Основы компьютерных сетей. Тема №4. Сетевые устройства и виды применяемых кабелей / Хабр

habr.com/ru/articles/312340

Денис 3 ноября 2016 г.

Приветствую всех! Добрались мы до 4-ой темы. Поговорим сегодня про различные сетевые устройства и применяемые кабели. Узнаем, чем отличается коммутатор от маршрутизатора, что такое концентратор и многое другое. Приглашаю заинтересовавшихся под кат.

Содержание

В ранних статьях я писал о разных сетевых моделях, протоколах, службах. Но мало рассказал об устройствах, которые тесно с этим работают, и самое главное, чем они все отличаются друг от друга. Эти знания очень важны для сетевого инженера, поэтому закрою эту брешь.

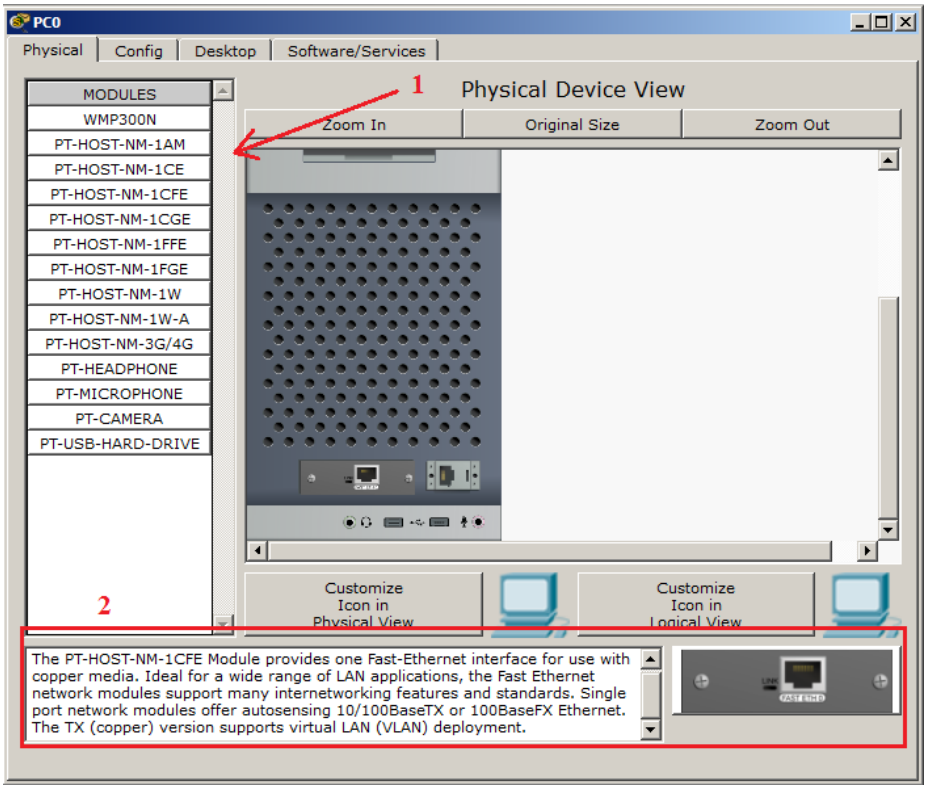
К счастью многие устройства доступны в Cisco Packet Tracer (версия 6.2), поэтому после каждого описанного устройства, я буду показывать это на практике.

Итак. Термин сетевые устройства применим к тем устройствам, которые подключены к сегменту сети и умеют принимать и/или передавать какие то данные. Самым простым и сразу приходящим в голову является **сетевая карта**.

Думаю все ее видели. Она стоит практически в каждом домашнем компьютере. Если не такая, то встроенная в материнскую плату.

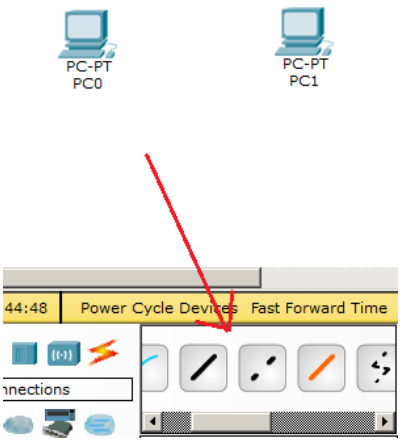
Раньше можно было встретить и другие ее виды. Например, как на картинке ниже.

Обратите внимание на вход для коаксиального кабеля, который раньше активно использовался. Сейчас такие уже редко где встретишь. Если интересно посмотреть на остальные виды, то в CPT есть очень хорошие примеры. Например, если кликнуть по компьютеру, то откроется такое окно.

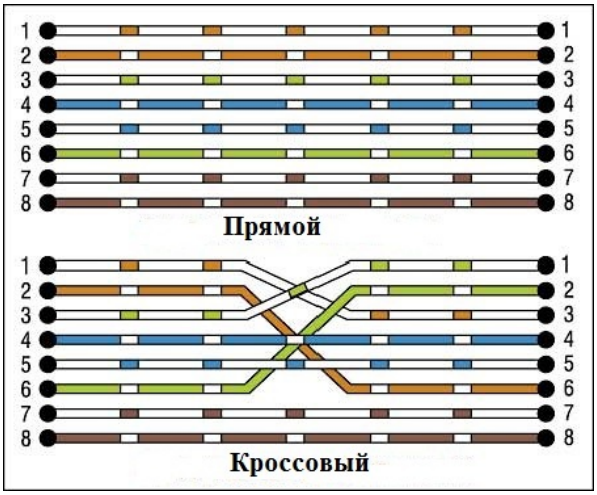


Слева расположено окошко модулей (обозначено на рисунке цифрой 1). По каждому модулю есть краткое описание и как выглядит (обозначено на рисунке цифрой 2). Например, я кликнул на модуль PT-HOST-NM-1CFE. Это сетевая карта, которая работает по технологии Fast-Ethernet и предназначена для работы с витой парой. Может работать на скорости 10 Мбит/с и 100 Мбит/с. Также поддерживает технологию VLAN, о которой будет следующая статья.

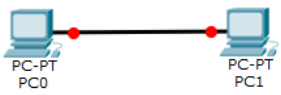
Работа такой карточки проста. У нее есть MAC-адрес (о чем я говорил ранее), который ей присвоили на заводе, и при помощи него она может общаться в сети с другими устройствами. Причем не обязательно ее соединять с коммутатором или другим устройством. Можно соединить ее с другой сетевой картой и организовать связь между ними. Таким образом раньше соединяли 2 компьютера в одной комнате. Это самое простое соединение. Давайте попробуем его организовать в CPT.



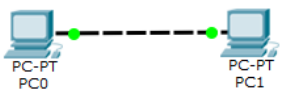
Немного отступлю от лабы, так как здесь есть важное замечание. Имеется 2 вида витой пары. Прямой (Straight-Through) и кроссовый (Cross-over). Прямой применяется, когда нужно соединить 2 разных устройства. Например, компьютер и коммутатор. А кроссовый — когда нужно соединить 2 компьютера, 2 коммутатора и т.д. Структурное различие в том, что пары проводов обжимаются по разному. Ниже привожу схему обжима.



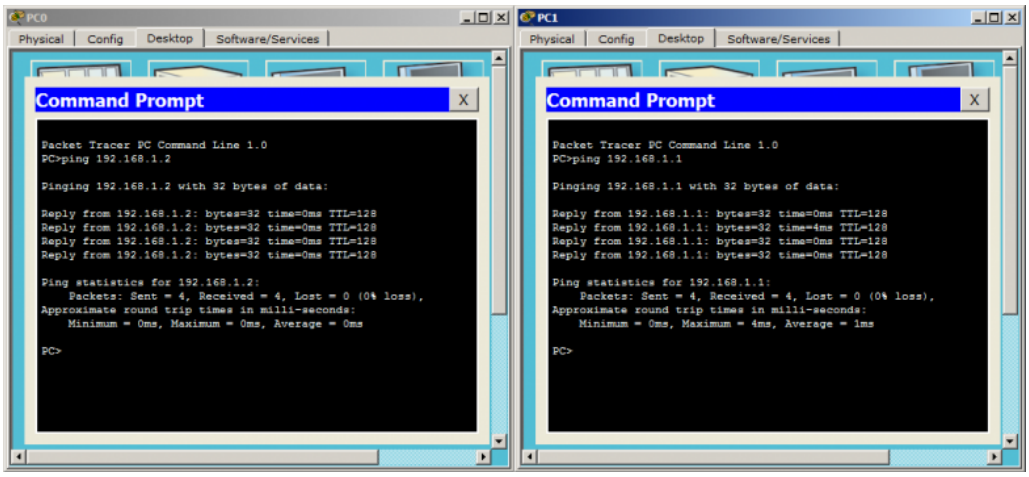
Соответственно, если соединить устройства не тем кабелем, то работать ничего не будет. Если вы только начинаете свой путь, то, возможно, уже не встретитесь с такой проблемой, так как большинство современных устройств поддерживают технологию **Auto-MDI(X)**. Эта технология позволяет понять устройству с кем оно соединено и в каком режиме ему работать. Причем достаточно, чтобы хотя бы один участник поддерживал ее для корректной работы. Но в любом случае это надо знать. Поэтому возьмите на заметку.
Возвращаемся к лабе. Предлагаю соединить 2 компьютера именно прямым кабелем, чтобы убедиться, что работать данная конструкция не будет.



И как видим, концы кабелей горят красным, что говорит о том, что соединение не работает.
Исправляем ошибку и подключим компьютеры кроссовым кабелем.



Наблюдаем зеленые огни. Радуемся и переходим к настройке IP-адресов. Первому присвоим адрес: 192.168.1.1 с маской: 255.255.255.0. Все остальное не важно. И, соответственно, второму компьютеру присвоим IP-адрес: 192.168.1.2 с аналогичной маской: 255.255.255.0. Проверим связь между ними.



Пинги успешны! Кому неохота соединять 2 компьютера, ссылка на скачивание.

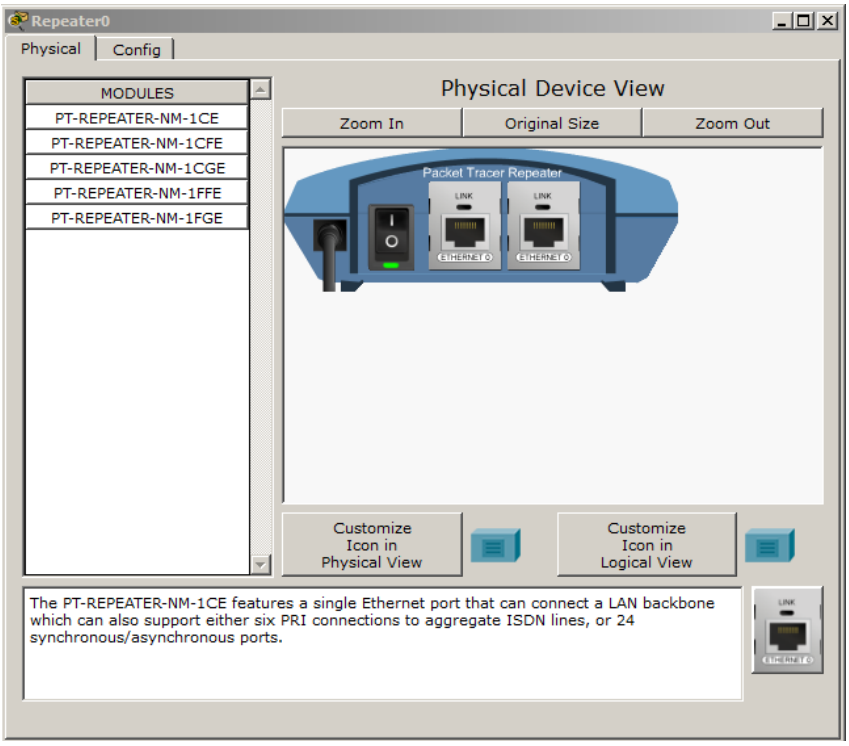
Следующее устройство на очереди — это **повторитель** или **repeater**.
Если рассматривать с точки зрения модели OSI, то данное устройство работает на первом уровне. То есть на физическом. Устройство очень простое. Основная задача — это усиление сигнала. Если вспомнить немного курс школьной физики, то у каждого кабеля есть предел затухания сигнала. Если мы говорим о витой паре, то ее максимальная длина может быть до 100 метров.@vilos) И для того, чтобы усилить сигнал, применяют данное устройство. Ethernet повторитель может усилить сигнал еще на 100 метров.

