GeekBrains

Компьютерные сети

Сетевой уровень. Часть 1. Классовая адресация и статическая маршрутизация

IPv4-адреса, классовая адресация. Особые случаи адресации Зарезервированные и частные адреса. Маршрутизация на каждом компьютере. Работа в CLI. Настраиваем сетевые интерфейсы. Статическая маршрутизация. Формат IP-пакета. ICMP, traceroute

Введение

OSI/ISO	TCP/IP (DOD)
7. Прикладной уровень	4. Уровень приложений
6. Уровень представления	
5. Сеансовый уровень	
4. Транспортный уровень	3. Транспортный уровень
3. Сетевой уровень	2. Сетевой уровень
2. Канальный уровень	1. Уровень сетевых интерфейсов
1. Физический уровень	

На прошлом занятии мы рассмотрели канальный уровень и физическую адресацию, которая применяется, чтобы сообщения могли передаваться по локальной сети через среду и быть полученными адресатами, то есть чтобы довести информацию до узла-соседа, находящегося в том же сегменте сети. Задача передачи информации в другой сегмент на канальном уровне не стоит.

В отличие от адресации канального уровня, работающей только локально, задача адресации на сетевом уровне в общем случае (пока не берем частные и широковещательные адреса) — глобальная адресация. Физическая адресация, несмотря на то, что MAC-адреса теоретически уникальны во всем Интернете, не решает задачи глобальной адресации и не предназначена для этого. Задача MAC-адресов — уникально идентифицировать сетевой интерфейс в рамках широковещательного домена, т. е. необходимо различать только устройства, непосредственно видящие друг друга. Для задачи глобальной адресации MAC-адреса не используются, хотя и могут (с определенного рода надстройкой, префиксом сети — что, кстати, и использовалось в IPX и частично в IPv6).

Таким образом, на сетевом уровне у нас возникают следующие задачи:

1. Создать глобальную адресацию, позволяющую связать разные сети в единое адресное пространство таким образом, чтобы по глобальному идентификатору узла (хоста) можно было найти его местоположение (в какой сети он находится).
2. Обеспечить маршрутизацию (в нее входит вычисление принадлежности адреса соответствующей сети, L3-коммутация, поиск маршрута, передача сообщения в другую сеть через разнородное оборудование, а также передача информации о маршрутах между сетями).

Таким образом, сетевой уровень служит для передачи данных из сети в сеть, между сетями (хотя и передачу данных внутри сети он также обеспечивает).

Конечно, адресация сетевого уровня не всегда глобальна (существуют частные и изолированные сети и соответствующие локальные адреса, которые не используются в Интернете; кроме того, даже в одной локальной сети помимо MAC-адресов требуются IP-адреса. Даже внутри крупной локальной сети может потребоваться маршрутизация, если эта сеть состоит из нескольких подсетей, объединенных уже на сетевом уровне, с применением маршрутизаторов.

В отличие от канального уровня, сетевые протоколы не зависят от среды и не ориентируются на тот или иной ее тип. Фактически сетевой уровень решает задачи объединения сетей, поэтому в модели TCP/IP он называется межсетевым уровнем.

Данные на сетевом уровне вместе с заголовком, как правило, называют пакетом (реже используется термин «дейтаграмма», также применяемый для сообщений транспортного уровня без гарантированной доставки).

Таким образом, протоколы сетевого уровня выполняют следующее:

1. Инкапсулируют протоколы транспортного уровня и некоторые протоколы сетевого уровня (как правило, служебные, такие как ICMP, IGMP).
2. Осуществляют адресацию отправителя и получателя, позволяющую доставить пакеты между разными сетями (логическая адресация).
3. Преобразуют логические (сетевые) адреса в физические (как правило, MAC-адреса) для доставки через локальную сеть.
4. Осуществляют маршрутизацию и пересылку пакетов.
5. Передают данные на канальный уровень.
6. Сообщают об ошибках в случае невозможности доставки.

На этом уровне не контролируется корректность доставки — это задачи канального и транспортного уровней. С другой стороны, здесь генерируются сообщения об ошибках, если доставка не может быть выполнена.

IP-адрес

IP-адрес — адрес для логической адресации на сетевом уровне. IPv4-адреса пока используются гораздо шире, чем IPv6-адреса. О них дальше и пойдёт речь.

Зачем же нужен сетевой адрес?

Предположим, у нас есть компьютер, в нем несколько сетевых карт, у каждой из которых есть свой MAC-адрес. Нужно сделать машину доступной по сети.

