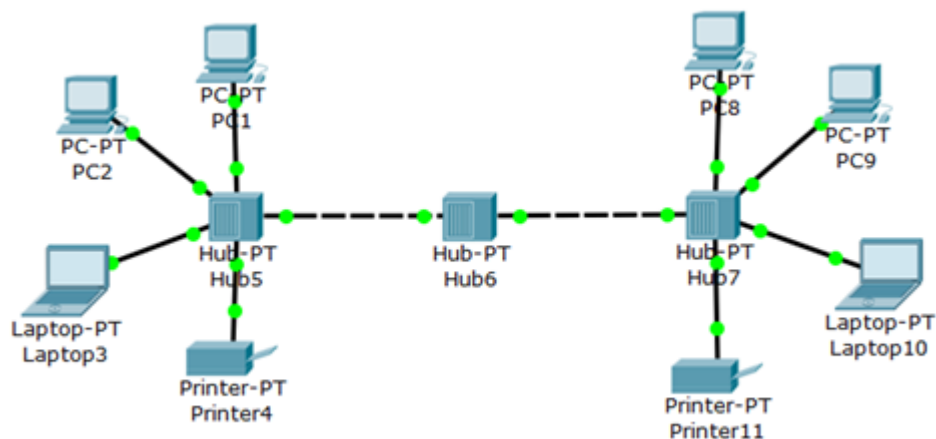
Произведите настройку и диагностику этой сети двумя способами (утилитой *ping* и в окне списка *PDU*). Убедитесь в успешности работы сети в режиме симуляции.

#### Примечание

Перед выполнением симуляции необходимо задать фильтрацию пакетов. Для этого нужно нажать на кнопку "Изменить фильтры", откроется окно, в котором нужно оставить только протоколы "ICMP" и "ARP". Кнопка "Авто захват/Воспроизведение" подразумевает моделирование всего *ping*-процесса в едином процессе, тогда как "Захват/Вперед" позволяет отображать его пошагово.

## Лабораторная работа 2-2. Исследование качества передачи трафика по сети

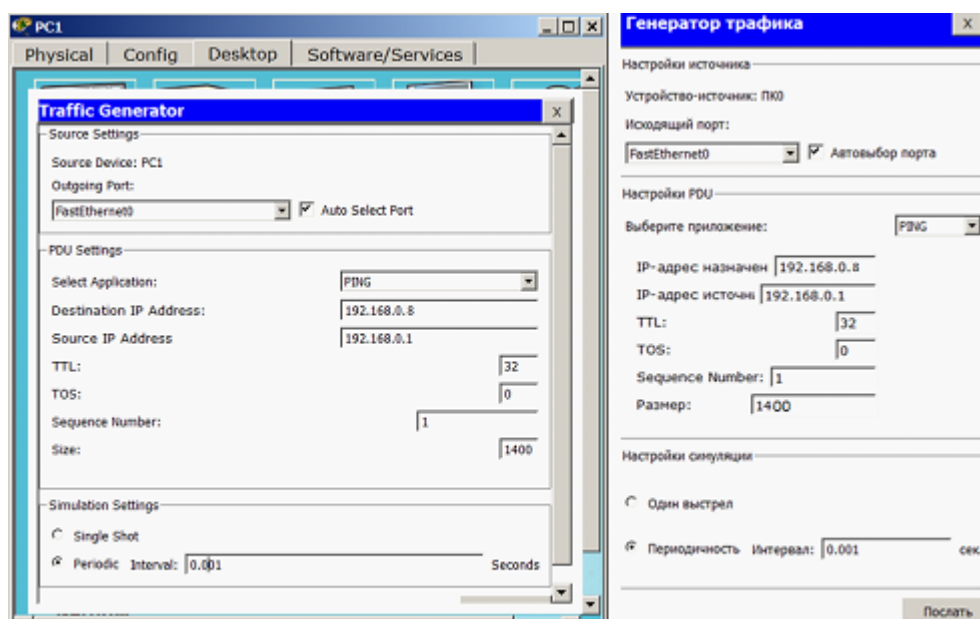
При исследовании пропускной способности *ЛВС* (качества передачи трафика по сети) желательно увеличить *размер пакета* и отправлять запросы с коротким интервалом времени, не ожидая ответа от удаленного узла, для того, чтобы создать серьезную нагрузку на *сеть*. Однако, утилита *ping* не позволяет отправлять эхо-запрос без получения эхо-ответа на предыдущий запрос и до истечения времени ожидания. Поэтому для организации существенного трафика воспользуемся программой *Traffic Generator*. Для работы создайте и настройте следующую *сеть* (рис. 4.4).