В связи с тем, что в настоящее время набрала популярность технология PoE (Power over Ethernet), то повторители используются в качестве удлинителей для удаленных устройств (например IP-камеры). На картинках ниже можно с ними познакомиться.

Повторитель старого образца (в настоящее время уже не производится)

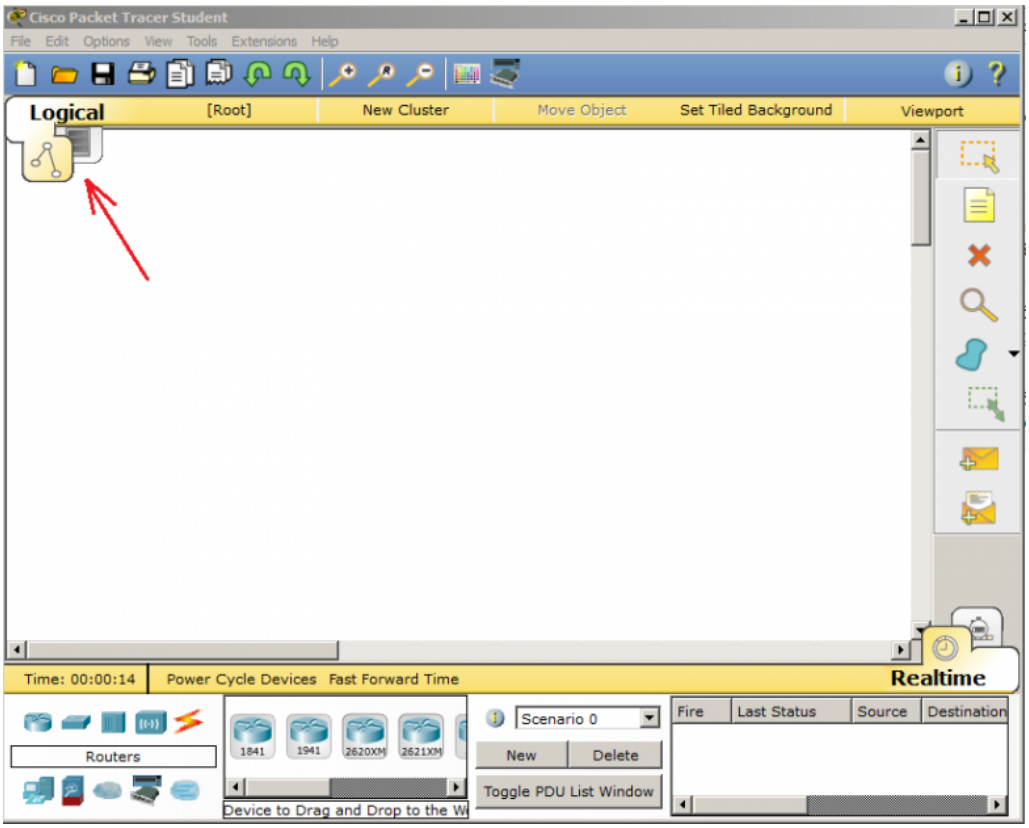
Один из современных повторителей.

В CPT оно присутствует, так что взглянем на него.

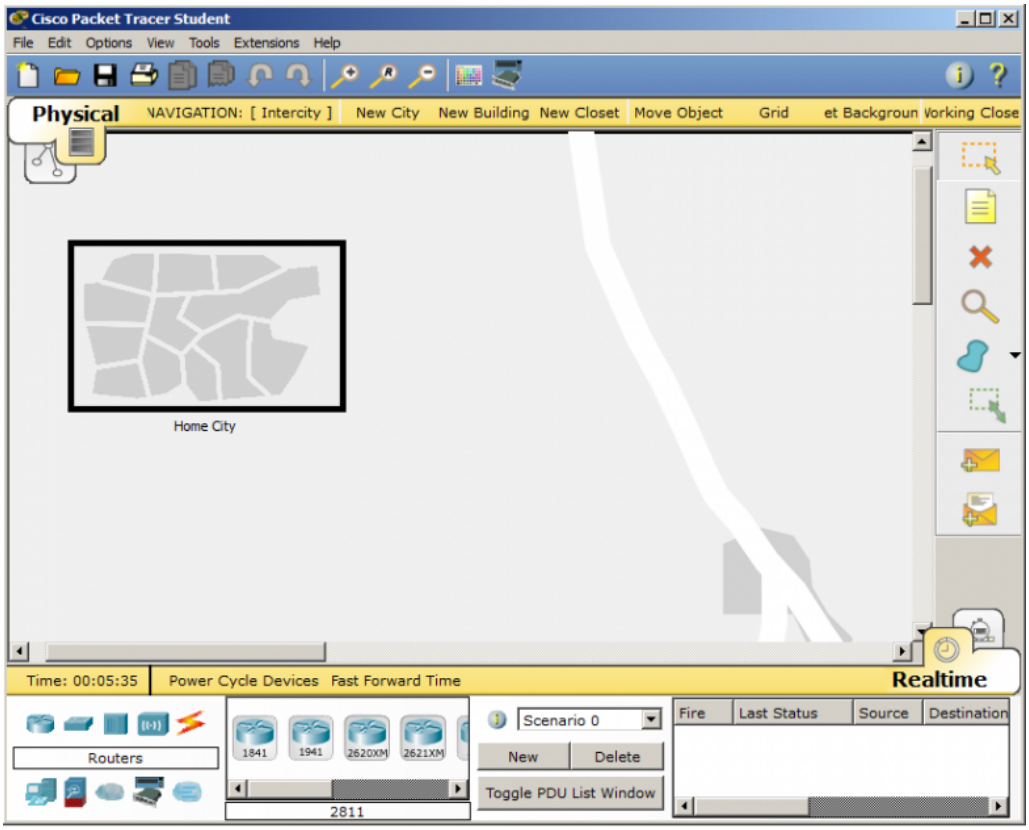


Можно его приблизить, отдалить, поменять ему интерфейсы. Все на ваше усмотрение. Я сымитирую ситуацию, когда у нас 2 компьютера находятся далеко друг от друга и соединены между собой при помощи повторителя.

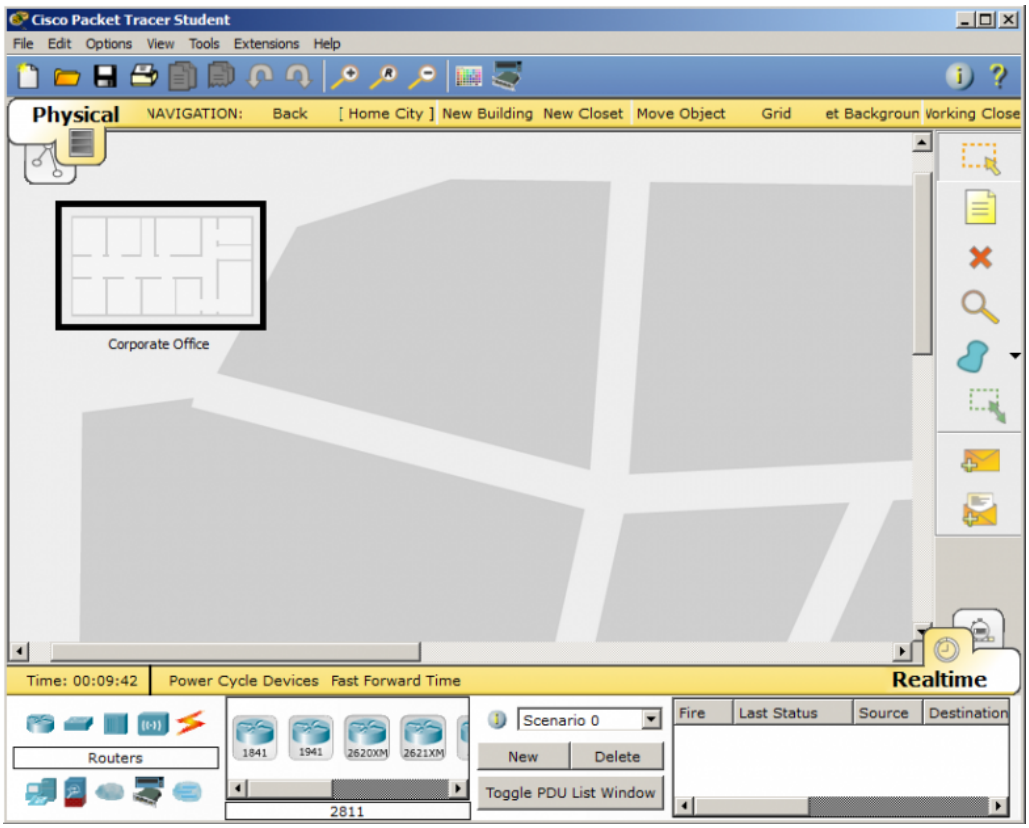
Хочу заметить важную функцию в CPT. Кроме построения логической топологии, есть еще и физическая топология. Очень удобная вещь, когда нужно проверить, как будет работать что-либо на определенном расстоянии. Не могу утверждать, что работает с точностью до метра, но приблизительные результаты проверить можно. Переключаться между ними можно в левом верхнем углу.



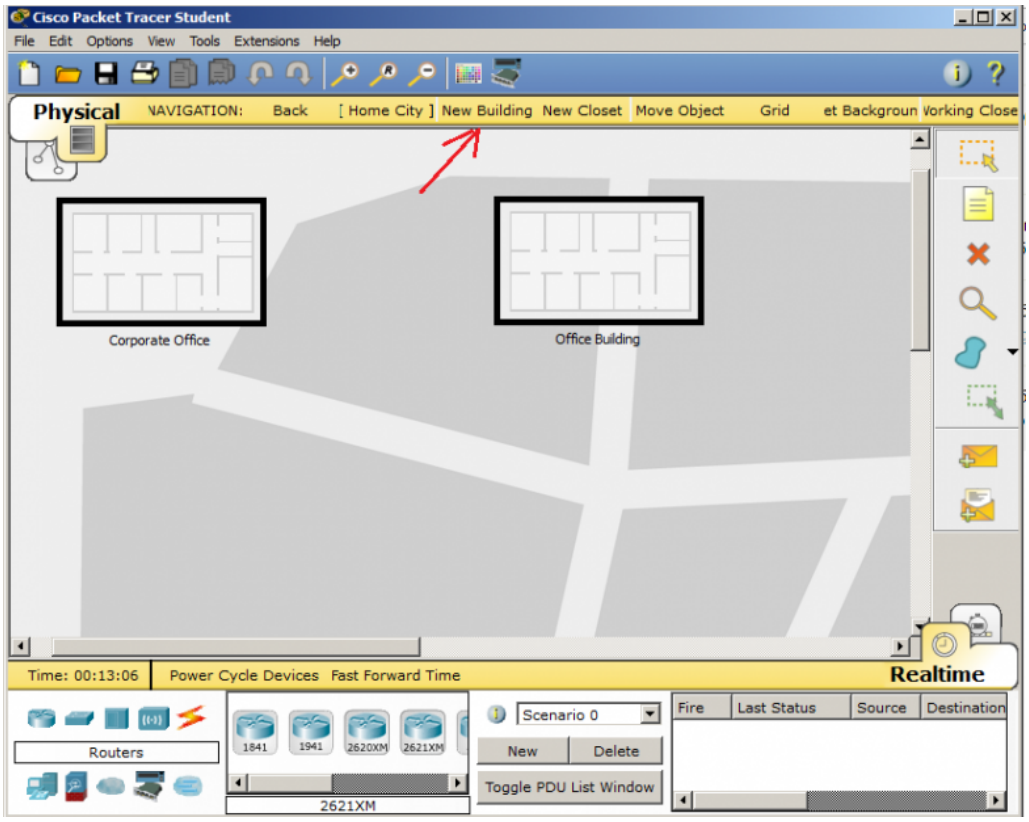
При переключении на физическую откроется следующая картинка.



