Сети для самых маленьких. Часть вторая. Коммутация

Мы в телеграм:
Системный администратор
Книги для Системного Администратора
Чат системных администраторов

После скучного рассказа о подключении к кошкам переходим к настройке сети. В этот раз темы будут для новичков сложные, для старичков избитые. Впрочем сетевым аксакалам едва ли удастся почерпнуть что-то новое из этого цикла. Итак, сегодня:
а) аккуратно впитываем теорию о коммутаторах, уровнях сетевой модели, понятии инкапсуляции и заголовков (не пугайтесь — еще не время),
б) собираем спланированную в нулевой части цикла сеть,
в) настраиваем VLAN'ы, разбираемся с access и trunk-портами и тегированными Ethernet-фреймами,
г) соотносим текущие знания со стеком протоколов TCP/IP и моделью OSI (да, наконец-то мы ее коснёмся).

https://youtu.be/OtY-Z6_PDpU

Перед тем, как вы обратитесь к практике, настоятельно рекомендуем почитать нулевую часть, где мы всё спланировали и запротоколировали.

Теория

Для начала необходимо определится с определениями и детерминировать терминологию. В начале пути с этим могут быть трудности, несмотря на горы википедии и прорву технических статей.
Рассмотрим самые общие термины, поскольку что такое коммутатор и маршрутизатор вы, во-первых, представляете, во-вторых, по ходу не раз ещё их затронем. Итак, тронулись:
СКС — структурированная кабельная система — это определение вы в любом яндексе найдёте. На деле это все провода, розетки, патчпанели и патчкорды, то есть грубо говоря, это физика вашей сети в узком смысле, в широком — это совокупность сетей: ЛВС, телефонные сети, системы видеонаблюдения и прочее. Это отдельный очень большой и порой сложный пласт знаний и технологий, который вообще не имеет точек пересечения с настройкой, поэтому к нему мы более обращаться не будем. Привели мы этот термин по большей части для того, чтобы читатель чувствовал отличие от следующего.
ЛВС = Локальная Вычислительная Сеть = LAN = Local Area Network. Актуальность слова “Вычислительная” сейчас можно поставить под сомнение, так же, как в слове ЭВМ. Всё-таки, говоря о современных сетях и устройствах, мы давно уже не держим в уме термин «вычисления», несмотря на то, что глубинная суть осталась неизменной. В этом плане буржуйские термин более универсален и даёт более простое представление о своём значении.
Итак, локальная сеть — в первом приближении — это сеть вашей организации. Вот, к примеру,

обслуживаем мы сейчас сеть компании «Лифт ми Ап» с двумя офисам, так вот сети этих двух офисов и будут являться локальной сетью.
При втором приближении, локальной называют сеть, которая находится под управлением одного сетевого администратора. То есть, например, вы отвечаете за районный сегмент сети провайдера, в таком случае ваша районная сеть со всеми подсетями будет являться локальной, в то время, как вышестоящая сеть и сети других районов уже нет, так как за них отвечает уже другие люди. Вообще говоря, это уже MAN — Metropolian Area Network — сеть уровня города. Но в какой-то степени к ней можно применить понятие LAN и уж тем более VLAN.
С точки зрения меня, как абонента этого провайдера, моя локальная сеть — это всё, что до моего домашнего роутера. Интуитивно, наверно, все понимают о чём идёт речь.
Именно с локальными сетями мы и будем иметь дело в ближайших выпусках.

И последнее, что хотелось бы отметить в связи с ЛВС — это IP-адресация.
Все вы знаете, что когда вы включаете какой-нибудь домашний Wi-Fi-роутер в сеть, он обычно выдаёт вам IP-адрес, вроде 192.168.1.x. Почему именно 192.168 в начале?

Дело в том, что все IP адреса делятся на приватные (private, он же внутренний, “серый”, локальный), и публичные. Публичные используются в интернет, каждый адрес уникален, их распределение контролирует организация IANA(Internet Assigned Numbers Authority).

Приватные используются для адресации хостов (ну, строго говоря, не хостов, а интерфейсов) внутри ЛВС, их распределение никто не контролирует. Для них выделили три диапазона адресов (по одному из каждого класса):

10.0.0.0 — 10.255.255.255
172.16.0.0 — 172.31.255.255
192.168.0.0 — 192.168.255.255

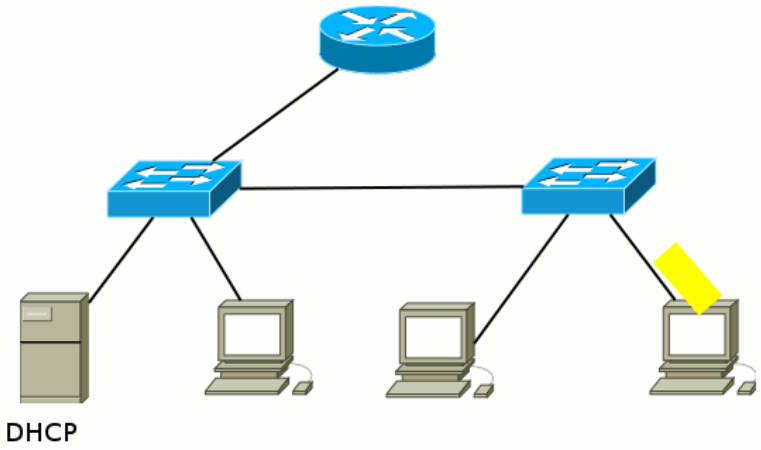
Важный момент касаемо “классов адресов”, об этом уже как-то писали на хабре: классов адресов уже давно не существует. Позже мы обстоятельно поговорим об адресации, но пока рекомендация такая: забыть про существование классов адресов, чтобы не попасть впросак на собеседовании или в разговоре.

Это те адреса, которые вы можете использовать в своей частной сети. Они вполне могут повторяться (и повторяются) в разных локальных сетях, и за её пределы они не выходят. Приватный адрес на то и приватный, поэтому любой пакет с адресом из диапазонов, указанных выше, попавший к провайдеру, будет отбрасываться.

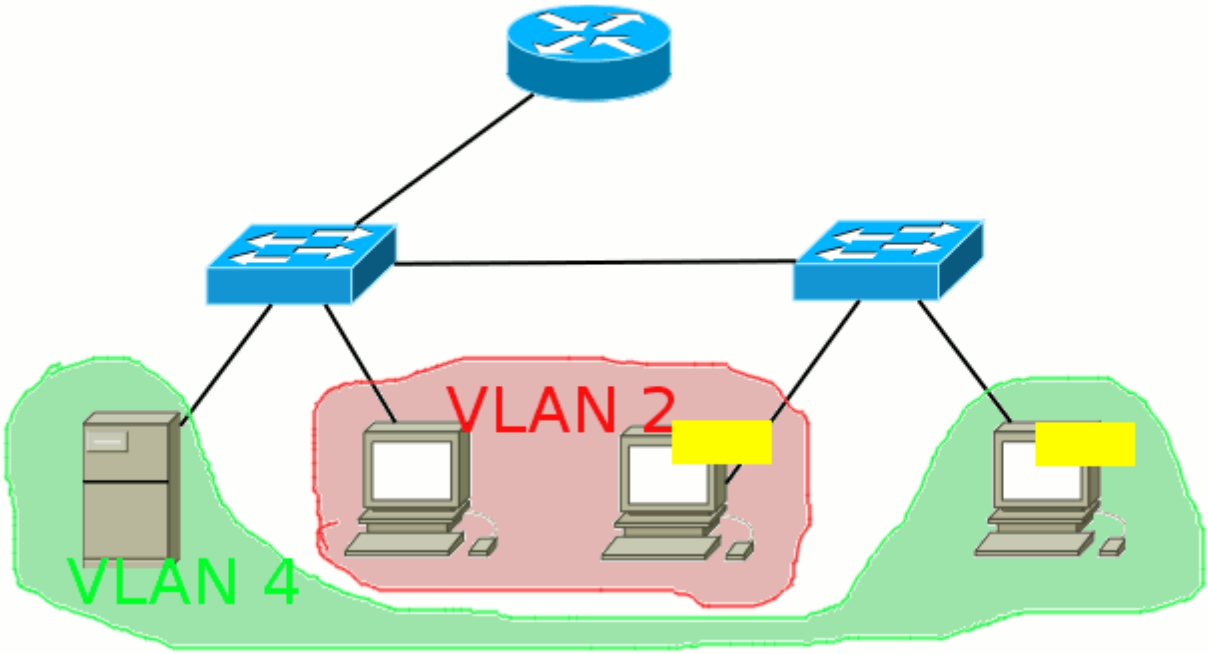
Если вернуться к нашей старой схеме то вы увидите, что для своей сети мы выбрали приватные адреса из диапазона 172.16.0.0 — 172.31.255.255.
Достаточно подробно об IP-адресах можно почитать тут и тут.
У всех провайдеров и во внутренней сети любой крупной организации используются именно эти *серые* подсети. Если только вы не государственный ВУЗ, которому в своё время выпала сеть на несколько тысяч публичных адресов: Кемеровский Государственный Университет, к примеру, не озадачивается NAT’ом и прочей чепухой — просто на все компьютеры университетской сети раздаются белые IP.

Широковещательный домен — область сети, в которой происходит обмен широковещательными сообщениями, и устройства могут отправлять друг другу сообщения непосредственно, без участия маршрутизатора.

О чём это мы тут говорим? Ну, например, послал ваш компьютер широковещательный запрос в сеть в поисках DHCP-сервера. *Фрейм* этот (он же *кадр*) адресован всем устройствам и имеет MAC-адрес получателя FF:FF:FF:FF:FF:FF. Сначала он попадает на коммутатор, с которого его копии рассылаются на все порты. Потом часть попадает на другие компьютеры, часть уходят в соседние коммутаторы, кто-то доходит до маршрутизатора, а одну копию принимает-таки DHCP-сервер. И вот участок сети, внутри которого могут жить эти кадры и называется широковещательным доменом. А кончают свою жизнь они на конечных хостах (компьютеры, серверы) или на маршрутизаторах, которые их отбрасывают, если они им не предназначены:

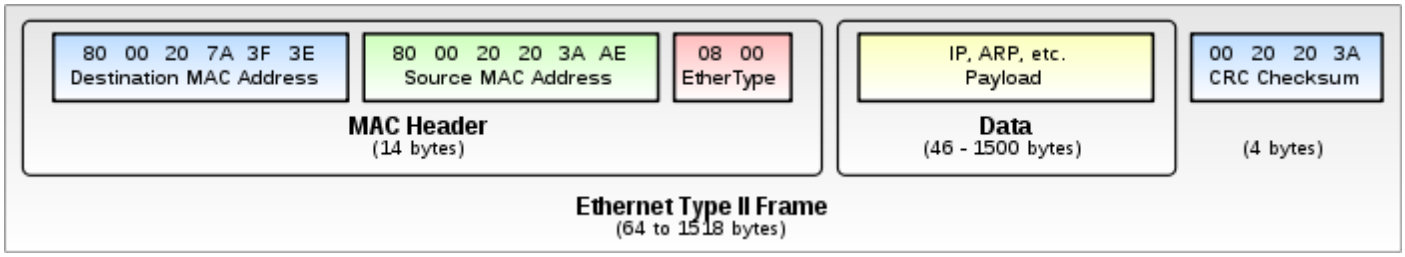


Если же на коммутаторе заведены VLAN’ы, то они также разделяют широковещательные домены, потому что пакет между ними обязательно должен проходить через маршрутизатор, который отбросит широковещательные сообщения. Таким образом, один VLAN — это один широковещательный домен.



Ещё раз: у нас есть три способа разграничить широковещательные домены:
1) Поставить маршрутизатор и разнести хосты в разные подсети,
2) Разделить сеть VLAN'ами,
3) Порвать кабель.

Ну и самая жесть, которой часто сторонятся начинающие: OSI. Open System Interconnection. Вообще в двух словах, чтобы мозг не захламить за одно занятие. Эту модель называют эталонной, потому что в реальном мире дело не дошло до реализации. Но она само совершенство, поэтому инженеры и админы вворачивают это слово повсюду.
В основе лежат 7 китов сетевой иерархии: 7 уровней. Сегодня коснёмся двух нижних: **первый — физический** — это представление информации в виде сигналов, прямо скажем, битов. Задача этого уровня сгенерировать электрический, оптический или радиосигнал, передать его в среду и принять его. К нему относится вся физика: интерфейсы, кабели, антенны, медиаконвертеры (конвертеры среды), репитеры, старые хабы. В общем низкоуровневая это работа. Это первый уровень модели OSI и стека TCP/IP.
Второй — канальный. На этом уровне работают коммутаторы. Идентификатор устройства здесь, это MAC-адрес. У каждого узла (компьютер, маршрутизатор, ноутбук, IP-телефон, любой Wi-Fi-клиент) есть этот уникальный адрес, который однозначно определяет устройство в локальной сети. В теории MAC-адреса не должны повторяться вообще, но на практике такое однако случается и в рамках одного широковещательного домена может приводить к сложноотлавливаемым проблемам.
Наиболее известным протоколом этого уровня является Ethernet. Данные на этом уровне передаются кусками, каждый из которых называется Ethernet-фрейм (он же Ethernet-кадр, он же PDU канального уровня). Что он представляет из себя?