Для каждого сетевого устройства задаем сетевой интерфейс, для которого мы назначаем IP-адрес. Например, 192.168.0.1.

IP-адрес может быть назначен статически, либо получен по протоколу DHCP (Dynamic Host Configuration Protocol — протокол динамической настройки хоста) через широковещательную рассылку. При этом в качестве адреса отправителя используется 0.0.0.0, а в качестве адреса получателя — broadcast-адрес 255.255.255.255. Таким образом, действие протокола DHCP ограничено широковещательным доменом.

Этот адрес будет адресом и отправителя (если мы отправляем сообщение), и получателя (если получаем). В любом случае будут указаны два адреса: отправителя и получателя.

Среди наиболее примечательных сетевых протоколов, позволяющих инкапсулировать данные вышестоящих уровней и идентифицировать отправителя и получателя с помощью логических адресов, можно назвать IPv4, IPX и IPv6. Часто под IP имеют в виду IPv4, но все большее распространение находит и IPv6. Каждый из этих протоколов имеет собственный формат адресов. В частности, IP-адреса (точнее IPv4-адреса) — это всем нам знакомые четыре октета, записываемые в десятичном виде и разделенные точками. Например, 192.168.0.1, 127.0.0.1, 8.8.8.8, 5.255.255.55. Так как на самом деле каждое число — один байт, оно может находиться в диапазоне от 0 до 255 (256 значений), то есть такие адреса, как 192.168.0.256, 10.0.0.300, 1000.0.0.1, невозможны. В памяти число хранится побайтово (побитово), и например, «в сыром виде» адрес 5.255.255.55 выглядит как 05FFFF37.

Особые случаи адресации

Среди IP-адресов есть несколько специальных.

0.0.0.0 используется в качестве адреса отправителя, пока IP-адрес не присвоен. В таблицах маршрутизации означает маршрут по умолчанию, в создании сокетов — что сокет прослушивает все сетевые интерфейсы.

255.255.255.255 — широковещательная рассылка, ограничена текущим широковещательным доменом (broadcast). Используется при поиске сервера, когда IP-адрес сервера неизвестен, в протоколах DHCP, PPPoE.

127.0.0.1 — текущий адрес машины. Используя его, можно связать два приложения на одной машине между собой (например, PHP и MySQL, nginx и Apache2).

Общеизвестные адреса

Некоторые IP-адреса, формально являющиеся самыми обычными, общеизвестны.

Таковыми являются приватные адреса (используемые только в локальных сетях) 192.168.0.1 и 192.168.1.1. Адрес 192.168.0.1 часто используют по умолчанию домашние роутеры, а 192.168.1.1 — DSL-модемы.

Есть и глобально маршрутизируемые адреса, которые знает каждый или почти каждый:

- 8.8.8.8 и 8.8.4.4 — адреса публичных DNS-серверов от Google,
- 1.1.1.1 и 1.0.0.1 — адреса публичных DNS-серверов от CloudFlare.

Несмотря на то, что, как правило, пользователь использует доменные имена сайтов, с помощью системы DNS они преобразуются, в IP-адреса, которые уже используются программами. Например, yandex.ru → 5.255.255.55. Для пользователя это происходит незаметно. Нередко добавляется ещё один этап: пользователь пишет в поисковой строке поисковый запрос (например, «одноклассники ру»), который передается поисковой системе (например, Яндексу), та находит адрес сайта (например, ok.ru), преобразующийся с помощью системы DNS в IP-адрес, например, 217.20.155.13.

Структура IPv4-адреса (классовая адресация)

Поговорим о структуре IP-адресов. При создании IPv4-адресов было решено использовать часть адреса для идентификации сети и часть — для хоста. При этом части оказались разной длины. Так была разработана классовая адресация. От 1 до 4 первых бит служили для обозначения типа сети, далее в зависимости от них следовал адрес сети нужной длины. Фактически биты входили как в сам адрес, так и в идентификатор сети.

Адреса сетей, начинающиеся с 0, относят к классу A, при этом номер сети занимает один байт, оставшиеся 3 байта используются для номера сетевого узла. Сети этого класса легко отличить по первому байту, который лежит в диапазоне от 1 до 126. (0 не используется, а 127 задействован для локальных адресов.)

Размер сети класса A вычисляется по формуле 2^24=16 777 216 узлов.