Это условная географическая карта с созданным городом. Вы можете сами понастроить таких же городов и развернуть междугороднюю связь. Но, так как повторитель усиливает всего на 100 метров, то надо искать что-то более близкое к данному расстоянию. Кликаем по Home City и попадаем в город.

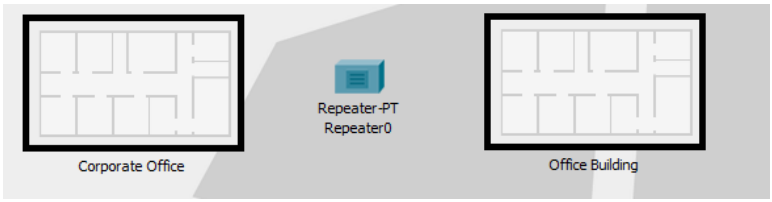


Здесь уже стоит какой-то корпоративный офис. Создадим еще один офис и между ними организуем связь при помощи повторителя. Данное расстояние уже будет более похожим на правду.

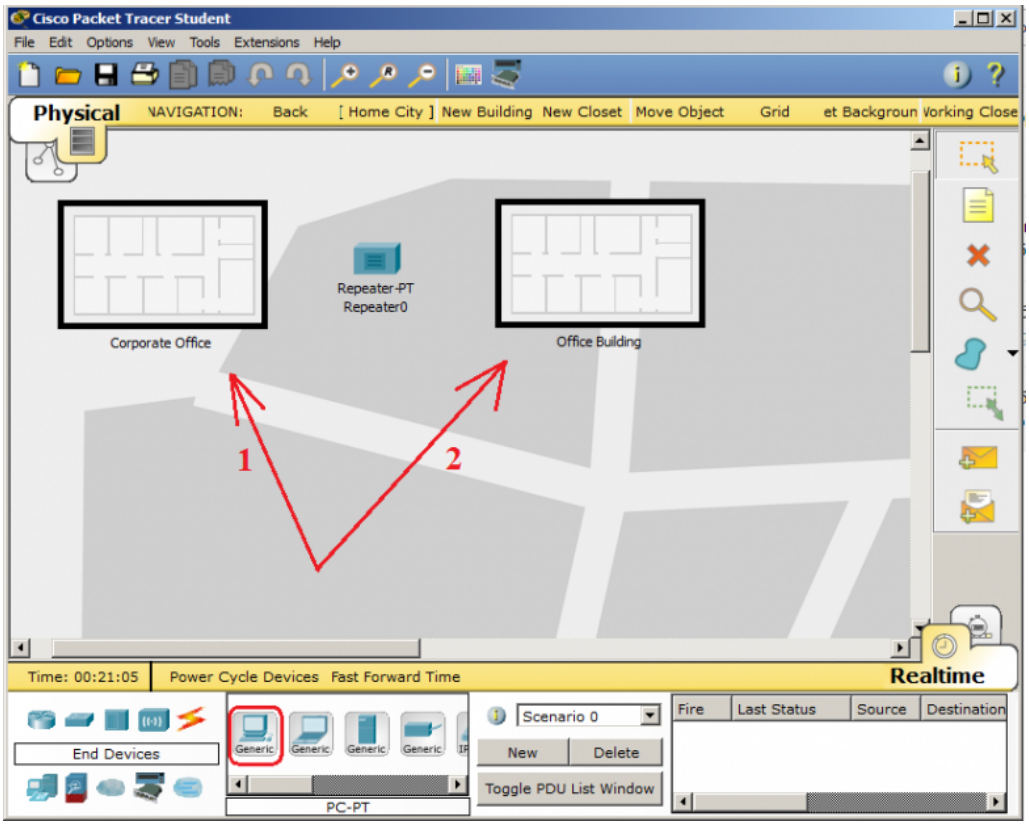


Кликаем по New Building и создается еще одно здание. Расположу его поудобнее.

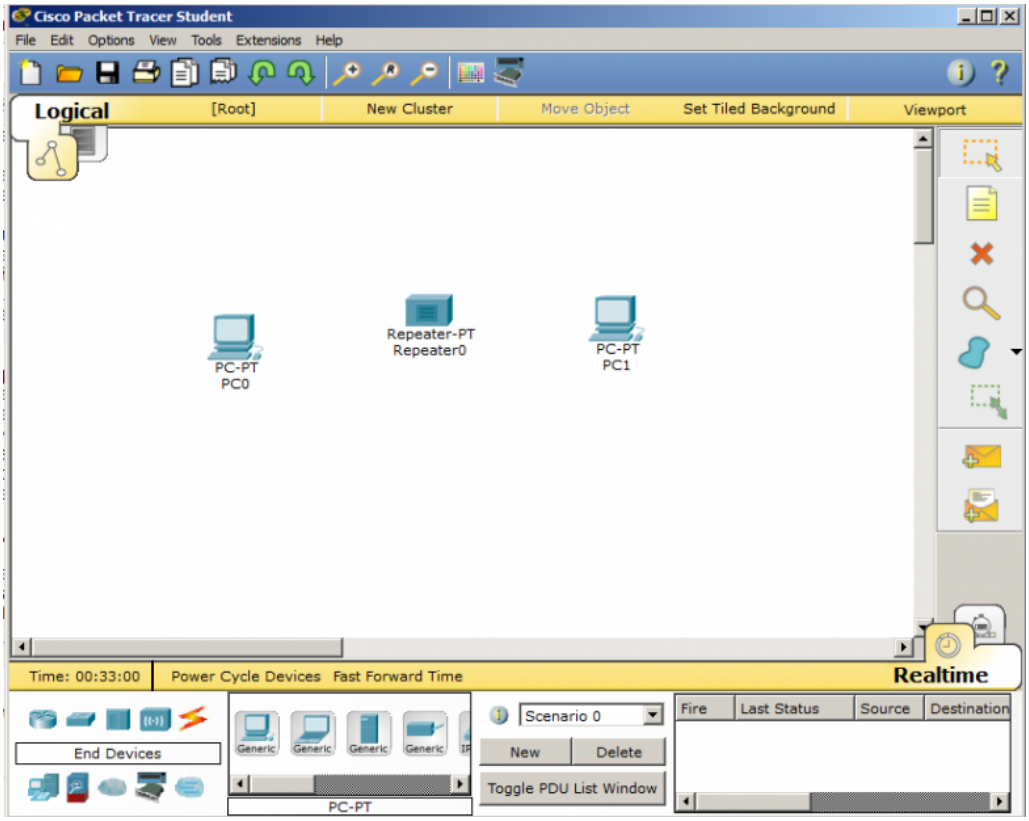
Теперь самое время расставлять узлы. Первым делом установлю между ними повторитель. Захожу на вкладку Hubs. Выбираю Repeater и ставлю его, как на картинке ниже.



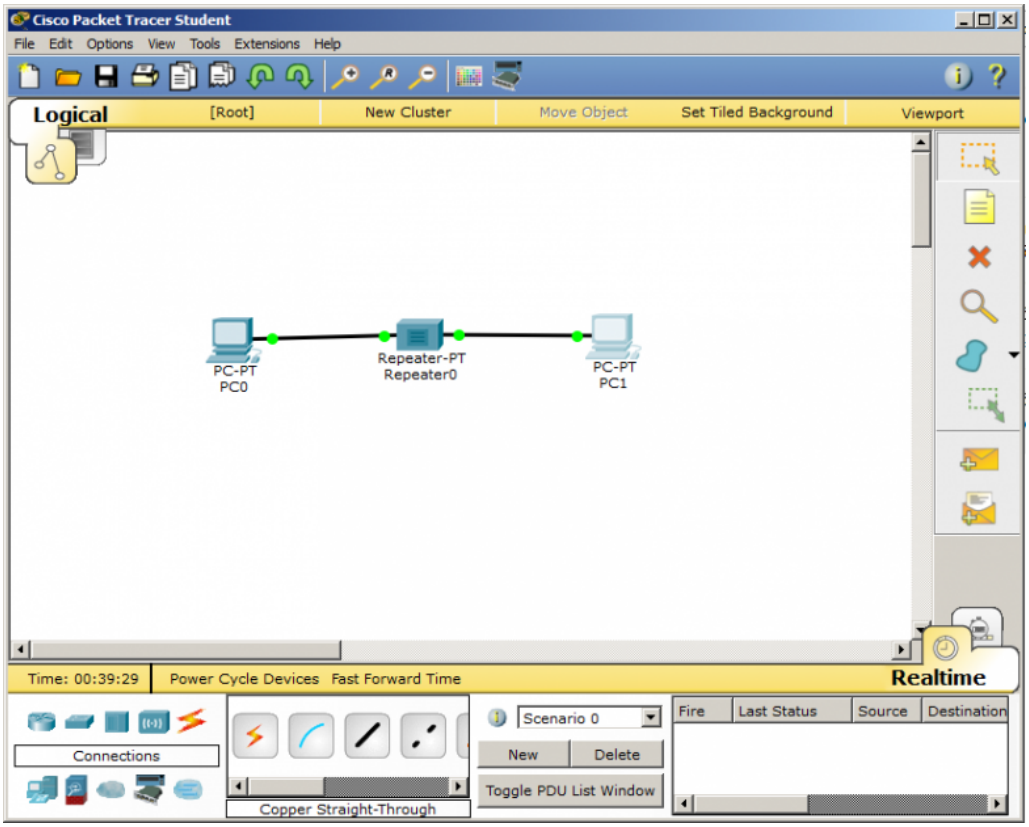
Теперь надо расставить компьютеры. Конечно это бредово, что в каждом офисе по одному компьютеру, которые соединены еще через повторитель. Но для простоты пусть будет так. Перехожу на вкладку End Devices и выберу PC. И кину в каждый офис по компу, как на картинке ниже.



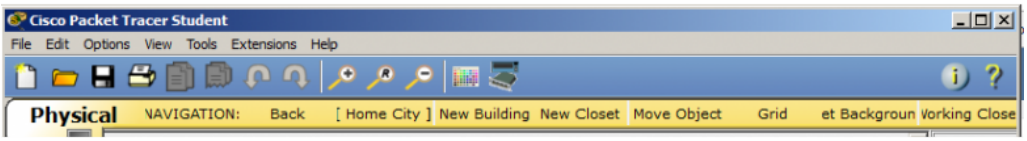
Самое время всё соединить. Переключаюсь на логическую топологию и наблюдаю следующую картину:



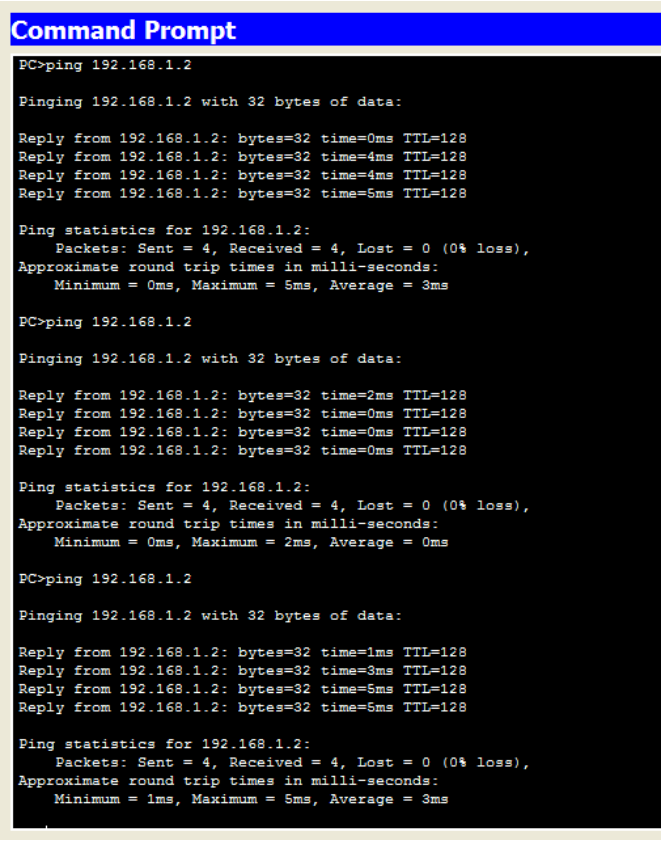
Здесь я вижу все устройства, которые присутствуют в проекте. Хотя в физической топологии видно только повторитель, а компьютеры скрыты в здании. Соединим их. Только соединять будем прямым кабелем, так как это разные устройства. Адресация будет такая же, как и в предыдущей лабораторке. Левый будет с IP-адресом: 192.168.1.1 и маской :255.255.255.0, а правый с IP-адресом:192.168.1.2 и аналогичной маской: 255.255.255.0.