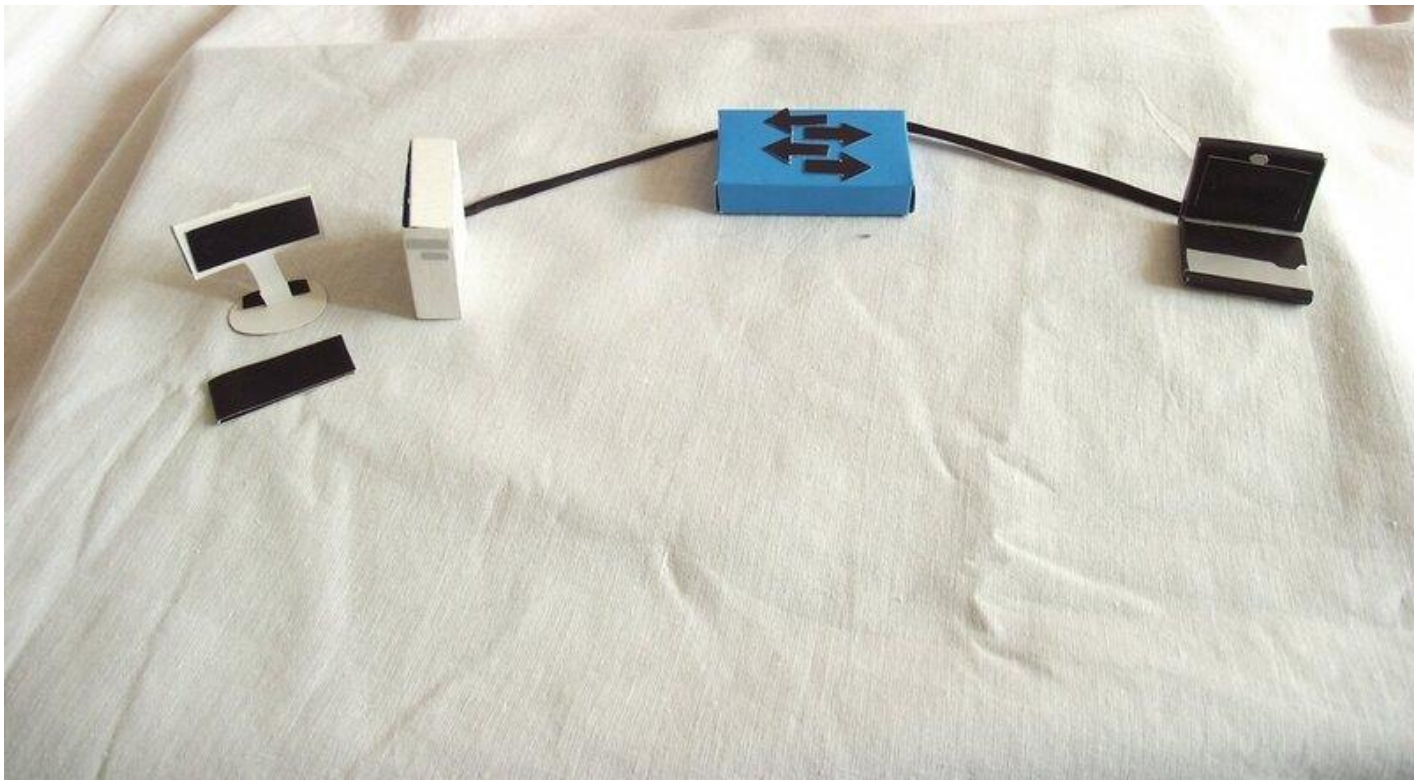


Картинку гнусно спёрли из википедии, потому что красивее не нарисуем
Payload — это полезная нагрузка — данные сетевого уровня, которые вкладываются (*инкапсулируются*) в кадр. *MAC Header (Заголовок)* — это служебная информация канального (второго) уровня. Самые важные пока для нас элементы — это source MAC-address (адрес отправителя кадра) и Destination MAC-address (адрес получателя кадра).

Третий уровень — сетевой (IP)
Четвёртый — транспортный (TCP, UDP)
С пятого по седьмой — сеансовый, представления и прикладной (в стеке TCP/IP они не различаются и называются просто прикладным. На нём работают протоколы вроде HTTP, FTP, telnet и многие другие)

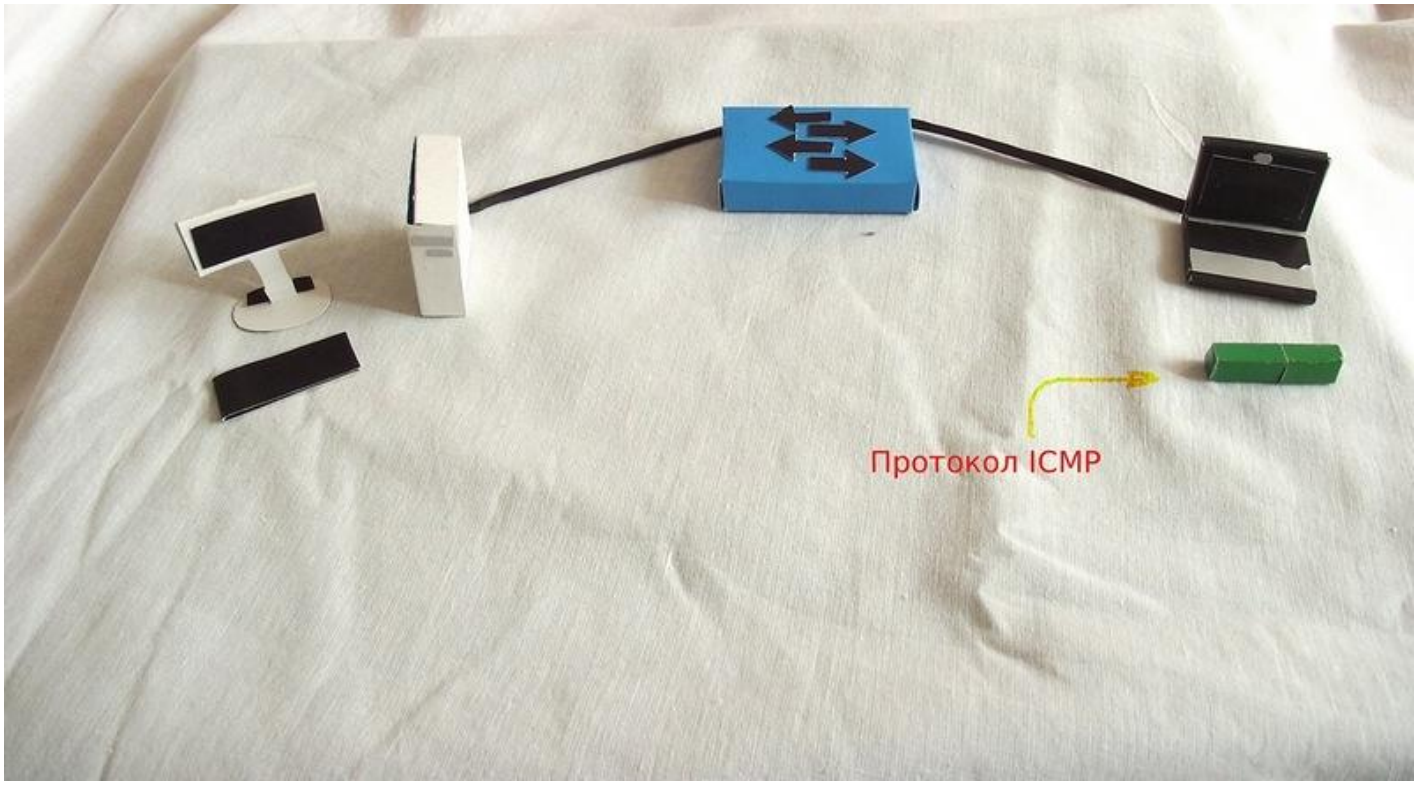
Сегодня мы акцентируемся на 1-м и 2-м уровнях, особенно на втором. Третьего и четвертого коснёмся в следующих выпусках.

Теперь проследим нелёгкий путь кадра.
Состояние покоя сети — утопия.

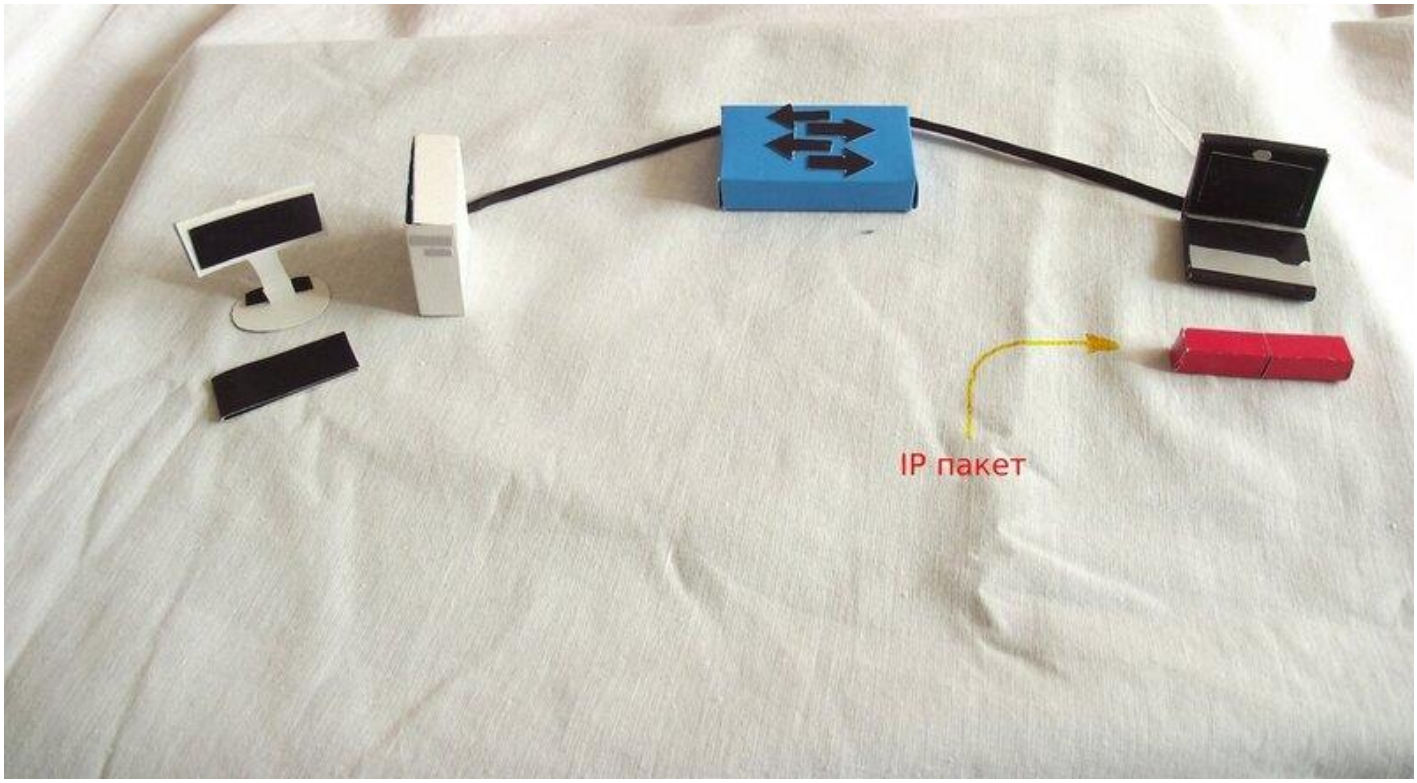


Вы пытаетесь пропинговать, например, адрес соседнего компьютера командой **ping 192.168.1.118**. Данные этого приложения показаны фиолетовым параллелепипедом.

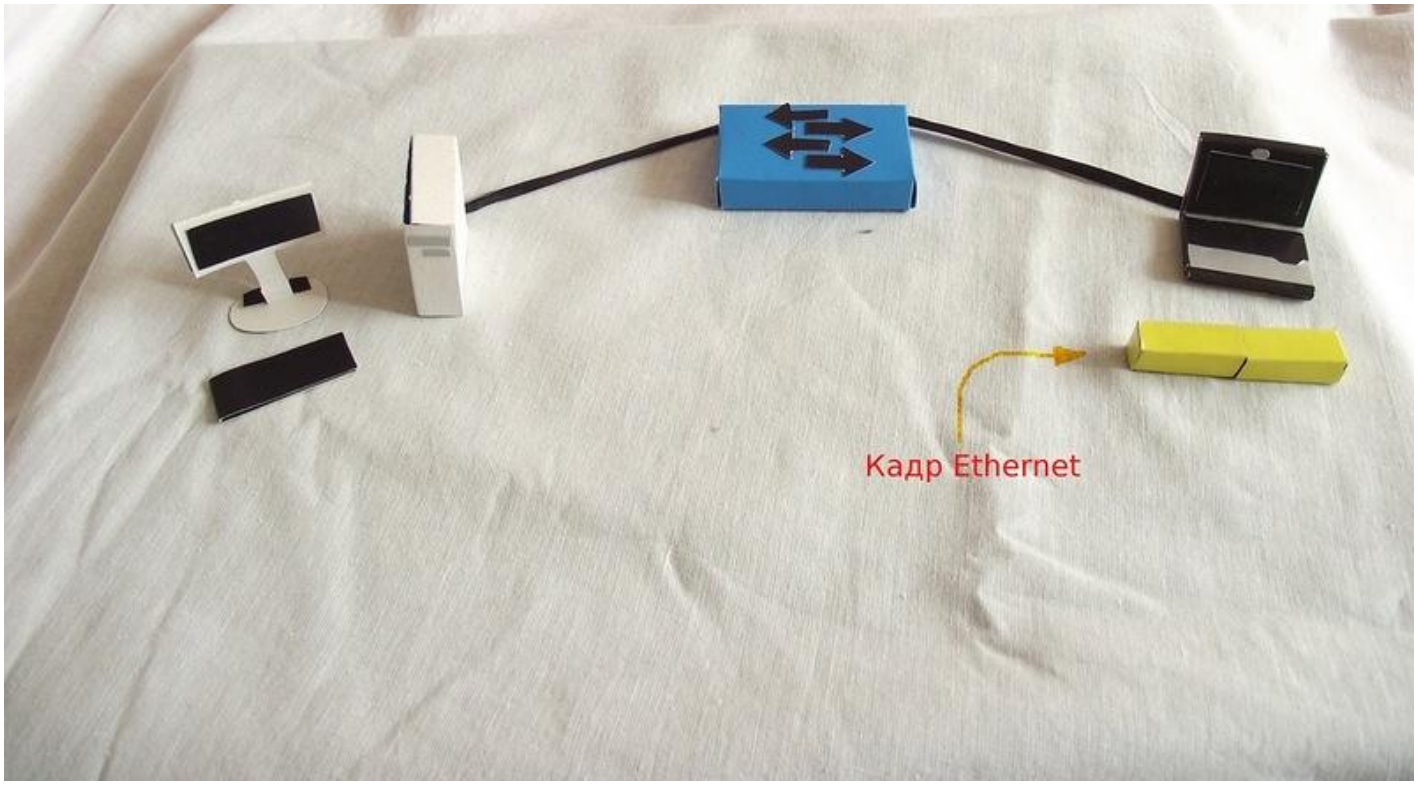
За это отвечает протокол ICMP. В него инкапсулируется информация от приложения — это означает, что к данным 5-го уровня добавляется заголовок со служебной информацией 4-го уровня.