Адреса сетей, начинающиеся с 10, относят к классу B, при этом номер сети занимает два байта, оставшиеся 2 байта используются для номера сетевого узла. Сети класса B является средними, и число узлов в них вычисляется по формуле 2^16= 65 536 узлов.

Адреса сетей, начинающиеся со 110, относят к классу C, при этом номер сети занимает уже три байта, оставшийся байт используются для номера сетевого узла. Сети класса C является малыми и число узлов в них вычисляется по формуле 2^8= 256 узлов.

Таким образом, теоретически могут существовать 127 сетей (за вычетом 0.0.0.0), где для адреса сети использовался первый байт, и для хоста — остальные три октета. Это сети класса A.

Затем 16 384 сетей с хостами по два октета — сети класса B.

И еще чуть более 2 миллионов сетей класса C по 254 машины (+ адрес сети и широковещательный адрес — т. е. на хост нам оставался один байт).

Также есть специальные адреса класса D, которые принадлежат не совсем сетям, а мультикастным группам подписки. Мультикастная группа имеет только один адрес, но всё, что отправляется на него, доставляется для всех подписанных на него узлов. Используется в задачах маршрутизации и для онлайн-стриминга. Пример — IPTV. С помощью протокола IGMP клиенты вступают в группу, после чего начинают получать широковещательный трафик. В одну мультикастную группу могут входить клиенты из разных сетей. Фактический IP-адрес у подписчика не меняется, но он получает трафик, адресованный мультикастно. С помощью протокола IGMP происходит и отписка от вещания.

Сети класса E были зарезервированы для будущего использования, но в таком статусе и остались. Особое значение имеет адрес 255.255.255.255. Сообщение с таким адресом отправителя будет разослано всем узлам широковещательного домена.

В классовой адресации традиционно в качестве адреса сети использовался такой, в котором адрес хоста состоял из двоичных нулей. Для широковещательного адреса — адрес хоста, состоящий из двоичных единиц (т. е. в октетах было 255).

Например, для сети 10.0.0.0 класса A:

- 10.0.0.0 — адрес сети.
- 10.255.255.255 — широковещательный адрес (Broadcast).

Некоторые из этих адресов были зарезервированы для специальных целей и не маршрутизируются в Интернете.

Например, сеть класса A 127.0.0.0 полностью отдавалась под локальный адрес. Пакеты, направляемые на этот адрес, не уйдут за пределы локальной машины. Тем не менее такие адреса могут использоваться локальными службами для взаимодействия и отладки. Также часто используется адрес 127.0.0.1 в качестве адреса веб-интерфейса локальных служб, которые должны открываться только на этой же машине.

Сеть класса A 10.0.0.0 была выделена под локальные сети. Эти адреса не маршрутизируются в Интернете, а значит, любая организация может их использовать для локальной сети. Благодаря механизму NAT (а в прошлом — Proxy) эти адреса часто используются интернет-провайдерами.

16 сетей класса B — от 172.16.0.0 до 172.31.0.0 — и все сети класса C от 192.168.0.0 до 192.168.255.0 также были выделены для локального использования.

Чтобы выделить адрес сети из хоста, используется маска сети. Напомним, что состоит она из двух частей. Часть, идентифицирующая сеть, содержит двоичные единицы. Часть, выделенная под хост, содержит двоичные нули. Например, 255.0.0.0 — маска для сетей класса A.

Позже под частные сети был выделен еще один диапазон — 100.64.0.0/10. Что значит такая запись мы узнаем позднее, а пока можно сформулировать то же самое другими словами: это 127 сетей от 100.64.0.0 до 100.127.0.0, при этом всё равно это подсети сети класса A (пример бесклассовой адресации). Этот диапазон используется провайдерами и сетями крупных предприятий для Carrier-Grade NAT (CGN).

Произведя двоичное умножение побитово маски сети на IP-адрес, мы получим адрес сети.

Для сетей класса B — маска 255.255.0.0.

Для сетей класса C — маска 255.255.255.0.

При отправке сообщения происходит маршрутизация — поиск маршрута по таблице. Именно благодаря маске мы определяем, к какой сети принадлежит адрес и в какой сетевой интерфейс или на какой шлюз его отправлять.

Предположим, есть два маршрута: default gw 192.168.1.1 и 192.168.1.0 255.255.255.0 eth0. Если мы пропингуем 192.168.1.20, система попробует вычислить адрес сети.

Адрес: **192.168.1.**20.

Маска: **255.255.255.**0.