После переключаемся на физическую топологию и наблюдаем следующее.



Все соединения, которые были произведены в логической топологии, автоматически отобразились и в физической. 2 офиса соединены. Самое время проверить доступность командой ping.



Как видим, все прекрасно работает. Но обратите внимание на одну вещь. Я намеренно пропинговал несколько раз, чтобы показать, что каждый раз мы получаем разные результаты (то 4мс, то 5мс). Если до этого время практически стабильно было 0 мс, то есть без задержек, то с повторителем оно уже присутствует.

Вот так работает повторитель. Привожу ссылку на скачивание.

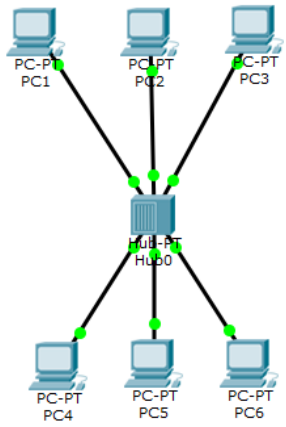
Далее в очереди стоит **концентратор** или **hub**. Устройство, которое охватило популярность, начиная с 90-х годов и до начало 2000-х. Причем слово «хаб» настолько сильно засело всем в голову, что до сих пор многие люди называют любое сетевое устройство этим именем. Многие еще называют его повторителем. Конечно это не совсем верно, так как повторитель — это устройство, показанное выше. Но и сказать, что это ложь, тоже нельзя. Так как это и есть многопортовый повторитель. Но корректнее все же называть его концентратором, либо хабом, чтобы четко отличать данное устройство от повторителя, показанного выше.

Далее вашему вниманию представлю парочку известных концентраторов.

Концентратор от компании Netgear.

Концентратор от компании Cisco.

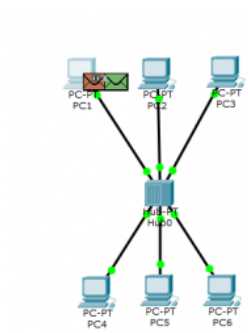
Логика работы его проста. Сигнал, полученный с порта, передается на все остальные порты, кроме исходного. Я перехожу к CPT и создаю лабораторку, как на картинке ниже.

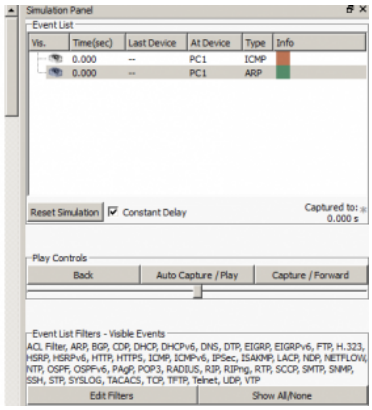


6 компьютеров подсоединены к одному концентратору. Концентратор настраивать не надо. Он работает сразу, как только вытащишь из коробки. А вот компьютеры я настроил и привожу настройки:

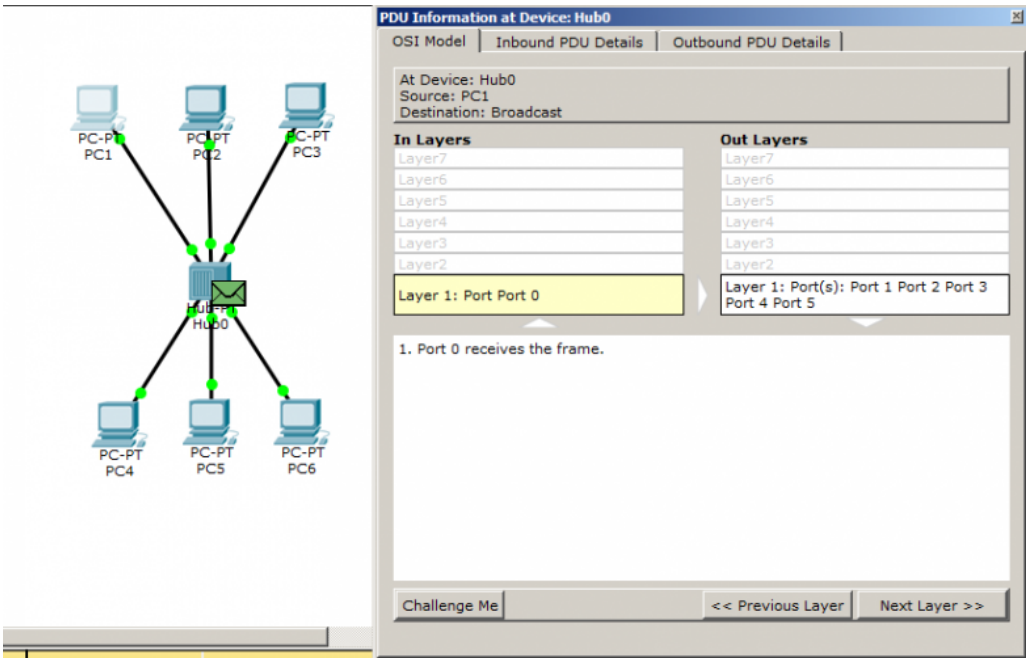
> 1) PC1: IP-192.168.1.1, Mask-255.255.255.0.
> 2) PC2: IP-192.168.1.2, Mask-255.255.255.0.
> 3) PC3: IP-192.168.1.3, Mask-255.255.255.0.
> 4) PC4: IP-192.168.1.4, Mask-255.255.255.0.
> 5) PC5: IP-192.168.1.5, Mask-255.255.255.0.
> 6) PC6: IP-192.168.1.6, Mask-255.255.255.0.

Перевожу CPT в режим симуляции и проверю доступность до PC6, используя компьютер PC1.

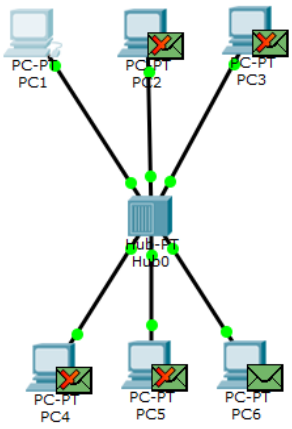




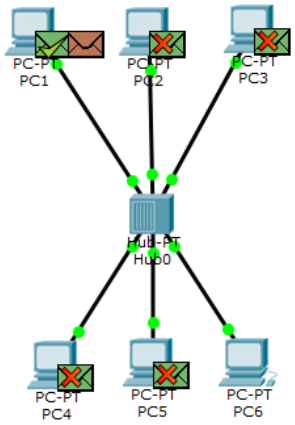
Формируется 2 сообщения. Одно из них — это ICMP, а второе — ARP. ICMP пока отрабатывать не будет, так как не знает MAC-адрес PC6. А вот ARP начнет сразу отрабатывать, чтобы достать MAC-адрес (об этом рассказывается в предыдущей статье подробно). Итак PC1 отправляет ARP на концентратор.



Сообщение пришло, и предлагаю внимательно посмотреть на его содержимое. Несмотря на то, что сообщение несет в себе какую-то информацию, для концентратора это просто поток битов. Он знает, что сообщение пришло с 0-ого порта и передать его надо на 1, 2, 3, 4, 5 порты.

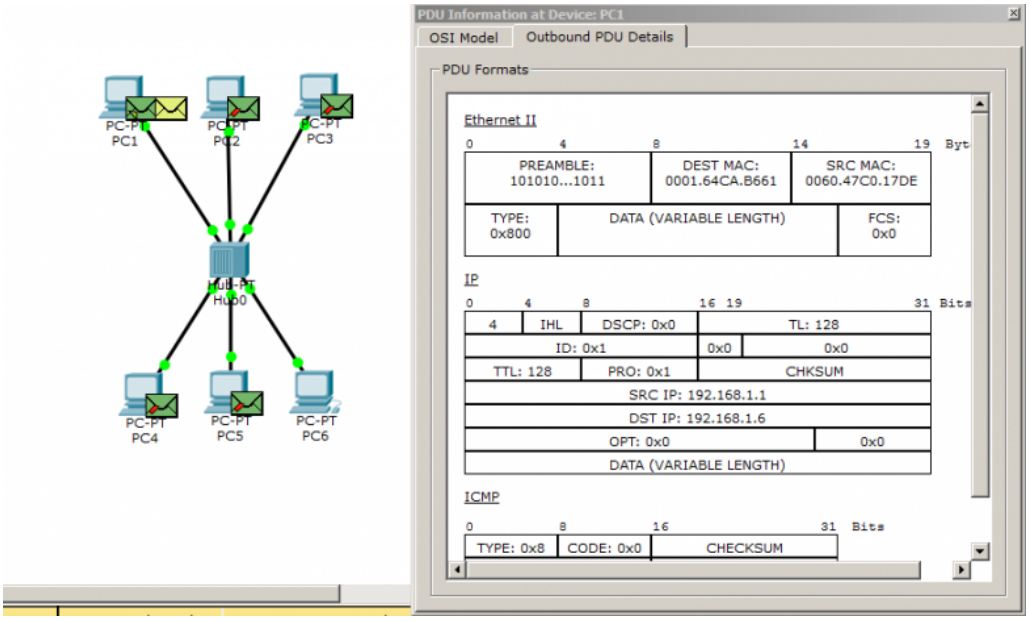


И действительно. Сообщение разослано на все компы, кроме исходящего. Соответственно, PC6 понимает, что это сообщение для него и сформирует ответ, а остальные компы проигнорирует. Вы можете возразить, что протокол ARP при поиске MAC-адреса всегда так работает, и будете правы. Но давайте посмотрим, что будет происходить дальше.

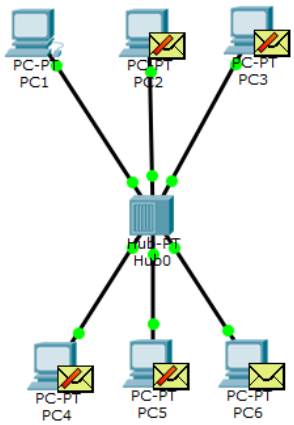


И что мы видим?! Сообщение так же рассылается на все компы, кроме исходящего. Хотя обратное ARP-сообщение содержит точного адресата.

Теперь когда PC1 знает MAC-адрес PC6, он сформирует ICMP сообщение, которое концентратор обработает точно так же, как и ARP. Перезапустил я CPT, и ICMP у меня теперь желтого цвета.



Перед дальнейшим просмотром открою сообщение и посмотрю, что внутри. Четко видно, что у него есть Source MAC, Destination MAC, Source IP и Destination IP. Соответственно, у сообщения задан конкретный получатель.



Но несмотря на вышесказанное, оно будет так же разослано на все порты, кроме исходящего. В этом суть работы концентратора. Для тех, кто хочет лично увидеть его работу, привожу ссылку на скачивание.

Если раньше такое поведение не вызывало сильных опасений (когда число компьютеров было до 10), то со временем увеличилось число компьютеров и устройств, которые подключались к сети. Это привело к тому, что сеть очень сильно нагружалась, и работать стало тяжело. Причем вся сеть в то время работала в режиме полудуплекса (half-duplex). Это значит, что по одним и тем же проводам велась передача или прием. Соответственно, чем больше компьютеров начинает вещать в сети, тем больше вероятность появления коллизии. Нужно было срочно находить решение, чтобы каким либо образом ограничивать сегменты сети. И для ее разрешения стали применять **мосты** или **bridge**.