Его данные упаковываются (инкапсулируются) в IP-пакеты, где в заголовке указан IP-адрес получателя (192.168.1.118) и IP-адрес отправителя — логические адреса.



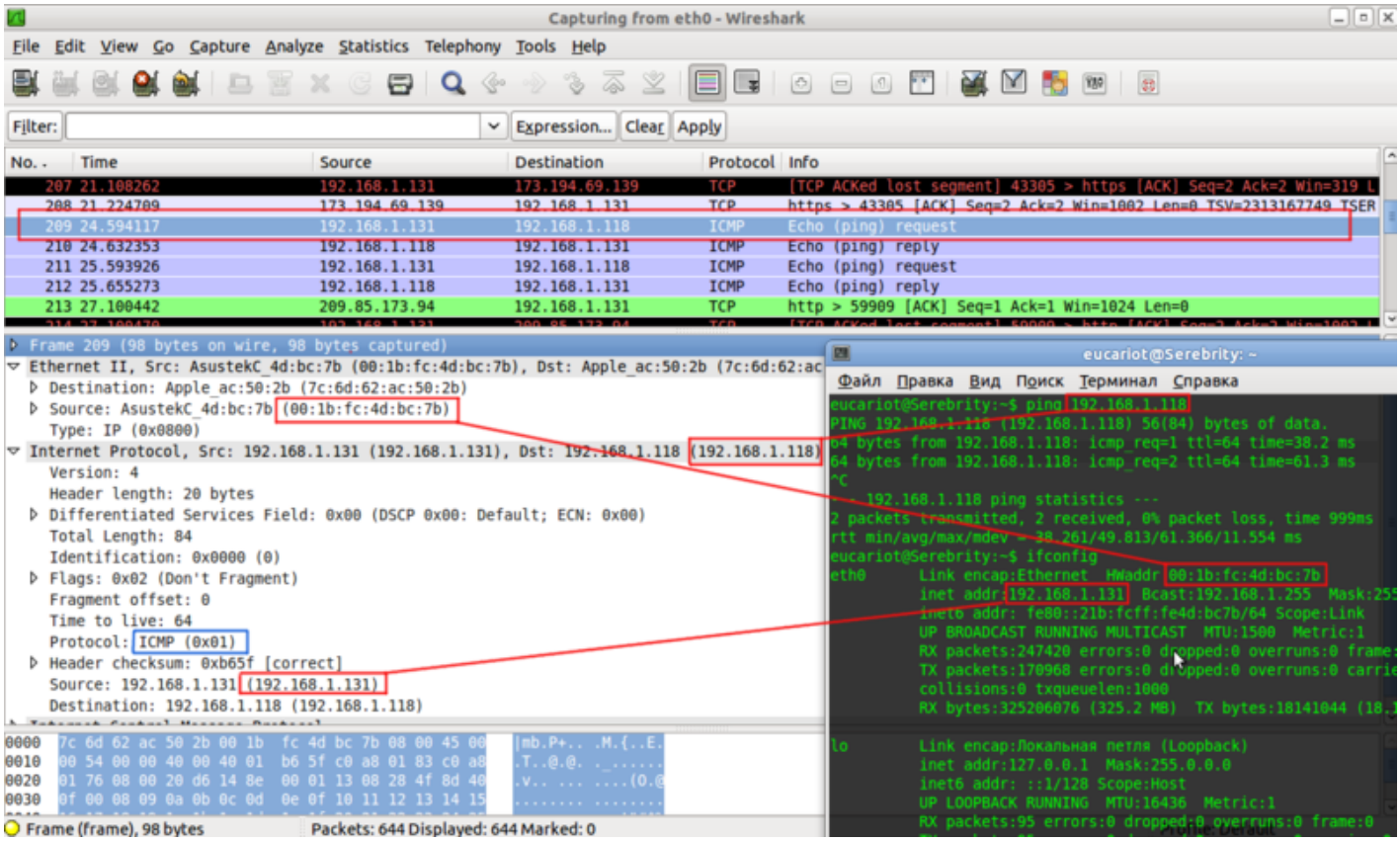
А затем всё это инкапсулируется в Ethernet-кадры с MAC-адресами отправителя и получателя — физическими адресами.



При формировании кадров в заголовке в качестве MAC-адреса источника (source) подставляется адрес вашего компьютера, а адресом получателя (destinantion) будет MAC-адрес компьютера — владельца IP-адреса 192.168.1.118 (о механизмах такого преобразования поговорим в следующий раз). То есть если бы вы смогли сфотографировать кадр, то вы бы увидели все эти данные в разрезе, так сказать.

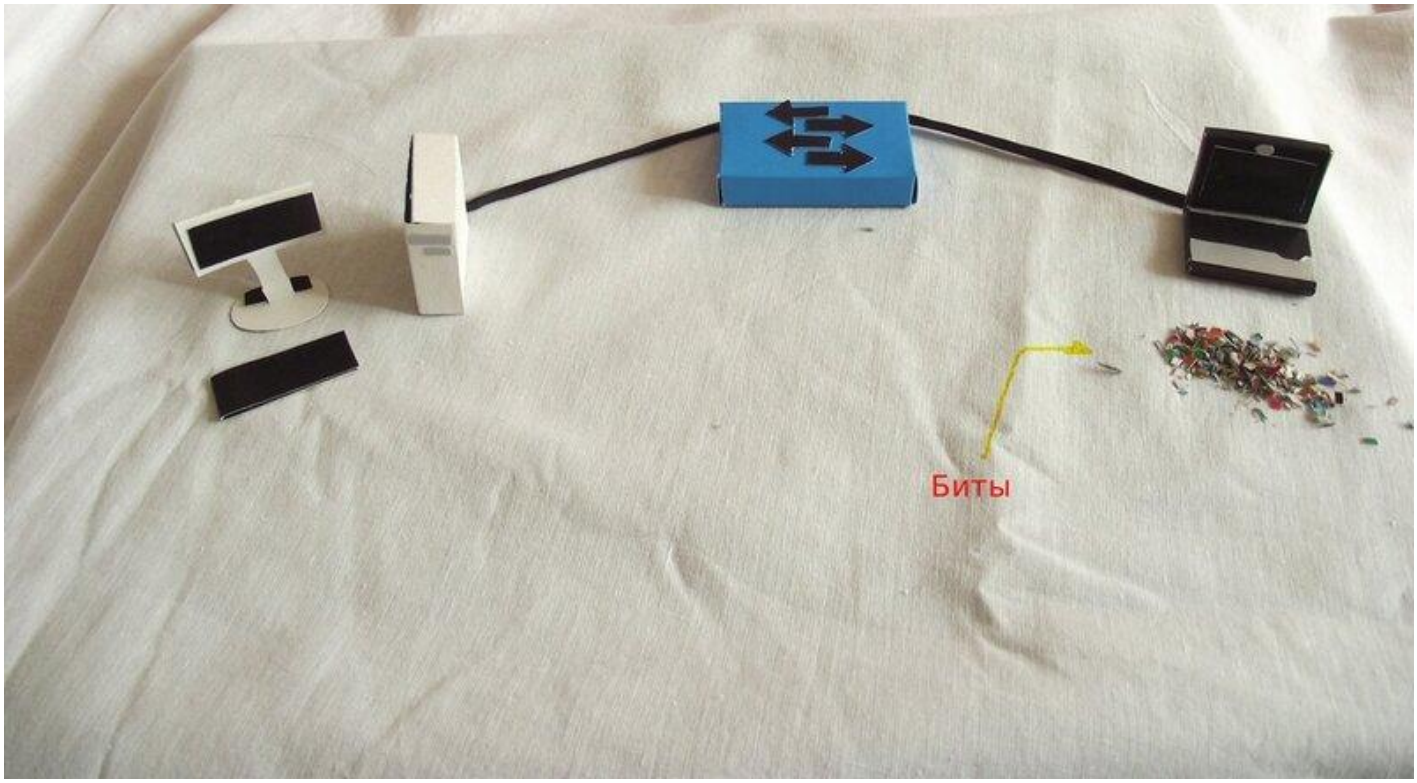
На самом деле, нет ничего проще: запускаете какой-нибудь анализатор трафика, например, замечательный Wireshark и Ethereal, на своём компьютере и пингуете другой хост. Вот такую картину вы

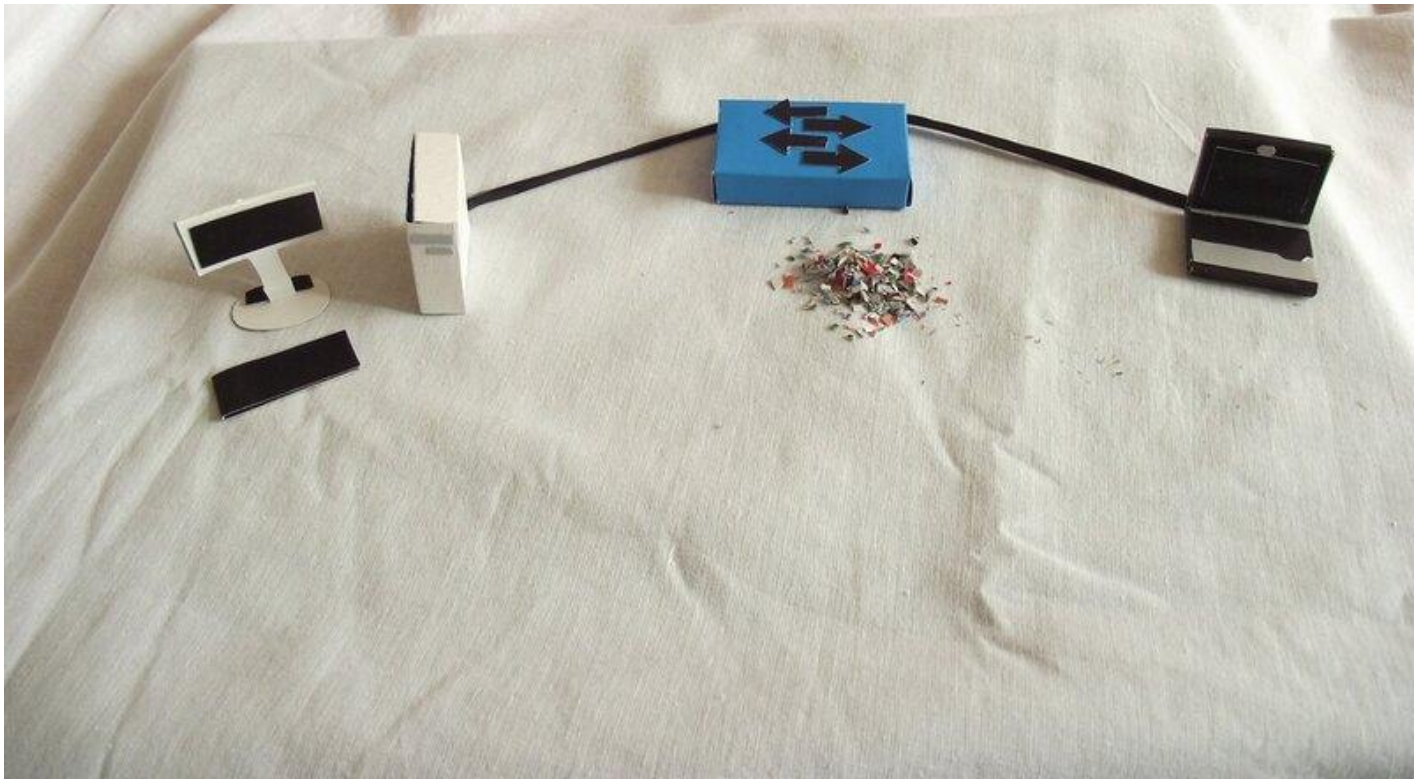
сможете лицезреть:



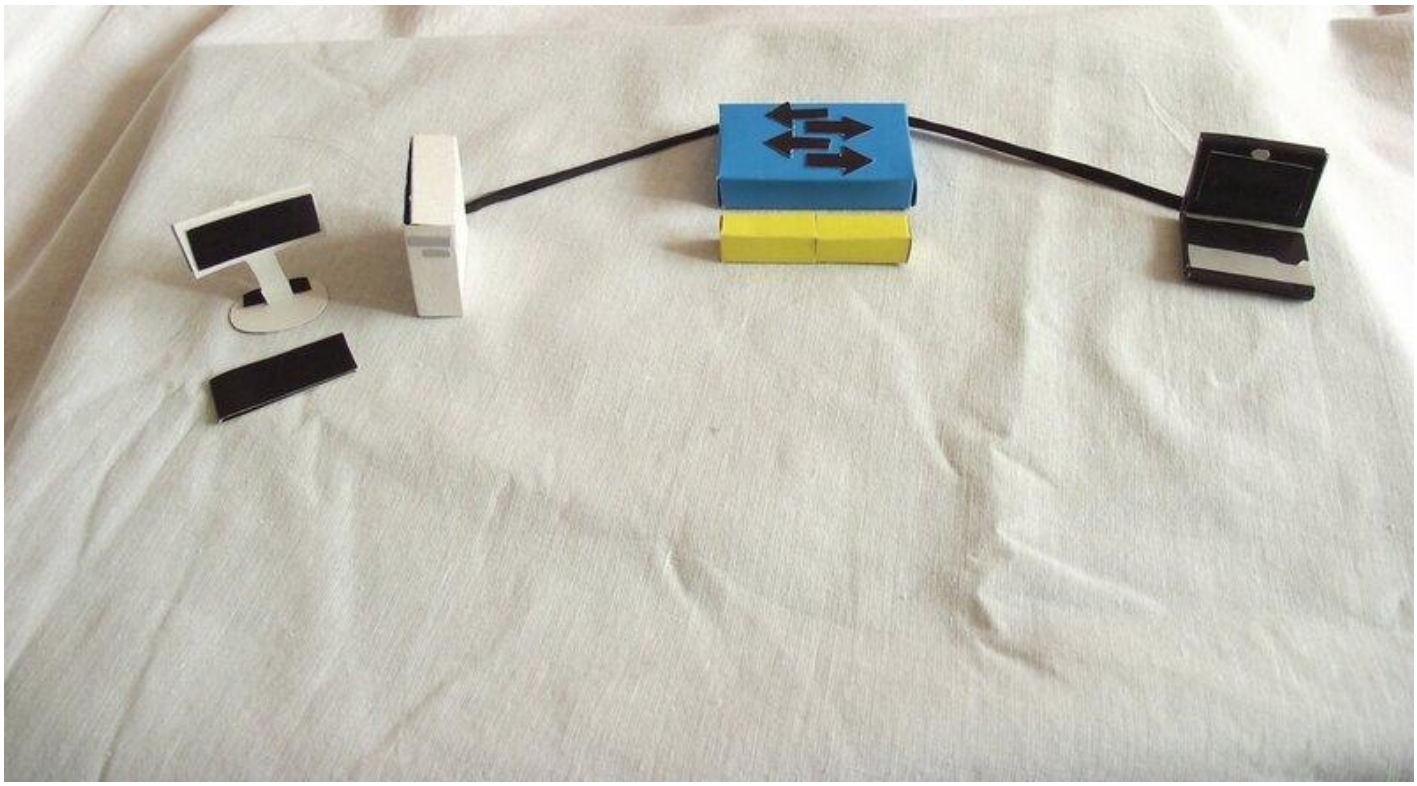
Вы это можете сделать прямо сейчас, читая эти строки, просто установив и запустив анализатор трафика.

В последнюю очередь сетевая карта вашего компьютера дробит фрейм на биты и отправляет их в кабель.



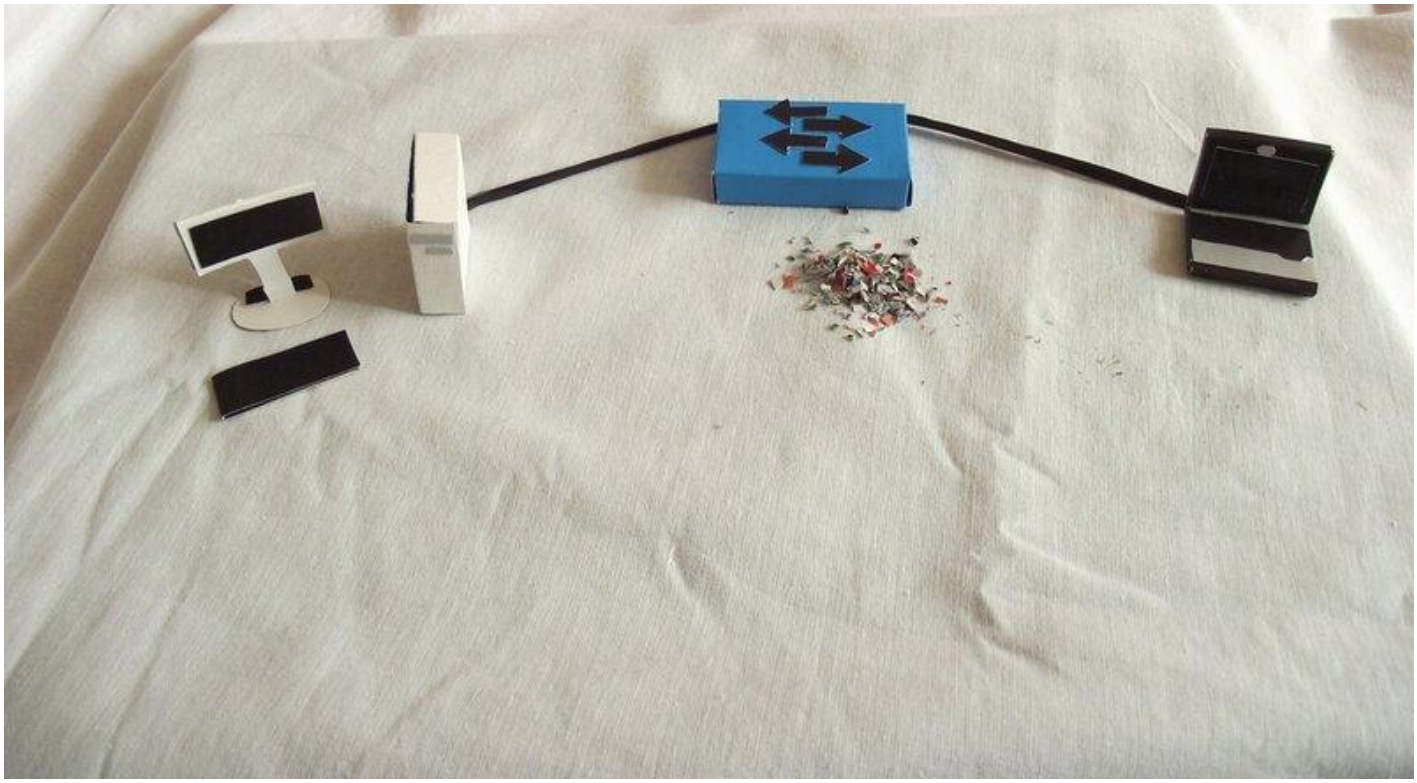


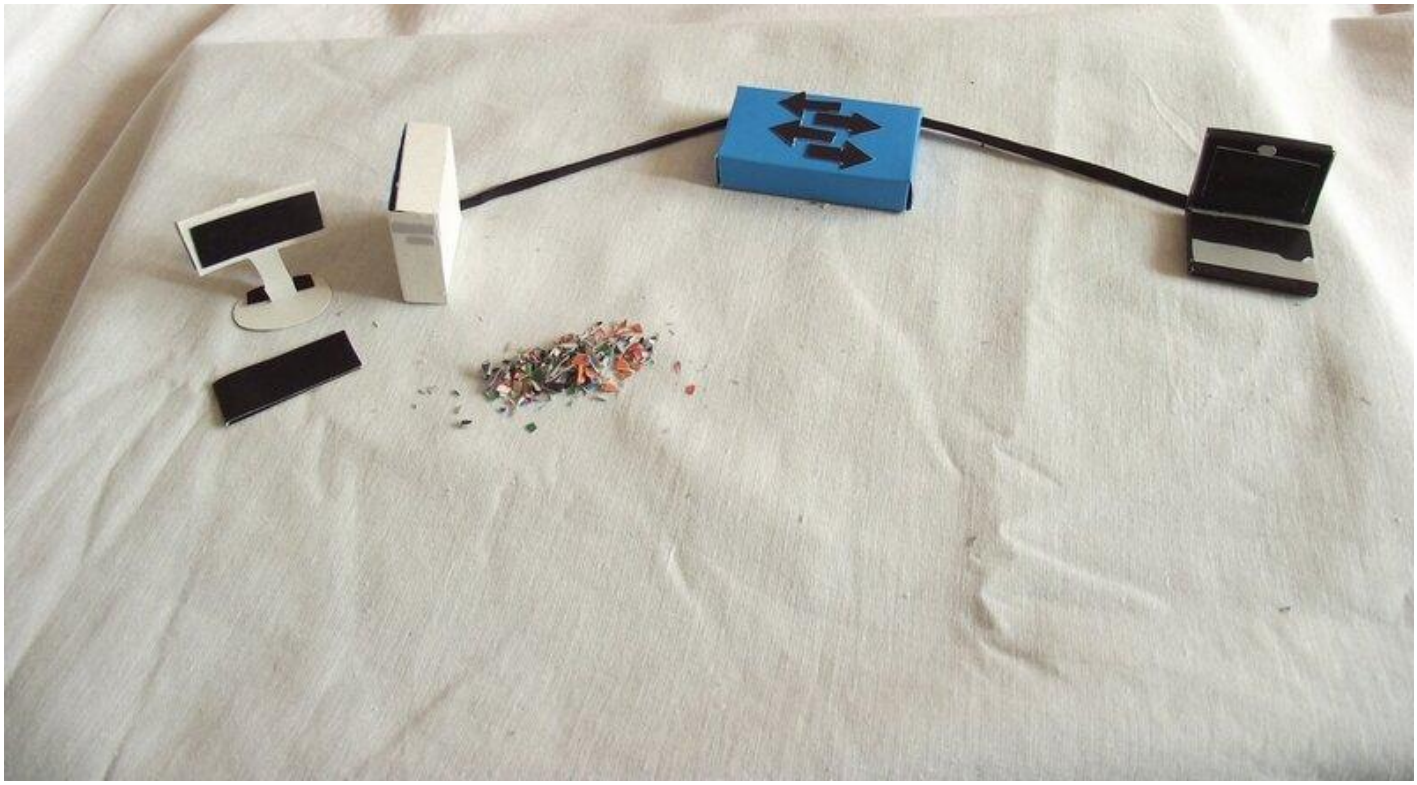
Коммутатор из поступивших битов собирает первоначальный кадр



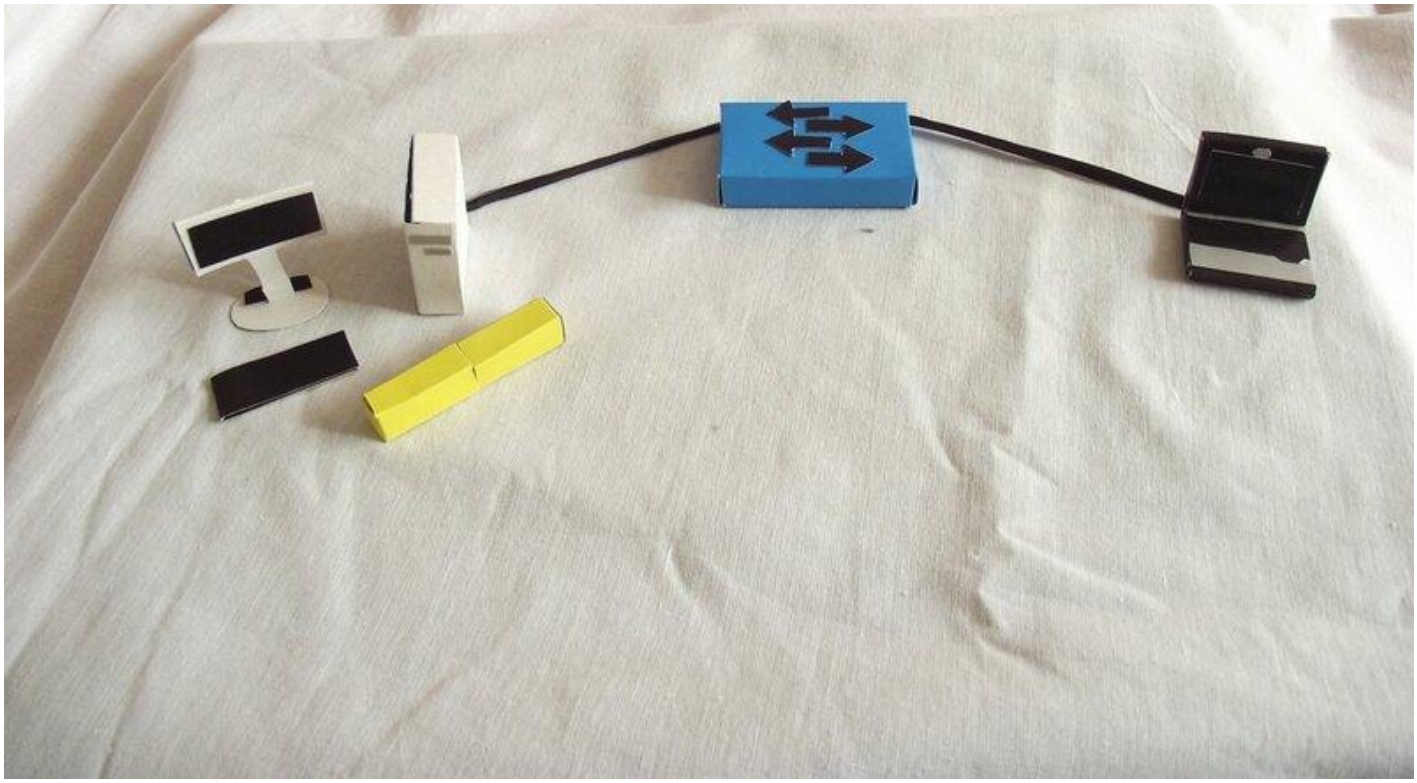
Далее начинается интеллектуальный труд: из заголовка извлекается адрес получателя, перетрясается таблица MAC-адресов на предмет совпадения и, как только оное найдено, кадр без изменений отправляется в указанный порт. Если же адреса пока ещё нет или кадр пришёл широковещательный, то он направляется на все порты, кроме того, откуда пришёл.