Сеть: **192.168.1**.0.

Значит, пакет будет направлен в сетевой интерфейс eth0.

Предположим теперь, что мы хотим пропинговать адрес 192.168.0.30. Точно так же вычисляется адрес сети.

Адрес: **192.168.0.**30.

Маска: **255.255.255.**0.

Сеть: **192.168.0**.0.

Это не та же сеть, которая доступна через eth0. Отдельный маршрут для этой сети неизвестен, значит, сообщение будет отправлено по маршруту по умолчанию.

Отправка работает через канальный уровень. Чтобы узнать MAC-адрес, нужно выполнить запрос ARP.

Запрос ARP рассылается бродкастно на MAC-адрес FF:FF:FF:FF:FF:FF. Как понять, какой MAC-адрес у хоста с IP-адресом 192.168.0.30? Если в сети присутствует такая машина, она отвечает, MAC-адрес

кешируется, и сообщение вкладывается в кадр, где указываются MAC-адреса получателя и отправителя.

Как быть, если у нас маршрут по умолчанию? Мы не сможем получить MAC-адрес для 192.168.100.1, а тем более для 5.255.255.55. Не сможем, но нам это и не нужно. ARP-запрос будет выяснять MAC-адрес шлюза. Поэтому в IP-заголовках будет IP-адрес получателя, а в заголовках кадра — MAC-адрес шлюза. Непосредственно IP-адрес шлюза на сетевом уровне никак не передается, он служит для преобразования в MAC-адрес.

Формат IPv4-пакета

- Версия (4 бита) — служит для определения IPv4, IPv5(ST) или IPv6. IPv4 соответствует значение 4.
- Длина заголовка (4 бита) — в 32-битных словах (на картинке эти слова пронумерованы в левой части — например, если нет опций, то длина будет 5). Т. е., чтобы получить длину заголовка в битах, необходимо умножить значение на 32, в байтах — на 4.

Обратите внимание: первый байт IPv4-пакета часто имеет значение 45 (проверьте в Wireshark).

Далее следуют:

- Тип обслуживания (ToS — type of service).
- Общая длина — длина пакета в байтах (заголовка и данных вместе).

Второе слово IPv4-пакета полностью управляет фрагментированием пакетов.

- Идентификатор пакета — служит для идентификации пакетов, главным образом при фрагментации (например, на маршрутизаторе при необходимости передать через канал с меньшим MTU). В случае фрагментации фрагменты пакета обладают одним и тем же идентификатором.
- Флаги — также служат для управления фрагментацией (или ее запретом). Если установлен флаг «не фрагментировать», то когда понадобится отправить пакет через канал с меньшим

MTU, пакет будет отброшен, а в ответ направлено ICMP-сообщение: требуется фрагментация, но фрагментация запрещена. Если флаг сброшен, пакет будет фрагментирован.

- Смещение фрагмента — служит для восстановления исходного пакета (в нефрагментированном виде). Если пакет фрагментирован, то каждый фрагмент обладает одним и тем же идентификатором, но разными смещениями, которые и позволят собрать пакет обратно. Если фрагмент не последний, устанавливается соответствующий флаг.

Третье слово содержит два очень важных параметра и один, не используемый на практике:

- Время жизни — TTL (time to live) — максимальное число хопов (прыжков). При каждом прохождении маршрутизатора уменьшается на единицу. Если TTL=0, пакет отбрасывается. Это важный параметр. Изначально планировалось, что оно должно изменяться в секундах, но сделать это оказалось сложно, поэтому уменьшение TTL реализовали как простое уменьшение значения счетчика. Как оказалось, нет ничего более постоянного, чем временное. В IPv6 аналогичное поле называется «Число прыжков». К слову сказать, TTL в секундах действительно применяется в DNS и служит для указания времени кеширования записи в секундах, а учитывать время прохождения пакета в сети умеет протокол NTP (Network Time Protocol).
- Контрольная сумма заголовка — только для заголовка, не для данных! Пересчитывается при прохождении каждого маршрутизатора, так как значение TTL меняется.
- Протокол — тип протокола верхнего уровня (ICMP, UDP, TCP и другие, которые инкапсулируются в IP).

Следующие два слова — адреса отправителя и получателя в формате IPv4 (по 8 октетов).

Опции — служат для расширения и настроек. Может содержать до 10 слов (число ограничено максимальным значением длины заголовка в словах, так как 4 бита позволяют указать до 16 значений, учитывая слова самого заголовка).