Мост от компании Netgear

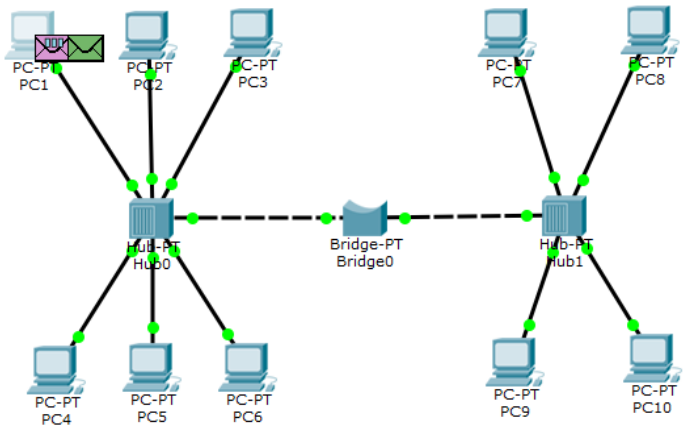
Теперь расскажу, что они из себя представляли. Это уже было более умное устройство, которое работало на 2-ом уровне модели OSI. То есть оно знало, что такое MAC-адреса и как с ними работать. Теперь каждый его порт был закреплен под конкретный сегмент сети, то есть он решал одну из важнейших проблем. Вдобавок у него была система фильтрации. То есть он не пересылал широковещательные кадры, которые не предназначены другому сегменту сети. У

него появилась своя таблица, куда он записывал, кто за каким портом сидит. То есть, кадр, пришедший на мост, не слепо отправлялся на другой порт, а сверялся с таблицей, и если за другим портом сидит адресат, кадр выпускался. В противном случае мост его уничтожал.

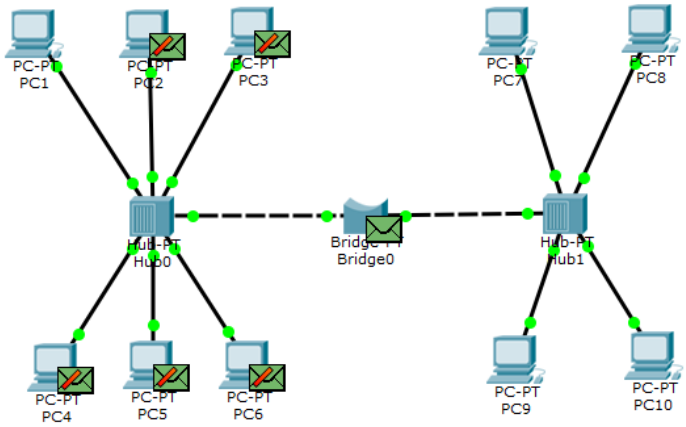
Почитали теорию и время перехода к практике. Так как нам в этой лабе понадобятся концентраторы и не один компьютер, я взял за основу предыдущую лабу и модернизировал ее. Единственное, что расстроило — это то, что мост в CPT реализован условно. Он выполняет все нужные функции, но зайти и посмотреть на его таблицу нельзя (хотя она у него присутствует). Но это не важно. Главная цель — это показать работу данного устройства. Итак в этой лабе добавился мост и концентратор с 4 компьютерами. Если у вас не хватает портов на концентраторе, чтобы соединить с мостом, то можете добавить ему дополнительный интерфейс. Только не забудьте перед этим переключить на нем выключатель. 6 левых компьютеров я не трогал, поэтому адресация у них не поменялась, а вот для 4 правых компьютеров приведу ниже:

> 1) PC7: IP-192.168.1.7, Mask-255.255.255.0.
> 2) PC8: IP-192.168.1.8, Mask-255.255.255.0.
> 3) PC9: IP-192.168.1.9, Mask-255.255.255.0.
> 4) PC10: IP-192.168.1.10, Mask-255.255.255.0.

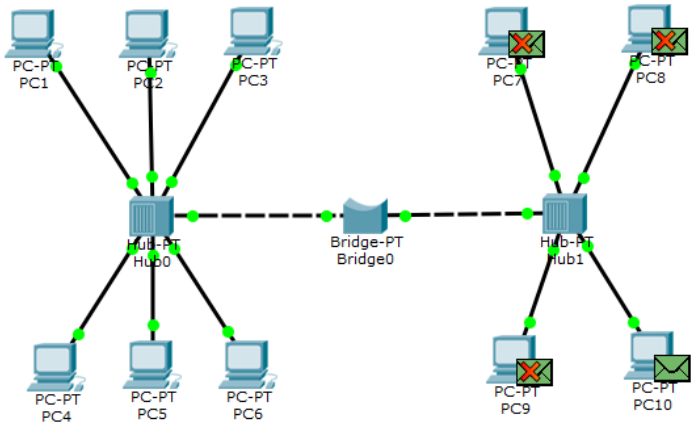
Перехожу в режим симуляции и попробую пингануть PC10 с компьютера PC1.



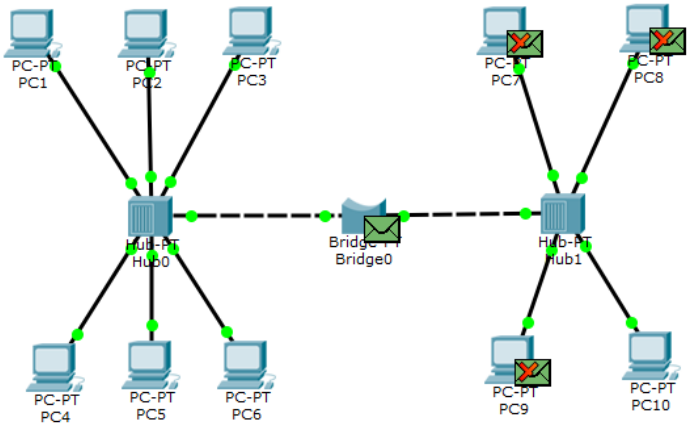
По стандартной схеме создаются 2 сообщения, но первым в бой идет ARP.



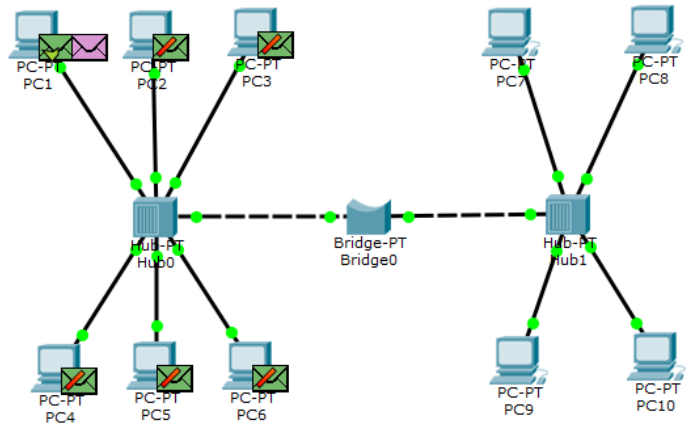
Концентратор отправляет его всем, кроме порта отправителя. И все его отбрасывают, кроме моста. Хоть он и не адресован мосту, он так же не знает, есть ли там такой получатель. Поэтому он его отправляет, чтобы проверить.



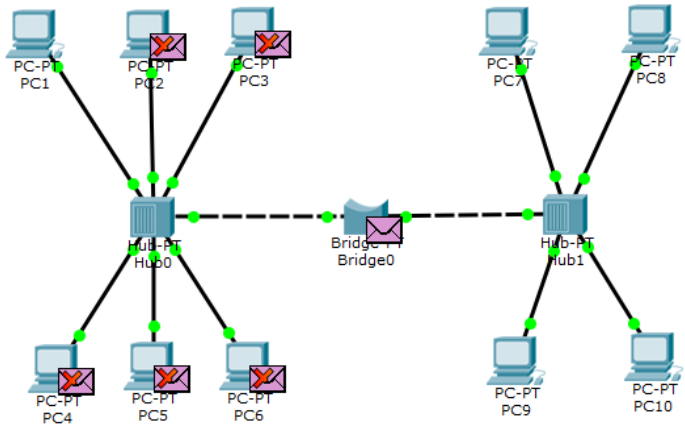
Концентратор на правой стороне отрабатывает как положено, и в данном сегменте находится получатель. Он отправляет ответное сообщение.



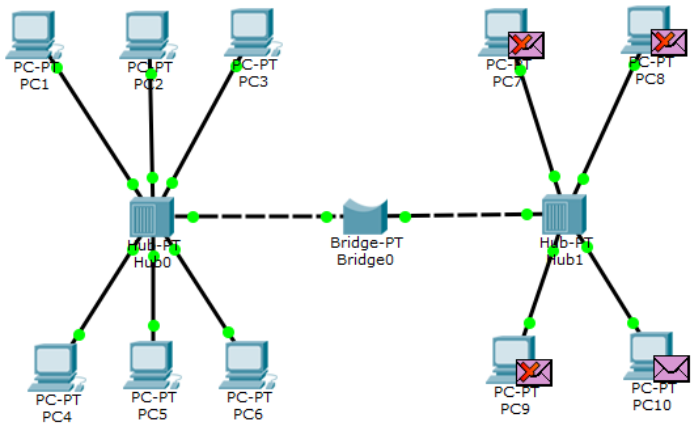
Концентратор отрабатывает, и все узлы, кроме моста, отбрасывают его.



Мост выкидывает это сообщение на левый концентратор. А тот, в свою очередь, выкидывает его всем участникам. PC1 узнает себя в этом сообщении и посылает теперь ICMP.

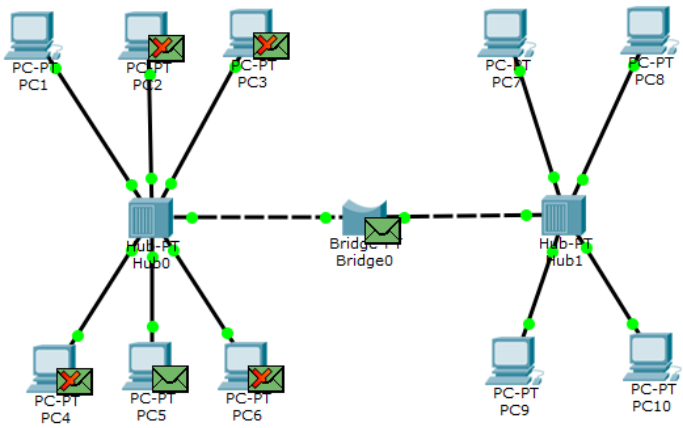


Концентратор отрабатывает. Сообщение попадает на мост. Он смотрит, есть ли у него такой получатель. Видит, что присутствует, и отправляет.



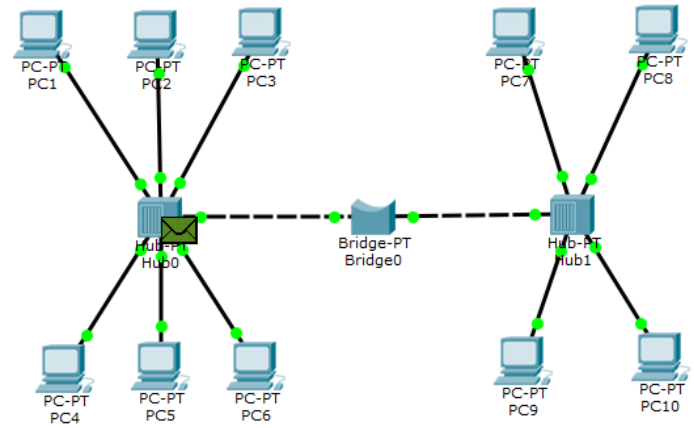
Концентратор рассылает его всем и PC10 получает его. Ответный пинг пройдет по такому же маршруту.