Если адреса отправителя в таблице до сих пор не было, то в этот момент коммутатор добавит его. Естественно, кадр опять передаётся в виде битов — это закон электроники, и вы должны просто всегда иметь это в виду.

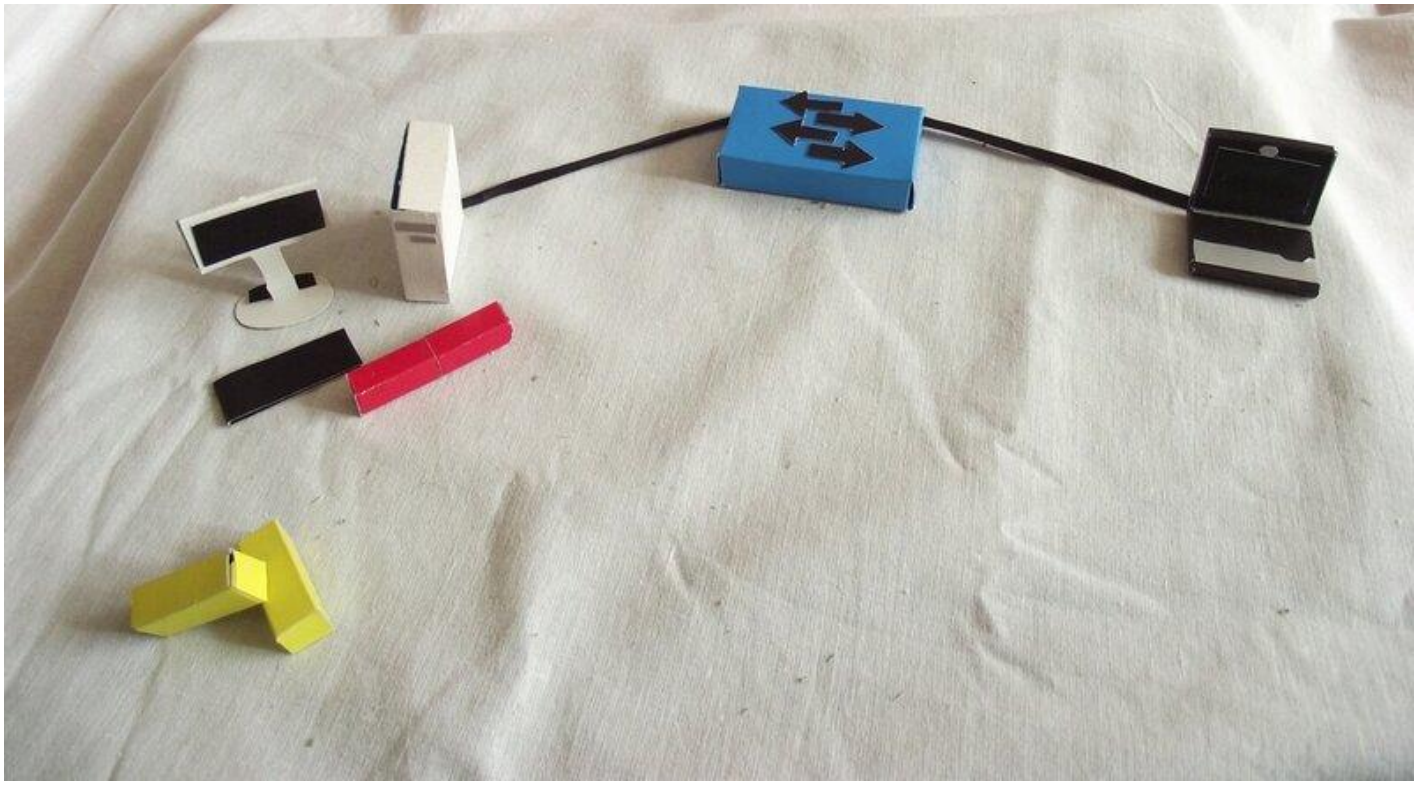




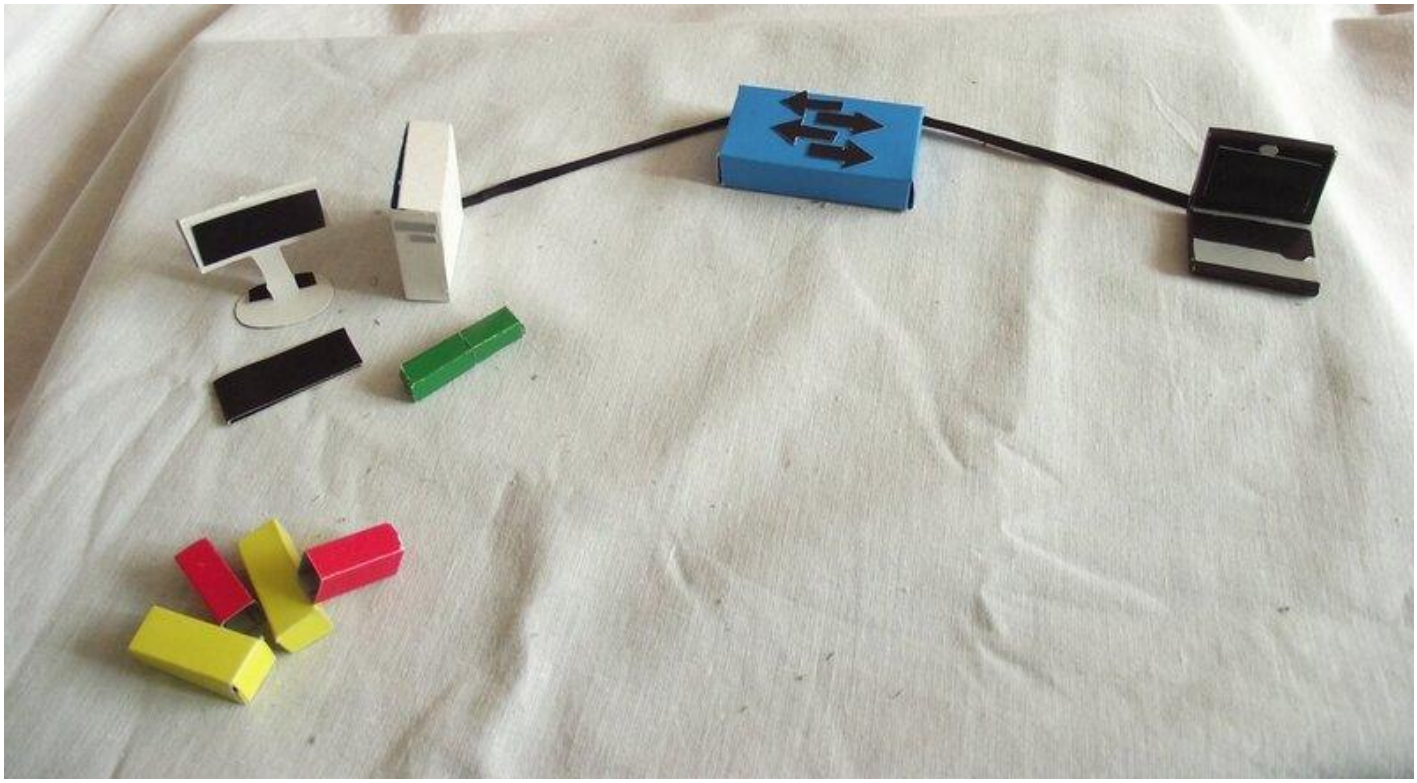
Конечный хост, получив поток битов, собирает из них кадр, ещё только предполагая, что он предназначается ему.