**Рис. 4.4.** Топология сети для нашей работы

## Первое знакомство с Traffic Generator

В окне управления PC1 во вкладке Desktop выберите *приложение Traffic Generator* и задайте настройки, как на рис. 4.5 для передачи трафика от PC1 на PC8. Для ясности я рядом с английской версией окна разместил тот же текст в русской версии программы CPT.



**Рис. 4.5.** Настройка генератора трафика (Вариант трафика от PC1 до PC8)

Итак, при помощи протокола *ICMP* мы сформировали трафик между компьютерами PC1 с адресом 192.168.0.1 и PC8 с адресом 192.168.0.8. При этом в разделе **Source Settings** (Настройки источника) необходимо установить флажок **Auto Select Port** (Автовыбор порта), а в разделе **PDU Settings** (настройки IP-пакета) задать следующие значения параметров этого поля:

**Select application:** *PING*

**Destination: IPAddress:** 192.168.0.8 (адресполучателя);

**Source IP Address:** 192.168.0.1 (*адрес* отправителя);

**TTL:** 32 (время жизни пакета);

**TOS:** 0 (тип обслуживания, "0" - обычный, без приоритета);

**Sequence Number:** 1 (начальное *значение* счетчика пакетов);

**Size:** 1400 (размер поля данных пакета в байтах);

**Simulations Settings** - здесь необходимо активировать *переключатель*;

**Periodic Interval:** 0.3 Seconds (период повторения пакетов)

### Внимание

Не обязательно использовать те настройки, которые задал автор. Можете указать свои, например, Size: 1500, PeriodicInterval: 0.5 Seconds. Однако, если неверно укажете IP источника, то генератор работать не будет.



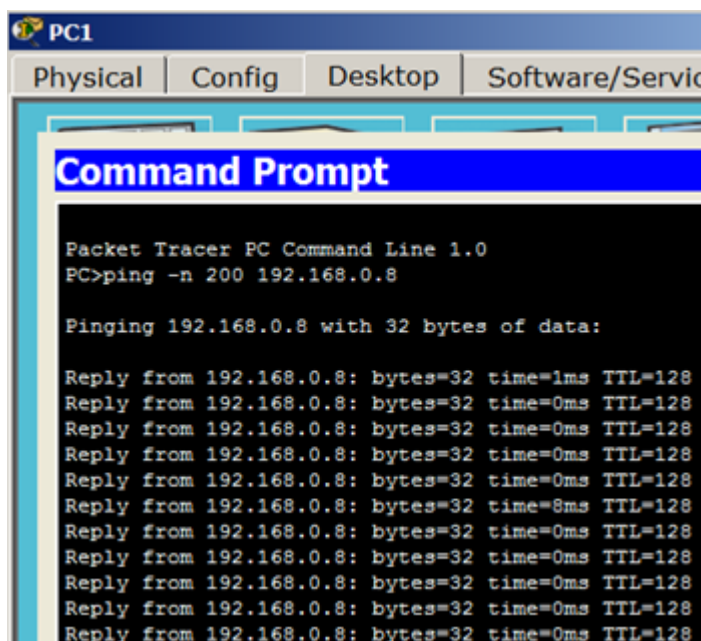
После нажатия на кнопку **Send** (Послать) между PC1 и PC8 начнется *активный обмен данными*. Не закрывайте окно генератора трафика настройки, чтобы не прервать *поток* трафика - лампочки должны постоянно мигать!

### Новый термин

TTL - время жизни пакета. Наличие этого параметра не позволяет пакету бесконечно ходить по сети. TTL уменьшается на единицу на каждом узле (хопе), через который проходит пакет.

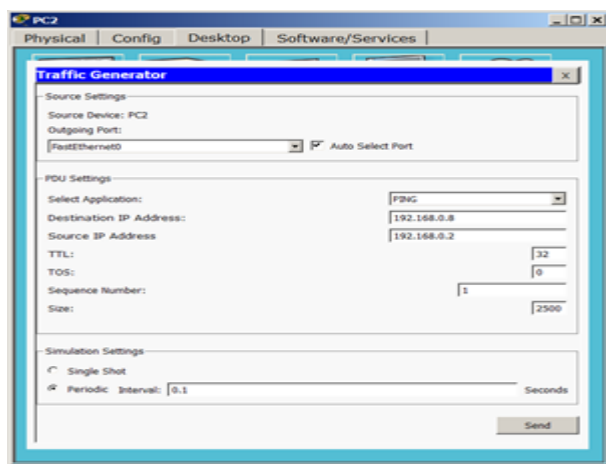
## Исследование качества работы сети

Для оценки качества работы сети передадим *поток* пакетов между PC1 и PC8 при помощи команды `ping -n 200 192.168.0.8` и будем оценивать качество работы сети по числу потерянных пакетов. Параметр "-n" позволяет задать количество передаваемых эхо-запросов (у нас их 200) – рис. 4.6.