Обратите внимание, что маска сети в протоколе не передается. Маска является принадлежностью непосредственно сетевого интерфейса. Поэтому хосты с IP-адресами в разных сетях, но входящими в диапазон обеих подсетей, будут пинговать друг друга — например, 172.17.10.1/255.255.255.0 и 172.17.10.100/255.255.255.128. Проверьте!

Разрешение IPv4 в MAC-адреса

IP-адреса нужны при маршрутизации (3 уровень OSI). При каждом прохождении маршрутизатора, маршрутизатор анализирует IP-адрес получателя и интерфейс, в который необходимо отправить пакет дальше, чтобы он достиг получателя.

Но в рамках сегмента используется физическая адресация сетевыми картами, чтобы отбрасывать кадры, которые в поле получателя имеют MAC-адрес, отличный от адреса карты. Коммутаторы ведут

таблицы соответствия своих портов и MAC-адресов. Поэтому каждый хост ведет таблицы ARP (address resolution protocol) — соответствие IP адресов и MAC-адресов.

Посмотрите таблицу на вашем компьютере.

```
arp –a
```

При этом записи обозначаются как статические и динамические.

Статические: вы можете принудительно сопоставить MAC-адрес сетевой карты вашего шлюза с его IP-адресом (arp -s).

Динамические записи определяются с помощью протокола ARP. Если MAC-адрес неизвестен, то отправляется ARP-запрос (вкладывается в кадр, занимает промежуточное положение между канальным и сетевым уровнем) с указанием IP-адресов отправителя и получателя и их MAC-адресов. В качестве MAC-адреса получателя выступает широковещательный адрес FF:FF:FF:FF:FF:FF.

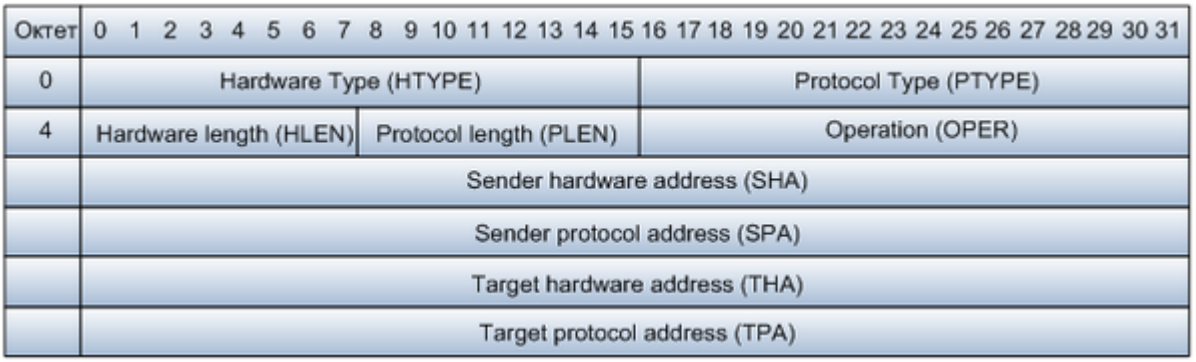
Для бродкаст-запросов ARP не используется, широковещательный адрес FF:FF:FF:FF:FF:FF используется сразу в качестве MAC-адреса назначения в заголовках кадра, несущего броадкаст-пакет. При этом узлы, получив сообщение, будут отвечать на юникаст-адрес. Они будут использовать ARP-запросы.

Соответственно, коммутатор направляет запрос на все порты, а нужная машина (искомый получатель либо шлюз) направляет ответ с указанием MAC-адреса. Результат кешируется в arp-таблицу, и пакеты на нужный IP-адрес уже инкапсулируются в кадры с MAC-адресом получателя, взятым из таблицы.

Протокол уязвим, потому как можно подделать ARP-ответ. Атака называется ARP-spoofing — это одна из реализаций атаки типа «человек посередине».

Для защиты от нее следует либо использовать статические ARP или Access List, либо изолировать широковещательные домены, т. е. использовать на канальном уровне VLAN либо на сетевом — тоннели.