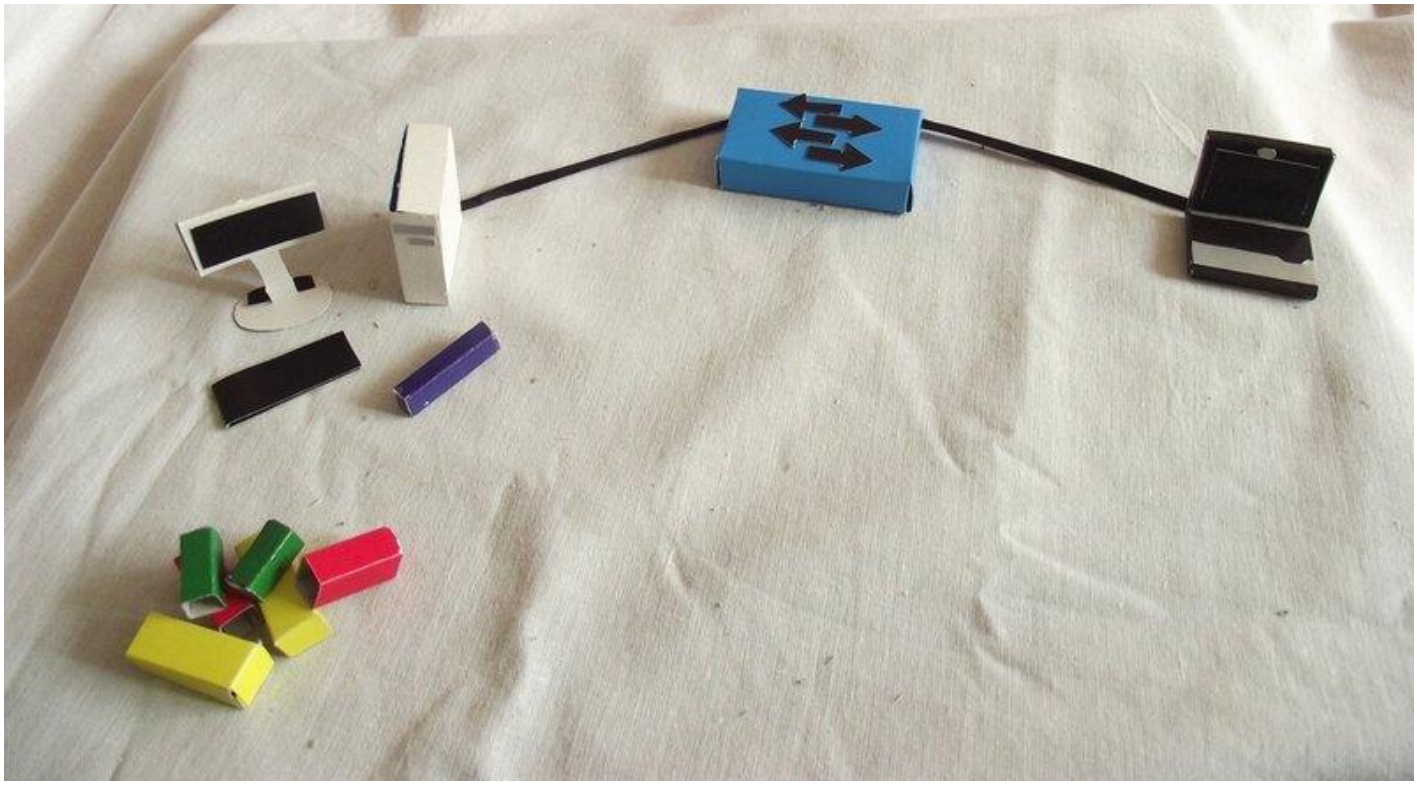


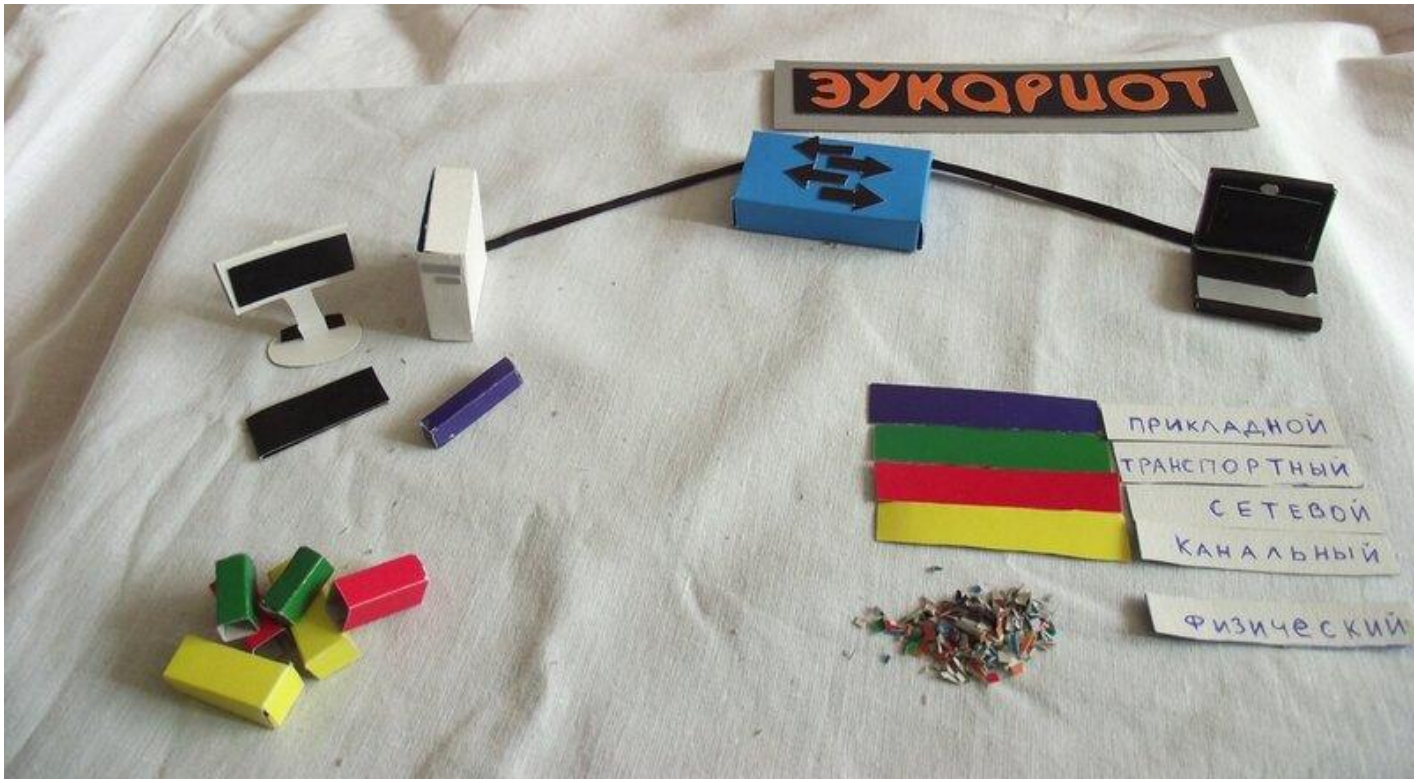
Далее он сравнивает MAC-адрес получателя со своим и, если они совпадают, то заголовок второго уровня отбрасывается, а IP-данные передаются на обработку вышестоящему протоколу. Если адреса не совпадают, то кадр отбрасывается вместе со всем содержимым.



Далее сравниваются IP-адрес получателя и этого устройства. Если совпадают, то заголовок сетевого уровня отбрасывается, и данные передаются транспортному уровню (ICMP)







Конечный хост обработал ICMP-запрос (echo-request) и готов послать ICMP-ответ (echo-reply) вашему компьютеру с адресом 192.168.1.131 и далее пункты 1-3 повторяются уже для нового кадра

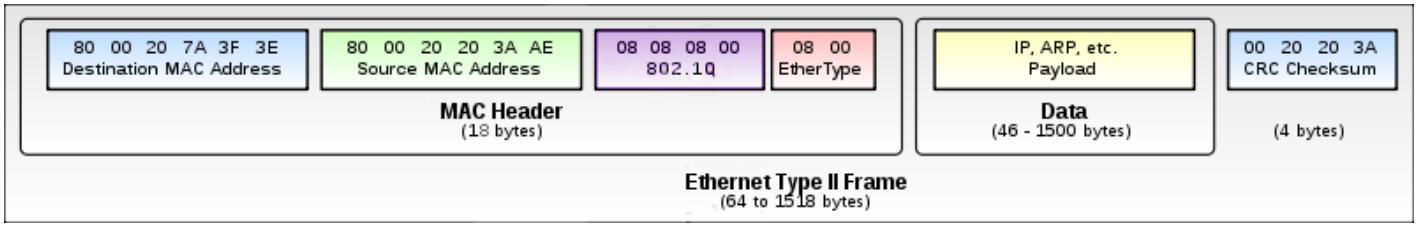
То, о чём мы писали до сих пор — это принцип работы любого коммутатора. Так делают даже простые длинки за 300 рублей.

VLAN

Ну а теперь, давайте, коллеги, финальный рывок: добавим сюда ещё VLAN'ы.

С ними работают уже только управляемые коммутаторы.
Напомним, что вланы нужны для разделения сетей. Соответственно появляется некий идентификатор, которым маркируется трафик разных подсетей на коммутаторе.
Говоря о VLAN'ах, часто используют заклинание 802.1q. Это и есть стандарт, описывающий как именно кадр маркируется/тегируется. Пугаться такого шифра не стоит. Так же, например, Wi-Fi описывается стандартом 802.11n, а протокол аутентификации — 802.1x. Нам с этим предстоит столкнуться в будущем, поэтому отложите это в своей энергонезависимой памяти.

Что же именно происходит на кухне коммутации?
Внутрь фрейма после Source MAC-адреса добавляется ещё одно поле, очень грубо говоря, содержащее номер VLAN'а. Длина, выделенная для номера влана равна 12 битам, это означает, что максимальное число вланов 4096. Мы хотим обратить внимание молодых инженеров на такие подробности. Дело в том, что мы в своём цикле в силу объективных причин, не можем обо всём рассказать, но такие вопросы, во-первых, часто задают на собеседованиях, во-вторых, это просто надо знать.



Кадры первого влана обычно не тегируются — он является родным вланом (native vlan).

Каждый коммутатор принимает теперь решение на основе этой метки-тега (или его отсутствия).
В таблицу MAC-адресов добавляется ещё столбец с номером VLAN'а и при поиске пары MAC-адрес/порт теперь будет сравниваться тег кадра с номером VLAN'а в таблице.

Существует два типа портов:
1. *Access port* — порт доступа — к нему подключаются, как правило, конечные узлы. Трафик между этим портом и устройством нетегированный. За каждым access-портом закреплён определённый VLAN, иногда этот параметр называют PVID. Весь трафик, приходящий на этот порт от конечного устройства, получает метку этого влана, а исходящий уходит без метки.

2. *Trunk port*. У этого порта два основных применения — линия между двумя коммутаторами или от коммутатора к маршрутизатору. Внутри такой линии, называемой в народе, что логично, транком, передаётся трафик нескольких вланов. Разумеется, тут трафик уже идёт с тегами, чтобы принимающая сторона могла отличить кадр, который идёт в бухгалтерию, от кадра, предназначенного для ИТ-отдела. За транковым портом закрепляется целый диапазон вланов.
Кроме того, существует вышеупомянутый *native vlan*. Трафик этого влана не тегируется даже в транке, по умолчанию это 1-й влан и по умолчанию он разрешён. Вы можете переопределить эти параметры.
Нужен он для совместимости с устройствами, незнакомыми с инкапсуляцией 802.1q. Например, вам нужно через Wi-Fi мост передать 3 влана, и один из них является вланом управления. Если Wi-Fi-модули не понимают стандарт 802.1q, то управлять ими вы сможете, только если этот влан настроите, как native vlan с обеих сторон.

Что происходит в сети с вланами?

1) Итак, от вашего компьютера с IP-адресом, например, 192.168.1.131 отправляется пакет другому компьютеру в вашей же сети. Этот пакет инкапсулируется в кадр, и пока никто ничего не знает о вланах, поэтому кадр уходит, как есть, на ближайший коммутатор.
2) На коммутаторе этот порт отмечен, как член, например, 2-го VLAN'а командой

```
Switch0#interface fa0/1

Switch0(config)#description "I am using simple frames"

Switch0(config-if)#switchport mode access

Switch0(config-if)#switchport access vlan 2
```

Это означает, что любой кадр, пришедший на этот интерфейс, автоматический тегируется: на него вешается

ленточка с номером VLAN'а. В данном случае с номером 2.
Далее коммутатор ищет в своей таблице MAC-адресов среди портов, принадлежащих 2-му влану, порт, к которому подключено устройство с MAC-адресом получателя.
3) Если получатель подключен к такому же access-порту, то ленточка с кадра отвязывается, и кадр отправляется в этот самый порт таким, каким он был изначально. То есть получателю также нет необходимости знать о существовании вланов.
4) Если же искомый порт, является транковым, то ленточка на нём остаётся.

```
Switch(config)#interface fa0/2

Switch(config-if)#description "I am using tagged frames"

Switch(config-if)#switchport mode trunk
```

Попробуем провести аналогию с реальными миром. Вы с другом, например, пакеты-туристы и летите отдыхать дикарями самолётом авиалиний Ethernet Airlines. Но по дороге вы поссорились, и потому, когда в аэропорту назначения, вас спрашивают в какую гостиницу вас везти, вы отвечаете "Рога", а ваш товарищ говорит "Копыта". И сразу после этого вас инкапсулируют в разные кадры-машины: вас в такси с тегом "Таксопарк "На рогах", а вашего товарища с его грузом в КамАЗ с тегом "Транспортная компания "В копыто". Теперь вам нельзя на автобусные полосы, а вашему другу под знаки, запрещающие проезд грузовиков. Так вот две гостиницы — это MAC-адреса назначения, а ограничения по маршруту — порты других вланов. Петляя, по улочкам, вам, как IP-пакету не о чем беспокоиться — кадр-автомобиль доставит вас до места назначения, и, грубо говоря, в зависимости от тега на каждом перекрёстке будет приниматься решение, как ехать дальше.

FAQ:

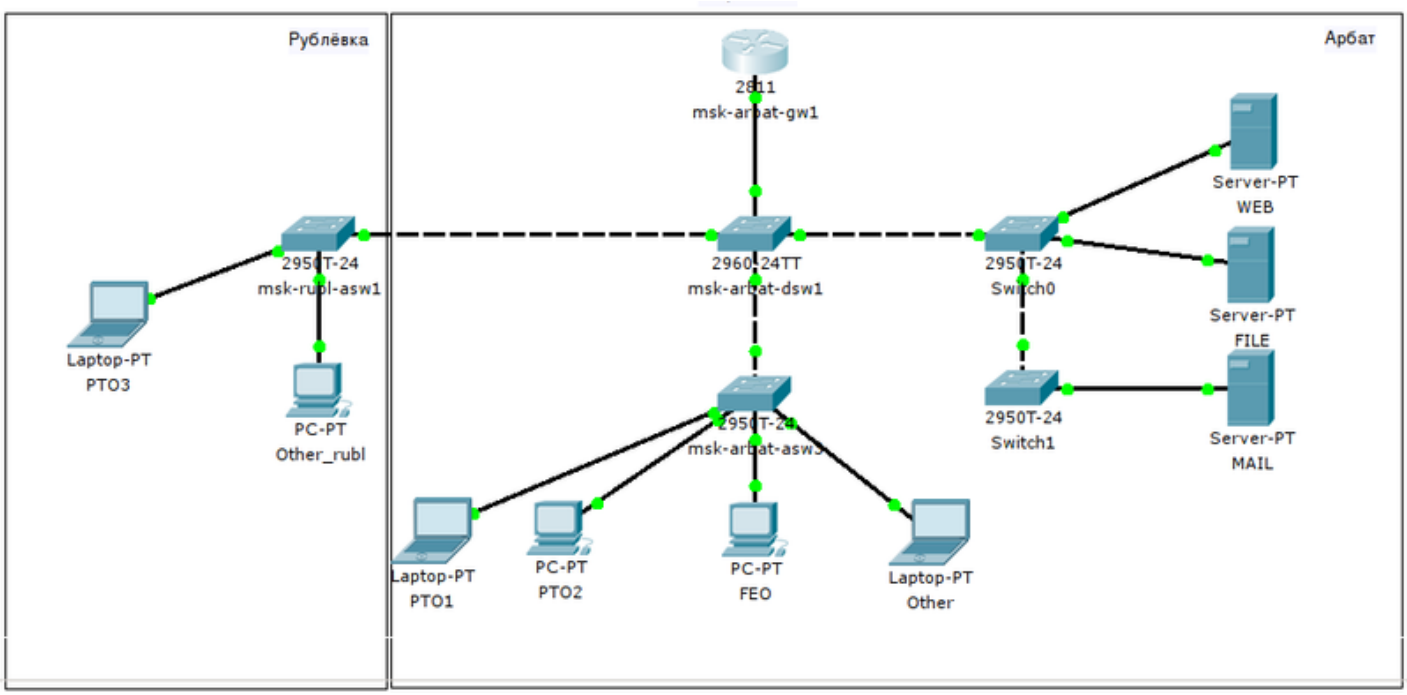
Q: Что произойдёт, если тегированный кадр прилетит на access-порт?
A: Он будет отброшен.
Q: Что произойдёт, если нетегированный кадр прилетит на trunk-порт?
A: Он будет помещён в Native VLAN. По умолчанию им является 1-й VLAN. Но вы можете поменять его командой **switchport trunk native vlan 2**
В этом случае все кадры, помеченные 2-м вланом будут уходить в этот порт нетегированными, а нетегированные кадры, приходящий на этот интерфейс, помечаться 2-м вланом.
Кадры с тегами других вланов останутся неизменными, проходя, через такой порт.
Q: Можно ли конечным узлам (компьютерам, ноутбукам, планшетам, телефонам) отправлять тегированные кадры и соответственно подключать их к транковым портам?
A: Да, можно. Если сетевая карта и программное обеспечение поддерживает стандарт 802.1q, то узел может работать с тегированными кадрами.

Q: Что будет с тегированными кадрами, если они попадут на обычный неуправляемый коммутатор или другое устройство, не понимающее стандарт 802.1q?
A: Скорее всего, свич его отбросит из-за увеличенного размера кадра. Зависит от разных факторов: производитель, софт (прошивка), тип форвардинга (cut-through, store-and-forward).