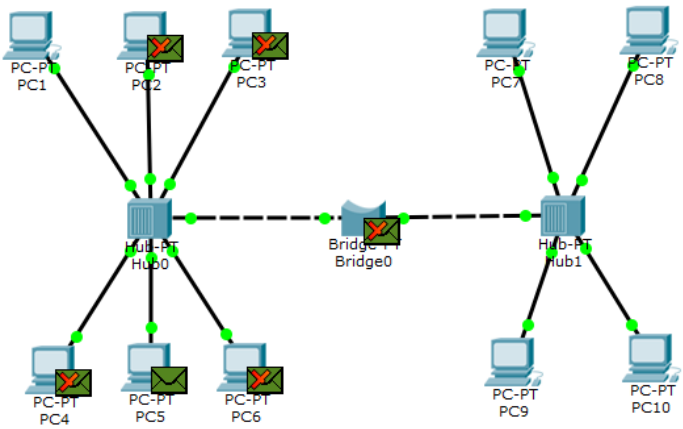
Давайте посмотрим ситуацию, когда обменивающиеся узлы находятся в одном сегменте, и не требуется отправлять сообщение через мост. Проверю доступность PC5 с PC1.



Отправляет на концентратор ARP, а тот, в свою очередь, на всех. И заметьте, что призадумались 2 устройства(мост и PC5). PC5 понимает, что это для него, и отправляет ответ. А мост решает проверить, есть ли справа такой получатель. Ответа он не дожидается и понимает, что такого там нету.

Теперь, когда PC1 знает про PC5 он формирует ICMP для него.





PC5, получив его, готовит ответ. А вот мост теперь знает, что справа нету такого получателя и сразу отбрасывает такой кадр. Тем самым здесь и показано то, каким образом он фильтрует.

Вот так и работали и работают мосты (если они еще где-то применяются). Как видите, мост создал 2 сегмента или 2 домена коллизий. То есть все, что происходит за левым портом моста, никак не влияет на правый, если только сообщение не предназначено для узла в другом сегменте. Тем самым это обеспечило снижение нагрузки на сеть. Привожу ссылку на скачивание.

Переходим дальше и поговорим о **коммутаторах**. Про них, наверное, слышали все, да и многие из вас работали с ними. Коммутаторы бывают разные, и отличаются они своими функциями и, конечно, ценой. Давайте поговорим о них и выделим главные концепции. С появлением мостов и их фильтрацией, инженеры задались вопросом, чтобы сделать устройство, которое будет разделять не только сегменты сети, но и компьютеры. То есть обеспечить микросегментацию. Когда устройство знает, за каким портом кто сидит, и не будет передавать сообщение всем узлам, предназначенное для определенного узла. В результате появился коммутатор. Так же, как и у моста, у него есть своя таблица. В ней записано, за каким портом сидит определенный MAC-адрес. Называется такая таблица — таблица коммутации. Запись в нее происходит тогда, когда устройство начинает проявлять активность. Например, отправляя какое-либо сообщение, оно в заголовке оставляет свой MAC-адрес. Коммутатор читает этот заголовок и понимает, какой у отправляющего устройства MAC-адрес, и записывает его. Теперь, если придет сообщение именно для этого устройства, он отправит его именно ему. Другим устройствам он отправлять сообщение не будет.

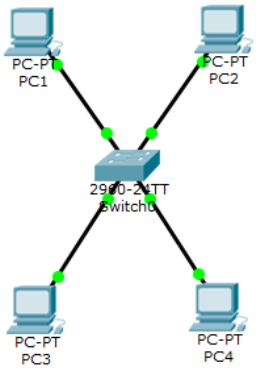
Однако не стоит забывать, что, если вы только что достали коммутатор из коробки и подсоедините к нему устройства, то он не сразу будет знать, кто за каким портом сидит. Изначально таблица у него пустая. И, как я уже писал выше, заполнять он ее будет по мере активности узла. Такой процесс называется **режимом обучения**. Но, как только он ее заполнит, все станет замечательно. При поступлении на коммутатор, какого-либо кадра, он посмотрит на заголовок и прочитает MAC-адрес назначения. Далее он посмотрит на свою таблицу и поищет порт, за которым сидит узел с данным MAC-адресом и, соответственно, отправит.

Процессы коммутации у коммутатора и моста схожи. Но есть важное отличие: коммутация у мостов программная, а у коммутаторов-аппаратная. Если у мостов коммутацию выполнял процессор, то для коммутаторов придумали специальные микросхемы ASIC. Это специализированные микросхемы, которые созданы для выполнения конкретной задачи. Следовательно, такой вид коммутации оказался гораздо быстрее, что и сделало коммутаторы настолько популярными.

С каждым годом коммутаторы становятся все быстрее и умнее. Если мы говорили о коммутаторах, как об устройствах 2-го уровня по модели OSI, то практически все современные коммутаторы от компании Cisco, умеют работать на уровнях выше.

Такие коммутаторы стали называть L2+ коммутаторы. Почему L2+, а не L3, я сейчас объясню на практике.

Открываю CPT и соберу лабораторку, как на картинке ниже.



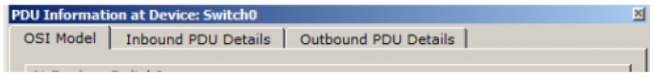
Присутствует коммутатор и 4 компьютера. Я пока не изменял традицию назначения IP-адресов, но все же предоставлю вам список:

> 1) PC1: IP-192.168.1.1, Mask-255.255.255.0.
> 2) PC2: IP-192.168.1.2, Mask-255.255.255.0.
> 3) PC3: IP-192.168.1.3, Mask-255.255.255.0.
> 4) PC4: IP-192.168.1.4, Mask-255.255.255.0.

Так как мы только включили коммутатор, то таблица MAC-адресов у него должна быть пуста. Проверим. Для проверки используем команду «show mac-address-table»:

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
```

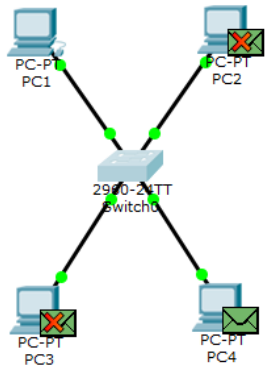
Убеждаемся, что она пустая, и переходим дальше. Самым простым и быстрым методом проверки будет команда ping. Проверим ею доступность PC4, используя PC1. Естественно, сначала должен будет отработать протокол ARP.



Коммутатор умный и может читать, что запаковано на втором уровне. Он видит MAC-адрес отправителя, который он запишет себе в таблицу. Еще он видит широковещательный MAC-адрес (то есть для всех). Значит надо передать этот кадр всем, кроме отправителя. Обратите внимание на 1-ый уровень. То есть на входе (In Layers), он получил кадр с 1 порта, а на выход (Out Layers) отправит по 2, 3 и 4 порту. В целом сейчас он работает, как концентратор. Не буду я пока передавать с коммутатора кадр. Перед этим нужно проверить таблицу MAC-адресов.

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0060.4767.0327    DYNAMIC     Fa0/1
```

И вижу первую запись. Он записал MAC-адрес и порт, с которого он был получен. Прекрасно! Смотрим, что будет дальше происходить.

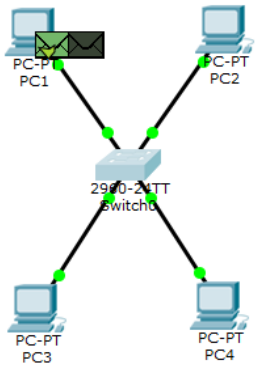


Отправляет он ARP всем, кроме отправителя. И мы видим, что PC4 понял, что это для него, и формирует ответ. Все остальные этот кадр отбрасывают.

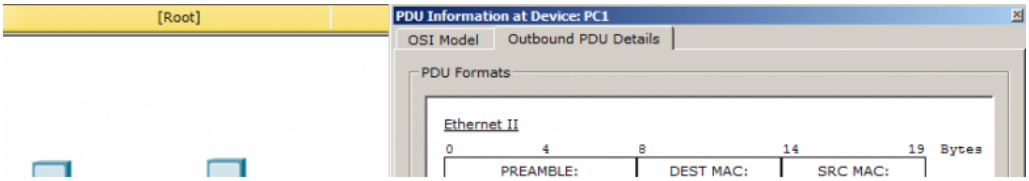
Коммутатор получает ответ и читает его. На втором уровне он видит MAC-адрес отправителя и MAC-адрес получателя. MAC-адрес отправителя он видит впервые, поэтому сразу занесет его в свою таблицу. А вот MAC-адрес получателя он уже знает, поэтому отправит он его только на 1-ый порт. Обратите внимание на данные 1-ого уровня. Получил он его с 4-ого порта, а отправит на 1-ый. Но перед отправкой проверим таблицу.