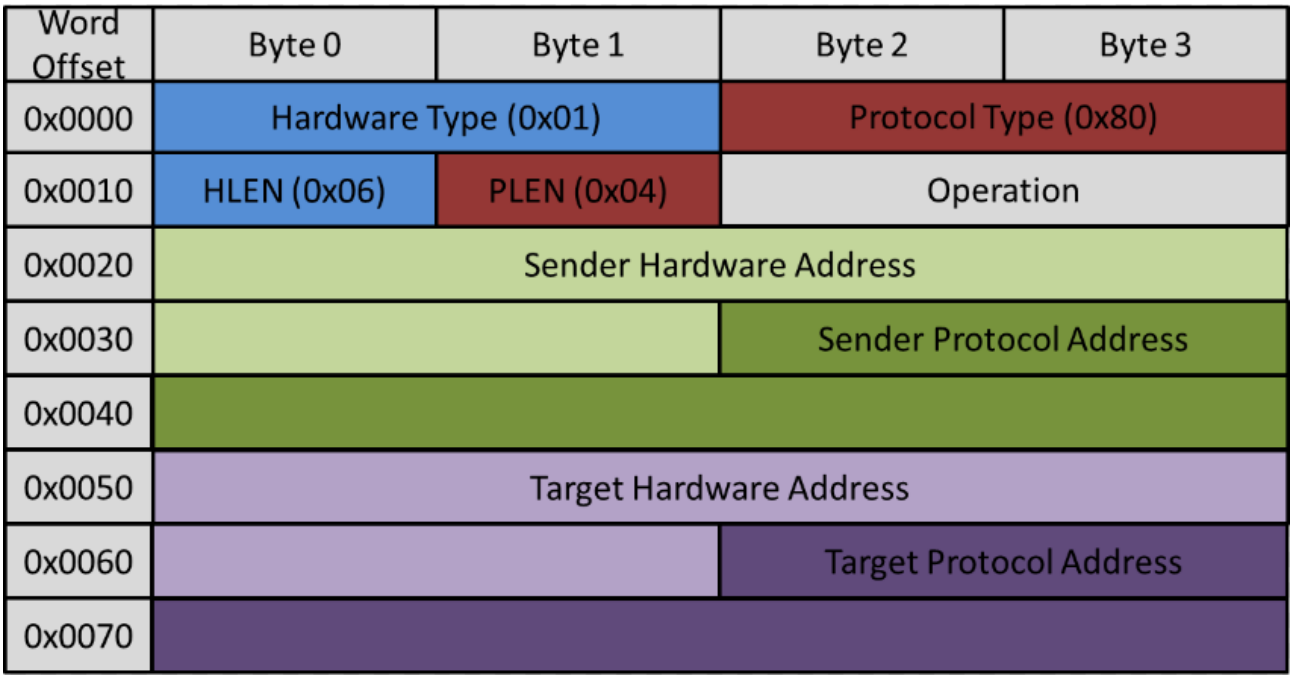
Формат ARP-запроса



ARP-запросы и ответы сразу вкладываются в поле «Данные» Ethernet-кадра.

1. HTYPE — тип протокола канального уровня. Например, Ethernet — 0x0001.
2. PTYPE — тип протокола сетевого уровня. Например, IPv4 — 0x0800.
3. HLEN — длина физического адреса. Например, для Ethernet — 6 байт.
4. PLEN — длина сетевого адреса. Например, для IPv4 — 4 байта.
5. OPER — тип операции.
    a. 1 — ARP-запрос.
    b. 2 — ARP-ответ.
    c. 3 — ReverseARP(RARP)-запрос.
    d. 4 — ReverseARP(RARP)-ответ.
6. SHA — физический адрес отправителя. Как правило, MAC-адрес отправителя.
7. SPA — сетевой адрес отправителя. Как правило, IP-адрес отправителя.
8. THA — физический адрес получателя. Если MAC-адрес неизвестен, устанавливается в 00:00:00:00:00:00 (в то время, как аналогичное поле Ethernet-кадра будет иметь значение FF:FF:FF:FF:FF:FF).
9. TPA — сетевой адрес получателя. Как правило, IP-адрес получателя.

Так как длина MAC-адреса и IP-адреса не кратны, на практике формат ARP-пакета выглядит следующим образом.