Практика. Настройка сети “Лифт ми Ап”

Ну и наконец-то обратимся к настройке. Вива ля практис!

Будет у нас такая сеть:



Вспомним, как мы её планировали:

- VLAN
- IP-план
- план коммутации

Советуем на дополнительной вкладке отрыть их, потому что мы будем обращаться туда периодически.

Мы могли бы сейчас броситься сразу настраивать всё по порядку: полностью одно устройство, потом другое. Но так не будет, пожалуй, понимания значения процессов.

Порты доступа (access)

Поэтому начнём с простого: настроим два порта на msk-arbat-asw3 как access для влана 101 (ПТО):

```
msk-arbat-asw3(config)#interface FastEthernet0/1

msk-arbat-asw3(config-if)#description PTO

msk-arbat-asw3(config-if)#switchport mode access

msk-arbat-asw3(config-if)#switchport access vlan 101

% Access VLAN does not exist. Creating vlan 101
```

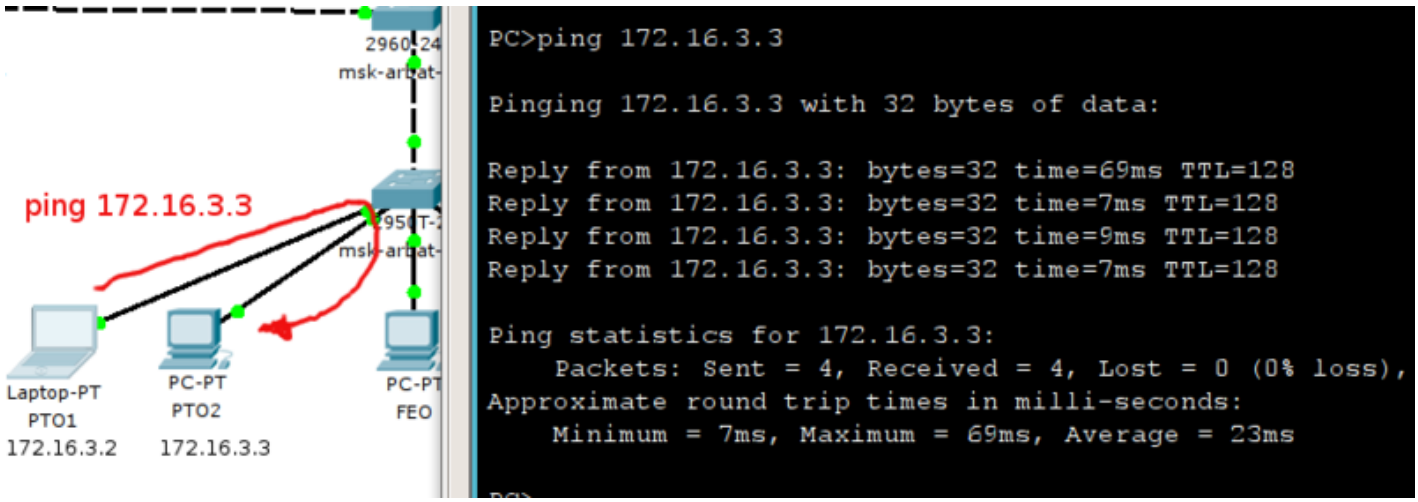
```
msk-arbat-asw3(config)#interface FastEthernet0/2

msk-arbat-asw3(config-if)#description PTO

msk-arbat-asw3(config-if)#switchport access vlan 101

msk-arbat-asw3(config-if)switchport mode access
```

Все настройки делаем сразу в соответствии с планом.

Заметили, что коммутатор ругается на отсутствие влана? Тут надо быть аккуратным. Некоторые версии ПО работают несколько нелогично.
Даже если вы его не создадите, то настройки применятся и при отладке на первый взгляд всё будет нормально, но связи не будет. Причём коварство заключается в том, что фраза **Creating vlan 101** вовсе не означает, что этот самый влан будет создан. Поэтому отправляемся в режим глобальной конфигурации и создаём его (а заодно и все другие вланы, нужные на этом коммутаторе):

```
msk-arbat-asw3>enable
msk-arbat-asw3#configure terminal
msk-arbat-asw3(config)#vlan 2
msk-arbat-asw3(config-vlan)#name Management
msk-arbat-asw3(config-vlan)#vlan 3
msk-arbat-asw3(config-vlan)#name Servers
msk-arbat-asw3(config-vlan)#vlan 101
msk-arbat-asw3(config-vlan)#name PTO
msk-arbat-asw3(config-vlan)#vlan 102
msk-arbat-asw3(config-vlan)#name FEO
msk-arbat-asw3(config-vlan)#vlan 103
msk-arbat-asw3(config-vlan)#name Accounting
msk-arbat-asw3(config-vlan)#vlan 104
msk-arbat-asw3(config-vlan)#name Other
```

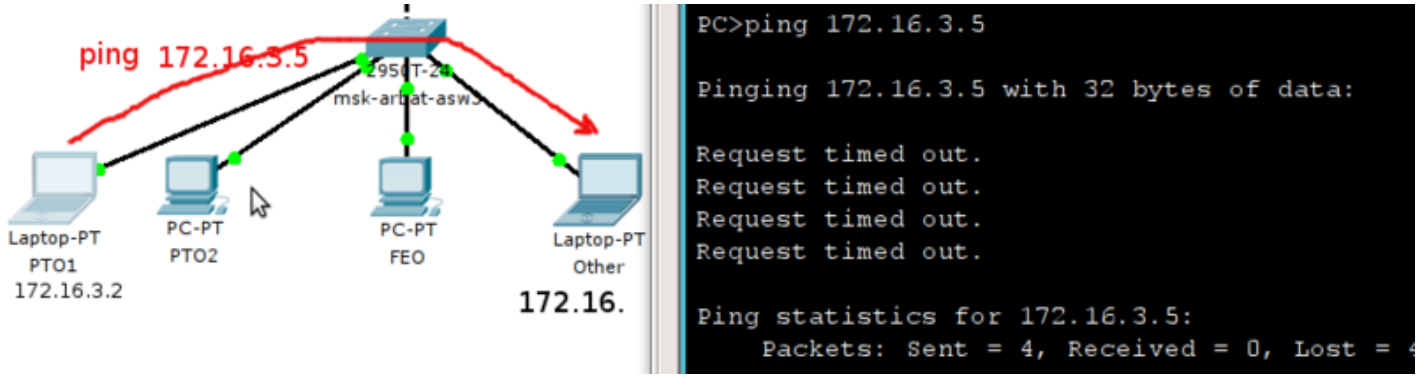
Теперь подключите компьютеры к портам FE0/1 и FE0/2, настройте на них адреса 172.16.3.2 и 172.16.3.3 с маской подсети 255.255.255.0 и шлюзом 172.16.3.1 и проверьте связь:



После того, как это получилось, настроим порт FE0/16, как access, для 104-го влана (сеть других пользователей):

```
msk-arbat-asw3(config)#interface FastEthernet0/16

msk-arbat-asw3(config-if)#description Other

msk-arbat-asw3(config-if)#switchport access vlan 104

msk-arbat-asw3(config-if)switchport mode access
```

Подключите к нему компьютер и настройте адрес из той же подсети, что ПТО, например, 172.16.3.5 с маской 255.255.255.0.
Если вы попытаетесь теперь пропинговать этот адрес, то у вас не должно этого получиться — компьютеры находятся в разных вланах и изолированы друг от друга:



То есть ещё раз, что происходит? От вашего компьютера приходит на 1-й порт широковещательный запрос: “Кто такой 172.16.3.5”, потому что сам компьютер пока не знает MAC-адреса получателя. Кадр, который несёт в себе этот запрос помечается, как принадлежащий 101-му VLAN’у в соответствии с портом, на который он поступил. И далее, чтобы узнать где-же находится компьютер 172.16.3.5, кадр рассылается на все порты-члены 101-го VLAN’а. А в их числе нет порта FE0/16, поэтому, естественно, этот адрес считается недостижимым, что приводит к ответу “Request timed out”.

Внимание! *Если в этом VLAN’е все-таки окажется устройство с таким IP, то это не будет тем же*

самым ноутбуком Other и при этом они не буду конфликтовать друг с другом, поскольку логически находятся в разных широковещательных доменах.

Транковые порты (trunk)

Итак, врата для вас открылись, теперь вам предстоит создать коридор — транк между тремя коммутаторами: msk-arbat-asw3, msk-arbat-dsw1 и msk-rubl-asw1.

Uplink портом на msk-arbat-asw3 является GE1/1. Ну а поскольку нам всё равно все влаы нужно будет пробросить, то сделаем это сейчас, то есть помимо 101 и 104 пропишем 2, 102 и 103:

```
msk-arbat-asw3(config)#interface GigabitEthernet1/1

msk-arbat-asw3(config-if)#description msk-arbat-dsw1

msk-arbat-asw3(config-if)#switchport trunk allowed vlan 2,101-104

msk-arbat-asw3(config-if)#switchport mode trunk
```

На самом деле на интерфейсе достаточно команды **#switchport mode trunk**, чтобы у вас через этот порт уже пошли тегированные кадры всех вланов, потому что по умолчанию транковый порт пропускает всё. Но мы же инженеры, а не эникейщики. Где это видано, чтобы безлимит творился за нашей спиной? Поэтому через нас проходит только то, что мы разрешаем. Как только вы дали команду **switchport trunk allowed vlan 101**, через порт не пройдёт кадр никаких вланов, кроме 101 (VLAN 1 ходит по умолчанию и нетегированным).

***Внимание!** Если вы хотите в транковый порт добавить ещё один влан, то вам необходимо использовать следующий синтаксис команды:*

```
msk-arbat-dsw1(config-if)#switchport trunk allowed vlan add 105
```

*В противном случае (написав **switchport trunk allowed vlan 105**) вы сотрёте все старые разрешения и добавите новый 105-й влан. И хорошо ещё, если при этом вы не потеряете доступ на этот коммутататор. Но за простой связи всё равно вы получите по пятое число)*

Переходим к msk-arbat-dsw1. На нём необходимо создать все влаы и настроить два порта:
GE1/2 в сторону msk-arbat-asw3
FE0/1 в сторону msk-rubl-asw1:

```
msk-arbat-dsw1(config)#interface GigabitEthernet1/2

msk-arbat-dsw1(config-if)#description msk-arbat-asw3

msk-arbat-dsw1(config-if)#switchport trunk allowed vlan 2,101-104
```

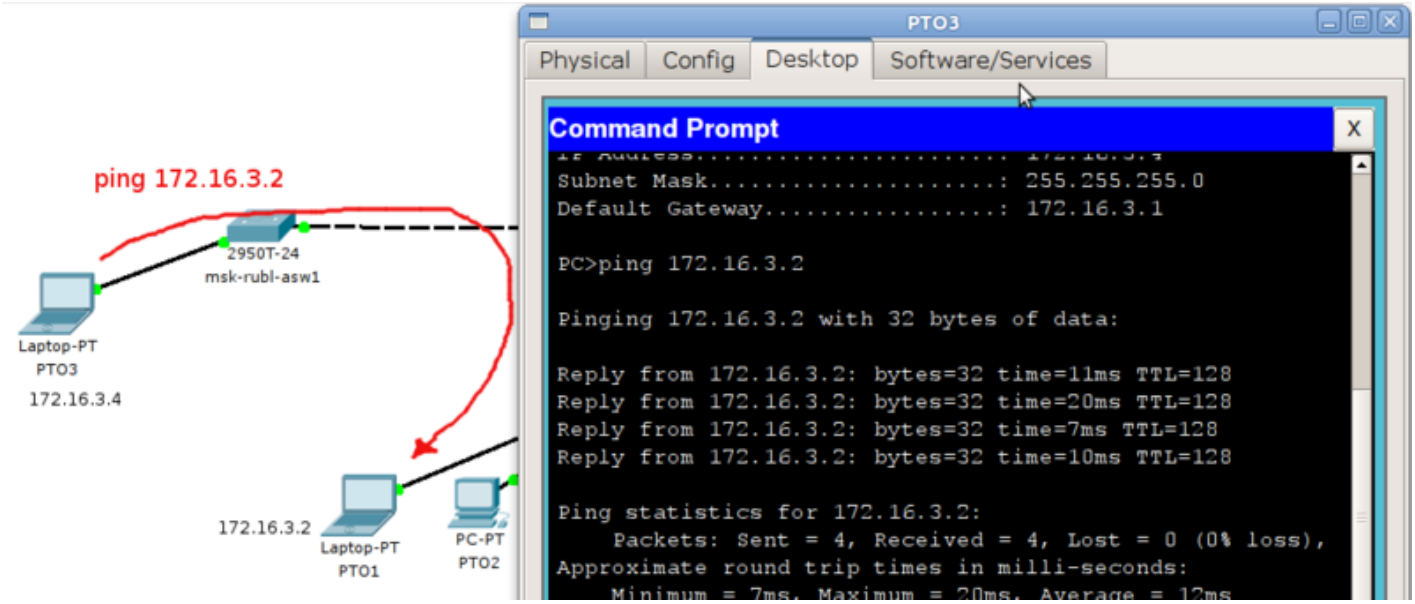
```
msk-arbat-dsw1(config-if)#switchport mode trunk

msk-arbat-dsw1(config)#interface FastEthernet0/1

msk-arbat-dsw1(config-if)#description msk-rubl-asw1

msk-arbat-dsw1(config-if)#switchport trunk allowed vlan 2,101,104

msk-arbat-dsw1(config-if)#switchport mode trunk
```

Ну и настроим, конечно, порты на msk-rubl-asw1:

```
msk-rubl-asw1(config)interface FastEthernet0/24

msk-rubl-asw1(config-if)switchport trunk allowed vlan 2,101,104

msk-rubl-asw1(config-if)switchport mode trunk

msk-rubl-asw1(config)#int FastEthernet0/1

msk-rubl-asw1(config-if)#description PTO

msk-rubl-asw1(config-if)#switchport mode access

msk-rubl-asw1(config-if)#switchport access vlan 101

% Access VLAN does not exist. Creating vlan 101
```

Снова нужно настроить вланы. И заметьте, при настройке транковых портов никаких сообщений нет.