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0060.4767.0327    DYNAMIC     Fa0/1
   1    0090.2bc6.21e5    DYNAMIC     Fa0/4
```

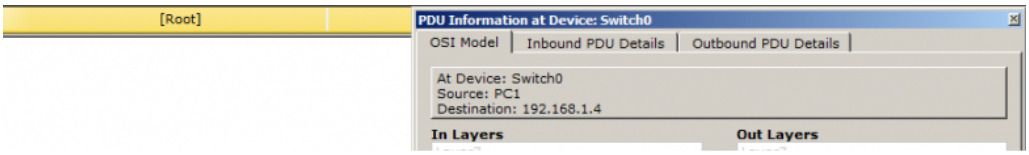
И действительно. MAC-адрес был занесен. Нажимаю я на Capture/Forward.



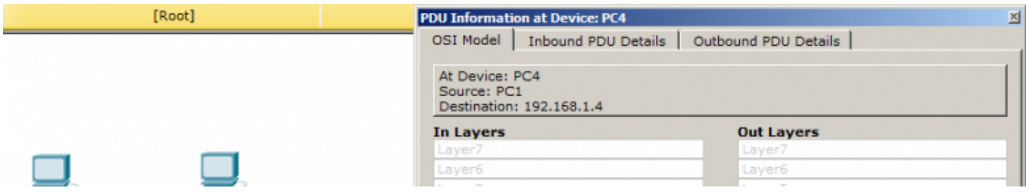
Видим, что сообщение было отправлено только на 1-ый порт (то есть для PC1). Так концентратор точно не делал. Дальше уже формируется ICMP сообщение.



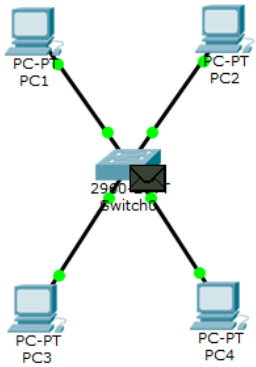
Оно работает на 3 уровне. Отправляем.

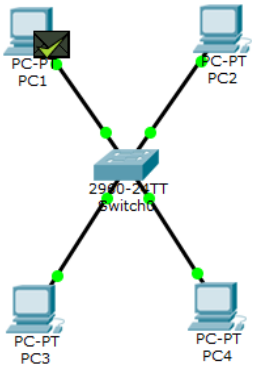


Доходит оно до коммутатора. Открываем и видим, что несмотря на то, что в сообщении есть заголовок 3-ого уровня, коммутатору это по барабану. Он читает только заголовок 2-ого уровня и принимает решение. MAC-адрес PC4 он знает и знает на какой порт отправлять.
Смотрим, как он отработает.



И отрабатывает он правильно. Сообщение отправляется только на 4-ый порт. PC4 формирует ответ.





И ICMP-сообщение без проблем доходит до PC1. Вот весь принцип работы коммутатора. Теперь объясню, почему этот коммутатор называют L2+ коммутатор. Лабораторная работа остается той же, за исключением пары изменений на самом коммутаторе. Выше мы говорили о том, что коммутаторы работают на 2-ом уровне модели OSI. Но с течением времени инженеры придумали управляемые коммутаторы. То есть это уже не просто железка, которая работает сама по себе, и что-то поменять в ходе ее работы не представляется возможным, а более умное устройство, которому есть возможность задать какие-то параметры (например IP-адрес) и настроить на удаленное управление. Продемонстрирую на примере. Открываю предыдущую лабу и меня здесь интересует коммутатор. Захожу на него и присваиваю свободный IP-адрес.

Switch>enable — *переход в привилегированный режим. Отсюда доступно большинство команд.*

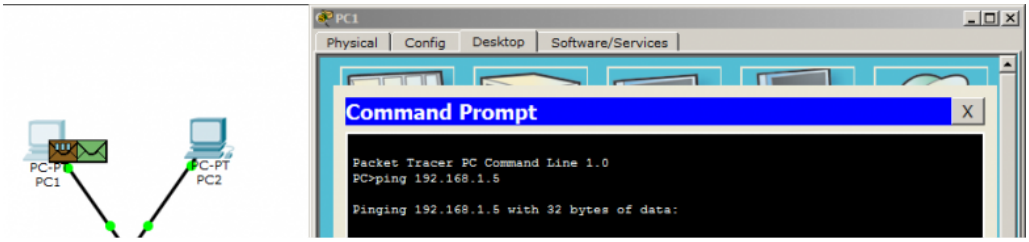
Switch#configure terminal — *переход в режим глобальной конфигурации. В этом режиме возможен ввод команд, позволяющих конфигурировать общие характеристики системы. Из режима глобальной конфигурации можно перейти во множество режимов конфигурации, специфических для конкретного протокола или функции.*

Switch(config)#interface vlan 1 — *так как это коммутатор 2 уровня, то назначить IP-адрес на порт нельзя. Но его можно назначить на виртуальный интерфейс. Поэтому выбираю его и перехожу дальше.*

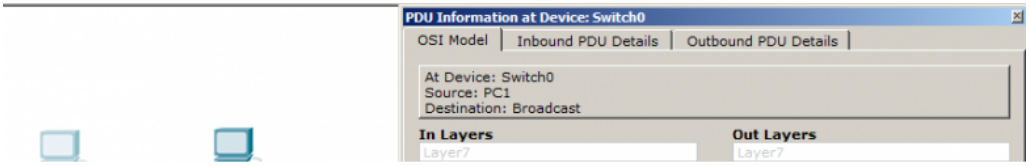
Switch(config-if)#ip address 192.168.1.5 255.255.255.0 — *присваиваю ему один из свободных IP-адресов: 192.168.1.5 и маской: 255.255.255.0.*

Switch(config-if)#no shutdown — *включаю интерфейс. По умолчанию он выключен.*

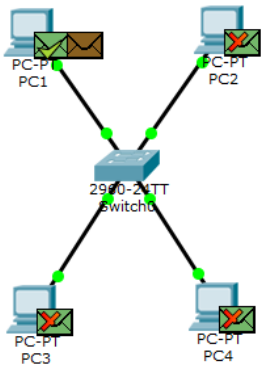
Коммутатор настроен и предлагаю проверить его доступность командой ping. Делать я это буду с PC1.



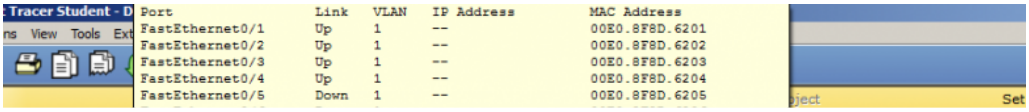
Думаю, что уже не для кого это секретом не будет, что изначально создается 2 сообщения. Итак первым идет ARP.



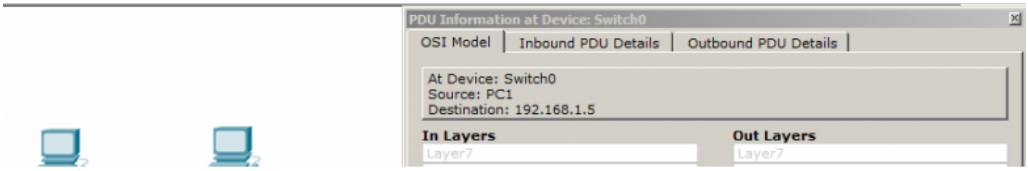
И тут происходит глюк CPT. Он получает ARP. Вскрывает его, видит, что IP-адрес назначения его. Но все равно хочет отправить его всем. Только PC1 он отправит ответный ARP, а всем остальным разошлет ARP от PC1. Будем наблюдать за дальнейшими событиями.



ARPы дошли до узлов. PC1 теперь знает MAC-адрес виртуального интерфейса коммутатора. О чем свидетельствует картинка ниже.



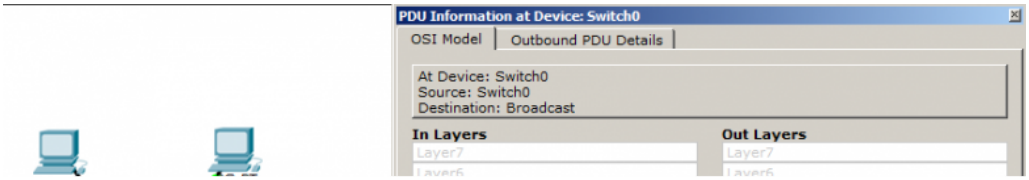
Пришло время ICMP сообщения. Формирует его и запускает.

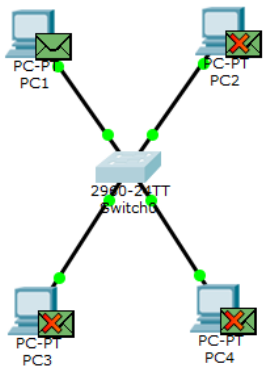


ICMP сообщение доходит до коммутатора. Смотрим, что же внутри. И видим, что коммутатор действительно смог прочитать заголовок 3-го уровня. Он узнает себя, но происходит еще один глюк. Посмотрите на колонку «Out Layers». Он не знает, какой MAC-адрес у PC1, что конечно является бредом. И я это сейчас покажу. Когда пришло ICMP сообщение (колонка «In Layers»), в заголовках 2 и 3 уровня, были записаны MAC-адрес отправителя и IP-адрес получателя. То есть он знал, какой ему нужен MAC-адрес для того, чтобы отправить ответ. Не продвигая пакет дальше, посмотрим на таблицу коммутации.

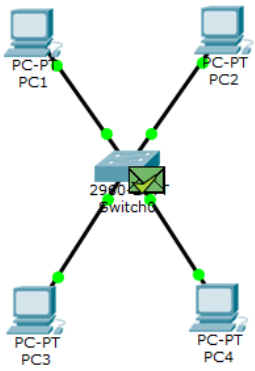
```
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0060.4767.0327    DYNAMIC     Fa0/1
```

Видим, что данный MAC-адрес действительно присутствует. Ну и раз он «не знает» MAC-адрес PC1, то вынужден запустить протокол ARP. Давайте посмотрим, что из этого выйдет.

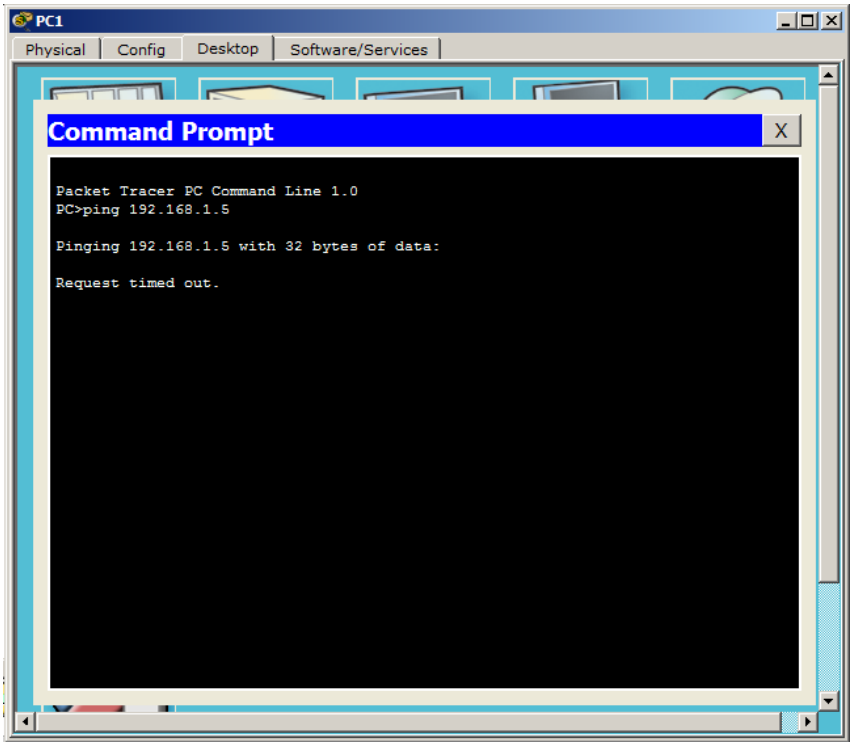




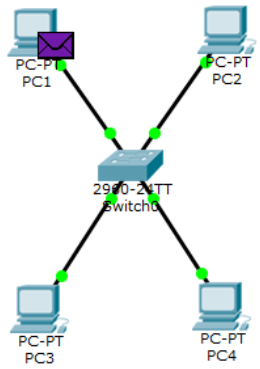
PC1 получает сообщение. Он в шоке и в недоумении, потому что уже сообщал ему свой MAC-адрес. Но раз попросил, то отправит еще раз.

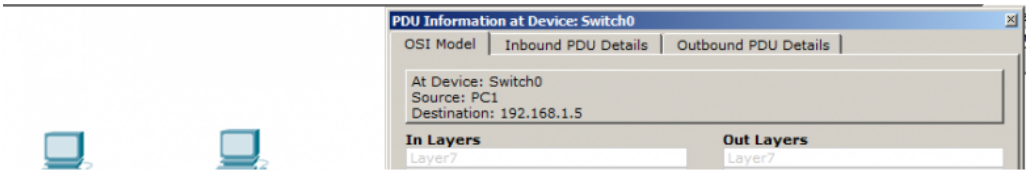


Коммутатор получает ответ и обновляет свою таблицу. При этом он забывает о том, что нужно ответить на ping. Потому что до этого он его отбросил. Что же будет происходить дальше?! Тот первый ICMP запрос затерялся и больше о нем никто не вспоминает. Истекает таймер запроса на PC1, о чем свидетельствует картинка ниже.

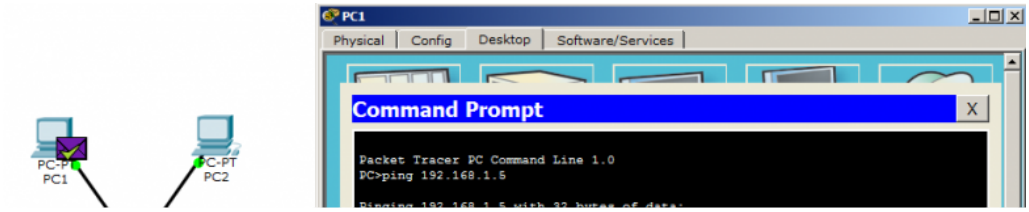


PC1 решает отправить второе ICMP сообщение.





Коммутатор получает его и начинает читать заголовки. На этот раз все работает хорошо. Он узнает себя в нем и знает, кому ответить.



Пакет успешно доходит до компьютера. Убедиться в этом можно, обратив внимание на скриншот из консоли. Дальше он сформирует еще 2 таких ICMP сообщения (суммарно 4). Показывать их я не буду, так как они дублируют предыдущие действия. Несмотря на то, что у CPT случился небольшой глюк, он заставил подробно себя изучить, что иногда весьма полезно. Из-за того, что коммутатор смог прочитать заголовок 3 уровня и ответить на ICMP сообщение (он также мог сам проверить доступность любого узла), его стали называть L2+ коммутатор. Чисто L2

коммутаторы с IP-адресами работать не умеют. Но вот вопрос, почему же данный коммутатор не L3? А все кроется в том, что он не умеет выполнять маршрутизацию (передача пакетов из одной канальной среды в другую). Есть конечно, L3 коммутаторы, но о них мы поговорим, когда разберем маршрутизатор. Прикладываю ссылку на данную лабораторную работу.

Итак, встречаем **маршрутизатор** или **router**. В принципе вы уже видели, как он работает в предыдущих статьях. Но освежу еще раз кратко.

Маршрутизатор — это устройство, предназначенное для пересылки пакетов из одной канальной среды в другую. Также главной его функцией является выбор наилучшего маршрута для пакета. Многие называют данное устройство шлюзом. Так как, если надо передать какой-то пакет из одной канальной среды в другую, промежуточным устройством будет именно шлюз.

Само устройство очень старое. Если верить истории, то первый роутер был создан в 1976 году и объединял 3 локальные сети. Вот, к примеру, один из первых маршрутизаторов компании Cisco (еще когда название начиналось с маленькой буквы).

Сейчас их тоже огромное количество. Причем они подразделяются по сферам применения. Есть домашние, магистральные и так далее. Вот, к примеру, один из современных магистральных маршрутизаторов.

Маршрутизаторы серии Cisco 7600

Или к примеру Cisco 2811, который будет использоваться в следующей лабе.

Предлагаю собирать лабораторку и переходить к практике.

Добавил один роутер, который будет перенаправлять пакеты из одной канальной среды в другую. И 2 коммутатора, к которым подключены по 2 компьютера. Настройки компьютеров следующие.

> 1) PC1: IP-192.168.1.2, Mask-255.255.255.0, Gateway: 192.168.1.1.
> 2) PC2: IP-192.168.1.3, Mask-255.255.255.0, Gateway: 192.168.1.1.
> 3) PC3: IP-192.168.2.2, Mask-255.255.255.0, Gateway: 192.168.2.1.
> 4) PC4: IP-192.168.2.3, Mask-255.255.255.0, Gateway: 192.168.2.1.

Как видите, добавился параметр основного шлюза (Gateway). Для компьютеров в левом сегменте он один, а для компьютеров в правом сегменте другой. Коммутаторы остаются без изменения настроек. А вот маршрутизатор требует настройки. Переходим к нему.

> Router>enable — *переход в привилегированный режим.*
> Router#configure terminal — *переход в режим глобальной конфигурации.*
> Router(config)#interface fastEthernet 0/0 — *переход в режим настройки данного интерфейса.*
> Router(config-if)#ip address 192.168.1.1 255.255.255.0 — *присваиваем ему IP-адрес. Данный интерфейс будет шлюзом для левой сегмента сети.*
> Router(config-if)#interface fastEthernet 0/1 — *переход в режим настройки данного интерфейса.*
> Router(config-if)#ip address 192.168.2.1 255.255.255.0 — *присваиваем ему IP-адрес. Данный интерфейс будет шлюзом для правого сегмента сети.*
> Router#copy running-config startup-config — *сохраняем конфигурацию*

Маршрутизатор настроен, и можно посмотреть таблицу маршрутизации командой show ip route.