RARP, BOOTP, DHCP

Стоит отметить, что существует целое семейство ARP-протоколов. ARP — наиболее известный и часто используемый протокол. Для решения обратной задачи (известен свой MAC-адрес, не известен свой IP-адрес) ранее применялся протокол RARP (Reverse ARP). Формат его пакета совпадает с

форматом пакета ARP, но используются другие OPER-коды (3 и 4), а самое главное — в поле «Тип пакета» канального уровня у RARP другой код протокола. То есть, несмотря на сходства, это разные протоколы, обрабатываемые разными механизмами. Тем не менее, если, например, сервис, отвечающий за RARP, получит ARP-запрос, он передаст его на обработку сервису, отвечающему за ARP. В случае использования RARP, в отличие от ARP, требовался выделенный сервер, содержащий таблицу соответствий MAC-адресов и IP-адресов. Для обращения к серверу использовался broadcast-адрес. Проблемой такого подхода оказалась необходимость использования отдельного RARP-сервера для каждого широковещательного домена, потому на замену RARP был создан BOOTP — Bootstrap Protocol. Он обладал многими новыми возможностями, позволял назначить не только IP-адрес, но и другие известные нам параметры, такие как маска сети, адрес маршрутизатора по умолчанию, DNS-сервер и даже указать для бездисковой машины TFTP (Trivial File Transfer Protocol) сервер с образом ОС для распаковки RAM. Протокол BOOTP инкапсулировался в UDP-дейтаграммы и мог маршрутизироваться. Но и у у него были проблемы. По-прежнему при появлении каждой новой машины необходимо было вручную настраивать BOOTP-сервер. Поэтому следующим развитием BOOTP-протокола стала его новая версия, получившая и новое название — Dynamic Host Configuration Protocol — протокол динамической настройки хоста, который позволяет динамически настраивать подключающиеся машины и широко используется в настоящее время.

ICMP (Internet Control Message Protocol)

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Type								Code								Checksum															
4	32	Rest of Header																															

Несмотря на то, что ICMP-пакеты упаковываются в поле данных IP-пакетов, ICMP относится к сетевому уровню и фактически является частью механизма IP.

В случае, если пакет не удается доставить, узел генерирует ICMP-сообщение в адрес отправителя (нет маршрута, требуется фрагментация пакета, но она запрещена флагом, хост недоступен и т. д.).

Некоторые типы и коды ICMP-сообщений

Тип=8. Код=0. Эхо-запрос.

Тип=0. Код=0. Эхо-ответ.

Поле данных содержит случайные тестовые данные (теоретически данные могут быть и осмысленными, что использовалось в свое время для организации так называемых ICMP-тоннелей).

Тип=3. Код=0. Сеть недостижима.

Тип=3. Код=1. Узел недостижим.

Тип=3. Код=3. Порт недостижим.

Тип=3. Код=4. Необходима фрагментация, но установлен флаг ее запрета (DF).

Тип=11. Код 0. TTL истекло.

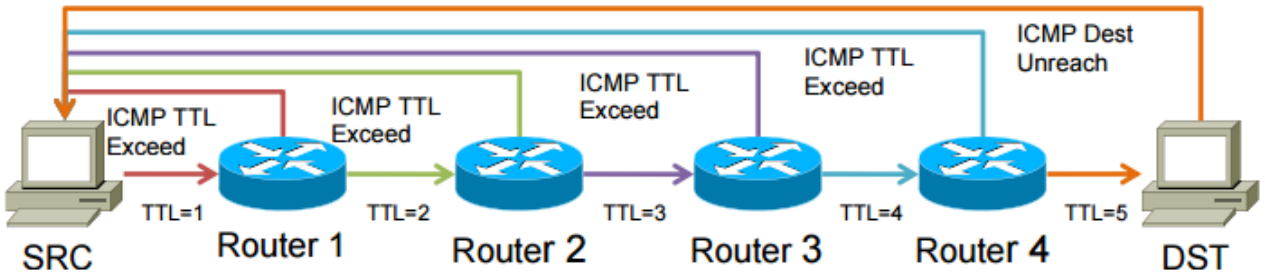
Для сообщений с типом 3 и 11 в поле данных возвращается заголовок IP-пакета и начало дейтаграммы.

Ping и tracert/traceroute

Особый случай — echo-запрос. Он используется для диагностики в утилитах ping и в tracert (**traceroute -I**). Команда **ping** формирует echo-запрос, содержащий случайное сообщение. Узел, получив echo-запрос, возвращает то же сообщение обратно, отметив его как echo-ответ. Таким образом, мы получаем сам факт того, что узел ответил, а кроме того, информацию о времени получения ответа, о дублированиях и потерях.

Traceroute в Linux по умолчанию использует UDP-протокол (но с помощью ключа **-I** можно использовать и ICMP).

Рассмотрим, как работает tracert с помощью ICMP (принцип UDP-трассировки аналогичен).