**Рис. 4.6.** Отправляем 200 пакетов на PC8

Одновременно с пингом, нагрузите *сеть*, включив *генератор* трафика на компьютере PC2 (узел назначения – PC8, размер поля данных–2500 *байт*, период повторения передачи - 0,1 сек. – рис. 4.7.



**Рис. 4.7.** Увеличиваем нагрузку на сеть

Для оценки качества работы сети - зафиксируйте число потерянных пакетов ( рис. 4.8).

```

Reply from 192.168.0.8: bytes=32 time=0ms TTL=128
Reply from 192.168.0.8: bytes=32 time=0ms TTL=128
Reply from 192.168.0.8: bytes=32 time=0ms TTL=128
Reply from 192.168.0.8: bytes=32 time=1ms TTL=128
Request timed out.
Reply from 192.168.0.8: bytes=32 time=0ms TTL=128
Reply from 192.168.0.8: bytes=32 time=0ms TTL=128
Reply from 192.168.0.8: bytes=32 time=0ms TTL=128
Reply from 192.168.0.8: bytes=32 time=0ms TTL=128
Reply from 192.168.0.8: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.8:
    Packets: Sent = 200, Received = 194, Lost = 6 (3% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 4294967295ms, Average = 0ms

```

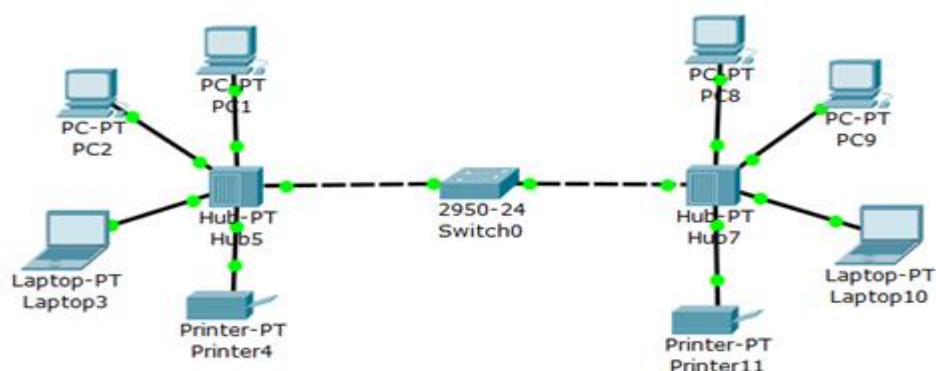
**Рис. 4.8.** Потеряно 6 пакетов  
**Примечание**

Как вариант можно было бы загрузить сеть путем организации еще одного потока трафика между какими-либо узлами сети, например, включив генератор трафика еще на ноутбуке PC2.

В заключение этой части нашей работы остановите *Traffic Generator* на всех узлах, нажав кнопку **Stop**.

## Повышение пропускной способности локальной вычислительной сети

Проверим тот факт, что установка коммутаторов вместо хабов устраняет возможность возникновения коллизий между пакетами пользователей сети. Замените центральный концентратор на *коммутатор* (рис. 4.9). Немного подождите и убедитесь, что *сеть* находится в рабочем состоянии - все маркеры портов не красные, а зеленые.



**Рис. 4.9.** Топология сети при замене центрального концентратора на коммутатор

Снова задайте *поток* пакетов между PC1 и PC8 при помощи команды *ping -n 200 192.168.0.8* и включите *Traffic Generator* на PC2. Проследите работу нового варианта сети. Убедитесь, что за счет снижения паразитного трафика качество работы сети стало выше (

```

Ping statistics for 192.168.0.8:
    Packets: Sent = 200, Received = 199, Lost = 1 (1% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 4294967295ms, Average = 0ms

```

**Рис. 4..**  
)

```

Ping statistics for 192.168.0.8:
    Packets: Sent = 200, Received = 199, Lost = 1 (1% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 4294967295ms, Average = 0ms

```

**Рис. 4.10.** Потерян 1 пакет