Видим 2 connected сети. Прописывать специфичную настройку маршрутизации не понадобится, так как сегменты у нас подключены через один маршрутизатор. Время проверить доступность PC3, используя PC1.

Путем простой математики, PC1 понимает, что получатель находится не в его сети, а значит передать надо через основной шлюз. Но возникает проблема, что он не знает MAC-адрес шлюза. В связи с этим пускает в разведку ARP.

Попадает ARP на коммутатор, и посмотрим на заголовок. И видим, что в Destination IP: 192.168.1.1.

Передает он его дальше, и маршрутизатор понимает, что это для него. И отправляет ответ.

ARP ответ доходит до компьютера и он формирует ICMP сообщение. Обратите внимание, что IP-адрес назначения — это адрес PC3. А MAC-адрес назначения — это адрес маршрутизатора.

Коммутатор прочтет Ethernet заголовок и передаст маршрутизатору.

Маршрутизатор, получив это сообщение, понимает, что он не знает, кто сидит в сети с IP:192.168.2.2. Отбрасывает ICMP сообщение и запускает ARP.

Коммутатор получив ARP, сразу рассылает его. Находится получатель, который формирует ответ.

Я, с вашего позволения, не буду показывать процессы, которые дублируются, по причине их очевидности. Итак ARP дойдет до маршрутизатора, и он теперь знает MAC-адрес PC3.

Тем временем истекает таймер у PC1 и он формирует следующее ICMP сообщение.

Коммутатор, по заголовку, принимает решение отправить это сообщение на маршрутизатор.

Маршрутизатор, просмотрев заголовок, понимает, что надо передать его в другую канальную среду, и меняет поля в заголовке Ethernet.

Доходит до коммутатора, где он понимает, что сообщение надо передать PC3, то есть на 1-ый порт.

PC3 формирует ответ.

И в результате ответ доходит до PC1, о чем свидетельствует окно консоли.

Вот весь принцип работы маршрутизатора. Если вы читали предыдущие статьи, то нового в основах работы с маршрутизатором мало узнали. Еще одна из фишек маршрутизатора — это выбор лучшего маршрута, но это мы разберем в следующей статье. Ну и по традиции привожу ссылку на скачивание.

Поговорили про маршрутизатор, и я предлагаю разобрать **L3 коммутатор**. Его еще называют MLS(Multi Layer Switch) коммутатор. Отличие его от обычного коммутатора в том, что он осуществляет маршрутизацию. Данный вид коммутаторов стал настолько популярным, что многие крупные вендоры стали вкладывать деньги в его развитие. Сейчас на рынке можно встретить L3 коммутаторы от таких производителей как HP, TP-Link, Cisco и так далее. Ниже приведу несколько моделей.

L3 коммутатор от компании TP-Link

L3 коммутатор от компании HP

L3 коммутатор от компании Cisco

Предлагаю перейти к практике. Я возьму за основу предыдущую лабораторную работу. Но вместо маршрутизатора поставлю L3 коммутатор.

Компьютеры настроены. Осталось настроить L3 коммутатор. Настраивается он немного иначе, нежели маршрутизатор. Переходим к его настройке.

> Switch>enable — *переход в привилегированный режим.*
> Switch#configure terminal — *переход в режим глобальной конфигурации.*
> Switch(config)#interface fastEthernet 0/1 — *переход к настройке интерфейса fa0/1.*
> Switch(config-if)#no switchport — *переводим порт в «роутерный» режим. Без этой команды вы не сможете повесить на него IP-адрес.*
> Switch(config-if)#ip address 192.168.1.1 255.255.255.0 — *присваиваем IP-адрес.*
> Switch(config-if)#interface fastEthernet 0/2 — *переход к настройке интерфейса fa0/2.*
> Switch(config-if)#no switchport
> Switch(config-if)#ip address 192.168.2.1 255.255.255.0 — *присваиваем IP-адрес.*
> Switch(config)#ip routing — *включаем маршрутизацию на интерфейсе.*

Настройка закончена. Настало время перейти к команде ping. Я не стал показывать работу команды ARP. Думаю каждый из вас знает, как она работает, а начал фиксировать моменты, когда начал работать ICMP. Привожу подробные картинки.

Я думаю процесс понятен. Он ничем не отличается от того, что происходило, когда там стоял маршрутизатор. Теперь отвечу на вопрос: Отличие L3 коммутатора от маршрутизатора, и что лучше. Я, в свое время, очень долго искал ответ на этот вопрос. И нашел его здесь. Если кратко, то самая большая разница в них заключается в цене. За счет того, что в L3 коммутаторе применяются интегральные схемы специального назначения, то он быстрее и в связи с этим дороже. Дублировать его статью я не буду, поэтому читайте. Там, действительно, очень хорошо об этом написано! От себя только добавлю ссылку на готовую лабораторку.

Забыл я упомянуть еще одно устройство. И это **dial-up модем**. То самое устройство, при помощи которого, многие стали выходить в Интернет. Единственное, что ему было нужно, это телефонная сеть. Компьютер, подключенный к модему, устанавливал связь с провайдером, который выделял ему канал и давал доступ. Такой процесс назывался дозвон. В связи с тем, что с того времени технологии шагнули далеко вперед, то такое соединение уже мало где встретишь. Хотя они еще встречаются в местах с низким населением или в отдельных странах. Давайте посмотрим, как выглядели эти устройства.

Модем от компании Zyxel

Модем от компании U.S. Robotics

Позже появились и сетевые адаптеры со встроенным модемом. То есть телефонная линия соединялась напрямую с компьютером. Ниже привожу один из таких образцов.

Долго я возился с вопросом, чтобы собрать простую лабораторку и показать, как это раньше работало. Вышло что то непонятное, но интересное.

Итак, что есть что. У нас есть 2 компьютера с модемными интерфейсами. И подключенные к облаку(это своеобразная эмуляция глобальной сети. Устройство с множеством интерфейсов) при помощи телефонного кабеля. И слева располагается маршрутизатор, соединенный 2-мя телефонными кабелями с облаком. Покажу, как менять интерфейсы на компьютере.

1) Отключаем питание.
2) Вытаскиваем разъем при помощи мышки и тянем в колонку с модулями.
3) Выбираем модемный модуль и вставляем его на пустое место.

И включаем питание обратно.

Такую же операцию проделываем с маршрутизатором.

1) Выключаем питание.
2) Выбираем модуль и вставляем в один из свободных слотов.
3) Включаем питание обратно.

Теперь перейдем к настройке маршрутизатора. Суть в том, что через CLI повесить адреса на новые модули не получится, ибо в CPT это оказалось не предусмотрено. Но можно это сделать через вкладку «Config».

Дальше создадим 2 DHCP пула (то есть на каждый компьютер свою подсеть) и заранее исключим IP-адреса, которые уже используются на маршрутизаторе.

> Router#configure terminal — *переходим в режим глобальной конфигурации.*
> Router(config)#ip dhcp excluded-address 192.168.1.1 — *исключаем из выдачи адрес, который висит на интерфейсе Modem 0/3/0.*
> Router(config)#ip dhcp excluded-address 192.168.2.1 — *исключаем из выдачи адрес, который висит на интерфейсе Modem 0/3/1.*
> Router(config)#ip dhcp pool FOR-PC1 — *создаем пул для PC1*
> Router(dhcp-config)#network 192.168.1.0 255.255.255.0 — *анонсируем сеть.*
> Router(dhcp-config)#default-router 192.168.1.1 — *указываем основной шлюз.*
> Router(config)#ip dhcp pool FOR-PC2 — *создаем пул для PC2*
> Router(dhcp-config)#network 192.168.2.0 255.255.255.0 — *анонсируем сеть.*
> Router(dhcp-config)#default-router 192.168.2.1 — *указываем основной шлюз.*

Для того, чтобы компьютеры смогли подсоединиться, они должны пройти аутентификацию. Для этого создадим логин и пароль (он будет одинаковым для двух компьютеров).

Router(config)#username admin password nimda — *создаем пользователя с логином: admin и паролем:nimda.*

Сохраняем конфигурацию и переходим к настройке нашего облака. Для начала посмотрим, какой интерфейс куда смотрит.

Теперь им нужно присвоить номера. Для простоты воспользуюсь 3-х значными номерами.

> Modem4 = 111
> Modem5 = 222
> Modem1 = 333
> Modem0 = 444

Немного не по порядку, но это не главное. На данном этапе базовая настройка закончена и настало время проверить работу. Открываю PC1 и перехожу на вкладку Desktop.

Я думаю, как вы догадались, нужна вкладка Dial-up. Открываем ее.

Открывается окно, где надо ввести логин, пароль и номер. Вводим, как на картинке. И нажимаем кнопку Dial.

Видим, что соединение установилось. О чем свидетельствует Status: Connected и зеленые огни на схеме. Раз соединение установлено, запросим IP адрес у DHCP сервера. Переходим на вкладку Desktop и выбираем IP Configuration.

Выбираем DHCP, и компьютер получает адрес из нужной подсети. Отлично!
Теперь проделаем аналогичные процедуры со вторым компьютером.

Обратите внимание, что логин и пароль тот же, а номер другой.

Установилось соединение.

Получаем адрес 192.168.2.2. Адрес получен из второго пула, как и было задумано. Воспользуемся командой ping и достучимся до PC2 с компьютера PC1.