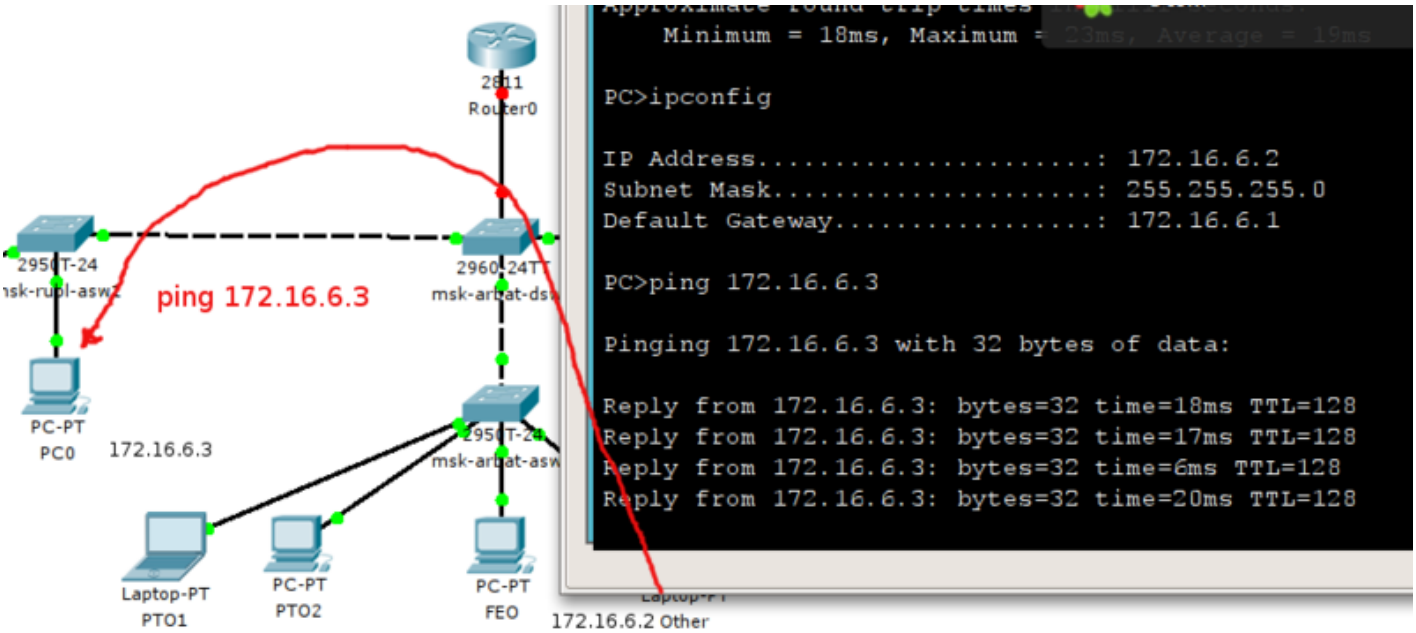
Если вы всё настроили правильно (в чём не приходится сомневаться), то с первого порта msk-rubl-asw1 вы увидите компьютеры ПТО, подключённые к msk-arbat-asw3.



Для уверенности проверим ещё и 104-й влан. Через транк мы его сюда уже доставили.

```
msk-rubl-asw1(config)#interface FastEthernet 0/16

msk-rubl-asw1(config-if)#switchport mode access

msk-rubl-asw1(config-if)#switchport access vlan 104
```

Подключаем компьютер к 16-му порт и настраиваем на нём IP-адрес 172.16.6.3 с маской 255.255.255.0 и шлюзом 172.16.6.1. А IP-адрес ноутбука на арбате поменяйте на 172.16.6.2 с теми же маской и шлюзом.



Сеть управления

Настроим IP-адрес для управления.
В наших лабах они не понадобятся, потому что мы настраиваем устройство через окно PT. А вот в реальной

жизни это вам жизненно необходимо.
Для этого мы создаём виртуальный интерфейс и указываем номер интересующего нас влана. А далее работаем с ним, как с самым обычным физическим интерфейсом.

msk-arbat-dsw1:

```
msk-arbat-dsw1(config)#interface vlan 2

msk-arbat-dsw1(config-if)#description Management

msk-arbat-dsw1(config-if)#ip address 172.16.1.2 255.255.255.0
```

msk-arbat-asw3:

```
msk-arbat-asw3(config)#vlan 2

msk-arbat-asw3(config)#interface vlan 2

msk-arbat-asw3(config-if)#description Management

msk-arbat-asw3(config-if)#ip address 172.16.1.5 255.255.255.0
```

С msk-arbat-asw3 запускаем пинг до msk-arbat-dsw1:

```
msk-arbat-asw3#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 60 percent (3/5), round-trip min/avg/max = 4/4/4 ms
```

Первые пару пакетов могут потеряться на работу протокола ARP: определение соответствия IP-адрес — MAC-адрес. При этом MAC-адрес, порт и номер влана добавляются в таблицу коммутатора.
Самостоятельно настройте IP-адреса сети управления на остальных коммутаторах и проверьте их доступность

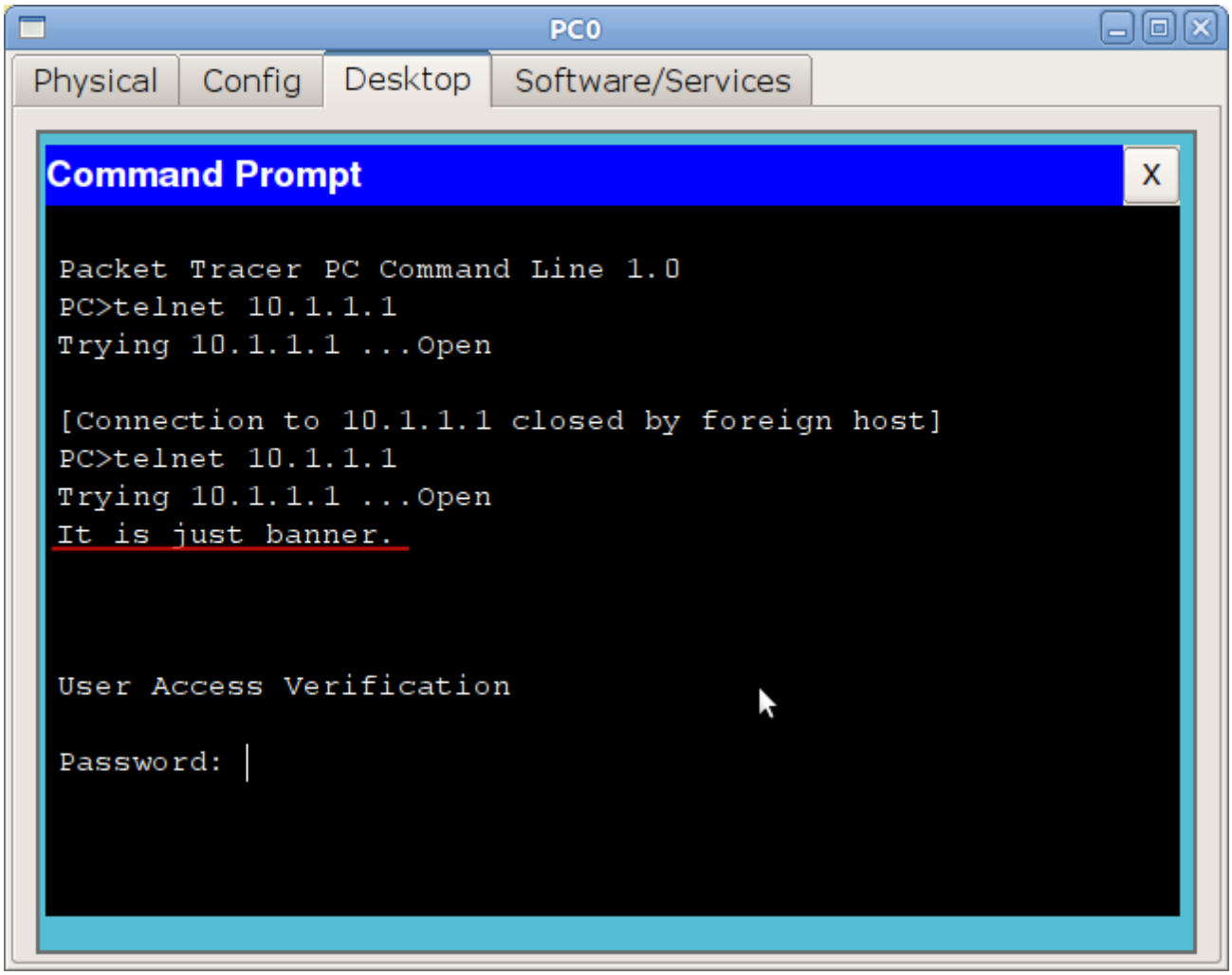
Собственно вот и вся магия. Зачастую к подобного рода действиям и сводится вся настройка, если вы не работаете в провайдере. С другой стороны, если вы работаете в провайдере, то, наверняка, такие вещи вам объяснять не нужно.

Если желаете знать больше об этом, читайте: VTP, QinQ, зарезервированные номера VLAN

Ещё один небольшой инструмент, который может немного увеличить удобство работы: banner. Это объявление, которое циска покажет перед авторизацией на устройство.

```
Switch(config)#banner motd q

Enter TEXT message. End with the character 'q'.

It is just banner.

q


Switch(config)#
```

После motd вы указываете символ, который будет служить сигналом о том, что строка закончена. В это примере мы поставили “q”.



Относительно содержания баннера. Существует такая легенда: хакер вломился в сеть, что-то там поломал\украл, его поймали, а на суде оправдали и отпустили. Почему? А потому, что на пограничном роутере(между интернет и внутренней сетью), в banner было написано слово “Welcome”. “Ну раз просят, я и зашел”)). Поэтому считается хорошей практикой в баннере писать что-то вроде “Доступ запрещен!”.

Для упорядочивания знаний по пунктам разберём, что вам необходимо сделать:

1) Настроить hostname. Это поможет вам в будущем на реальной сети быстро сориентироваться, где вы находитесь.

```
Switch(config)#hostname HOSTNAME
```

2) Создать все вланы и дать им название

```
Switch(config)#vlan VLAN-NUMBER

Switch(config-vlan)#name NAME-OF-VLAN
```

3) Настроить все access-порты и задать им имя

```
Switch(config-if)#description DESCRIPTION-OF-INTERFACE

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan VLAN-NUMBER
```

Удобно иногда бывает настраивать интерфейсы пачками:

```
msk-arbat-asw3(config)#interface range fastEthernet 0/6 — 10

msk-arbat-asw3(config-if-range)#description FEO

msk-arbat-asw3(config-if-range)#switchport mode access

msk-arbat-asw3(config-if-range)#switchport access vlan 102
```

4) Настроить все транковые порты и задать им имя:

```
Switch(config-if)#description DESCRIPTION-OF-INTERFACE

Switch(config-if)#switchport mode trunk

Switch(config-if)#switchport trunk allowed vlan VLAN-NUMBERS
```

5) **Не забывайте сохраняться:**

```
Switch#copy running-config startup-config
```

Итого: чего мы добились? Все устройства в одной подсети видят друг друга, но не видят устройства из другой. В следующем выпуске разбираемся с этим вопросом, а также обратимся к статической маршрутизации и L3-коммутаторам.
В общем-то на этом данный урок можно закончить. В видео вы сможете ещё раз увидеть, как настраиваются вланы. В качестве домашнего задания настройте вланы на коммутаторах для серверов.

Здесь вы можете скачать конфигурацию всех устройств:
liftmeup_configuration.zip.

P.S.
Важное дополнение: в предыдущей части, говоря о native vlan мы вас немного дезинформировали. На оборудовании cisco такая схема работы невозможна.
Напомним, что нами предлагалось передавать на коммутатор msk-rubl-asw1 нетегированными кадры 101-го влана и принимать их там в первый.
Дело в том, что, как мы уже упомянули выше, с точки зрения cisco с обеих сторон на коммутаторах должен быть настроен одинаковый номер влана, иначе начинаются проблемы с протоколом STP и в логах можно увидеть предупреждения о неверной настройке. Поэтому 101-й влан мы передаём на устройство обычным образом, кадры будут тегированными и соответственно, 101-й влан тоже необходимо создавать на msk-rubl-asw1.

Ещё раз хотим заметить, что при всём желании мы не сможем охватить все нюансы и тонкости, поэтому и не ставим перед собой такой задачи. Такие вещи, как принцип построения MAC-адреса, значения поля Ether Type или для чего нужен CRC в конце кадра, вам предстоит изучить самостоятельно.

Спасибо соавтору этого цикла, Максиму aka gluck.
За предоставление дополнительных материалов хочу поблагодарить Наташу Самойленко

Мы в телеграм:
Системный администратор
Книги для Системного Администратора
Чат системных администраторов