Клиент посылает ICMP-сообщение на узел назначения точно так же, как и при **ping**, но ему задается TTL=1. Первый маршрутизатор, получив сообщение, отбрасывает его и в ответ генерирует ICMP-сообщение об ошибке (TTL истекло) с указанием собственного IP-адреса в качестве адреса отправителя (его мы и видим в выдаче).

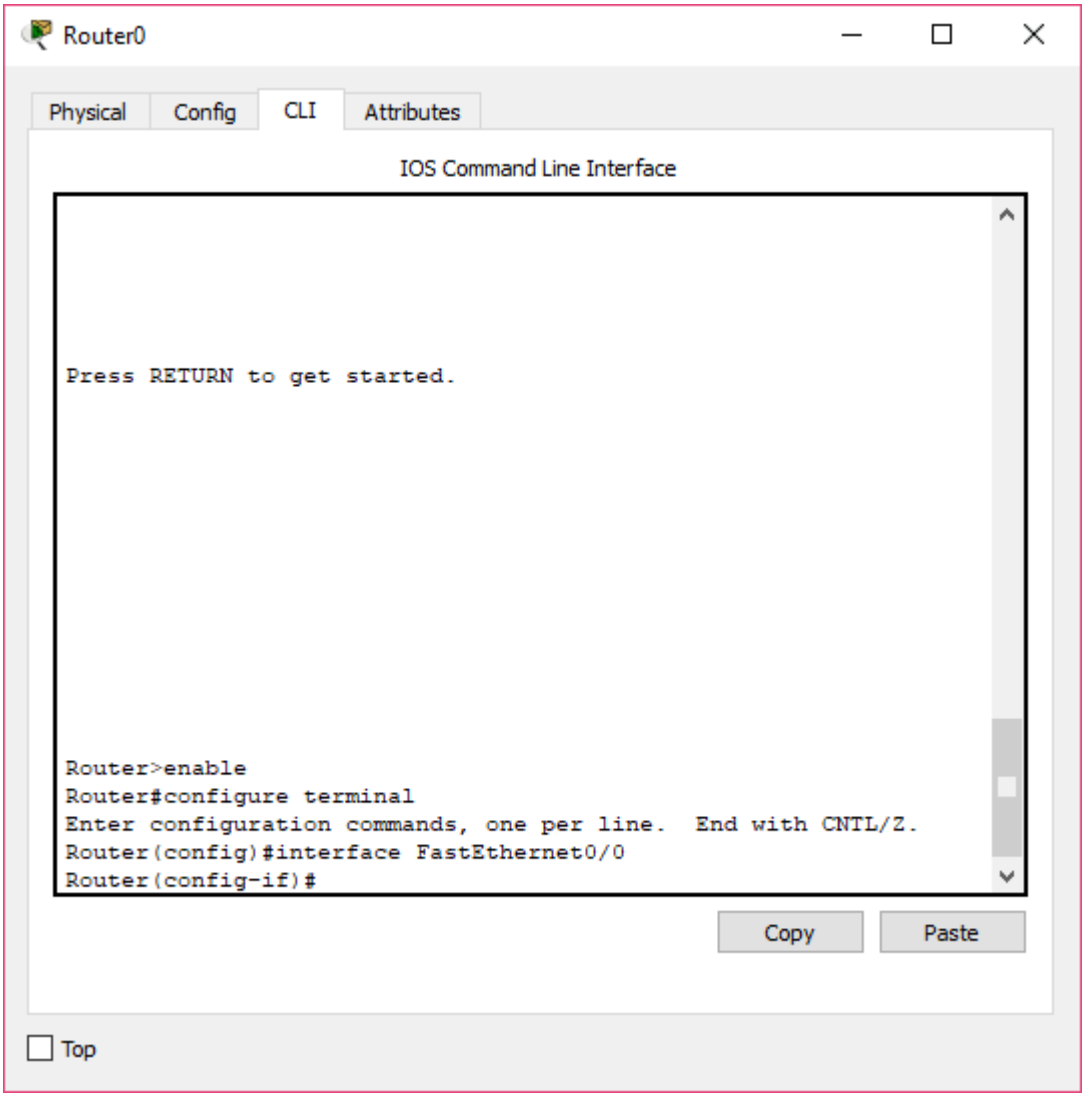
Сначала высылаются 3 сообщения с TTL=1, затем с TTL=2, и мы получаем сообщение от второго прыжка. И так далее, пока не получим сообщения от всех маршрутизаторов.

Работа в консоли CLI

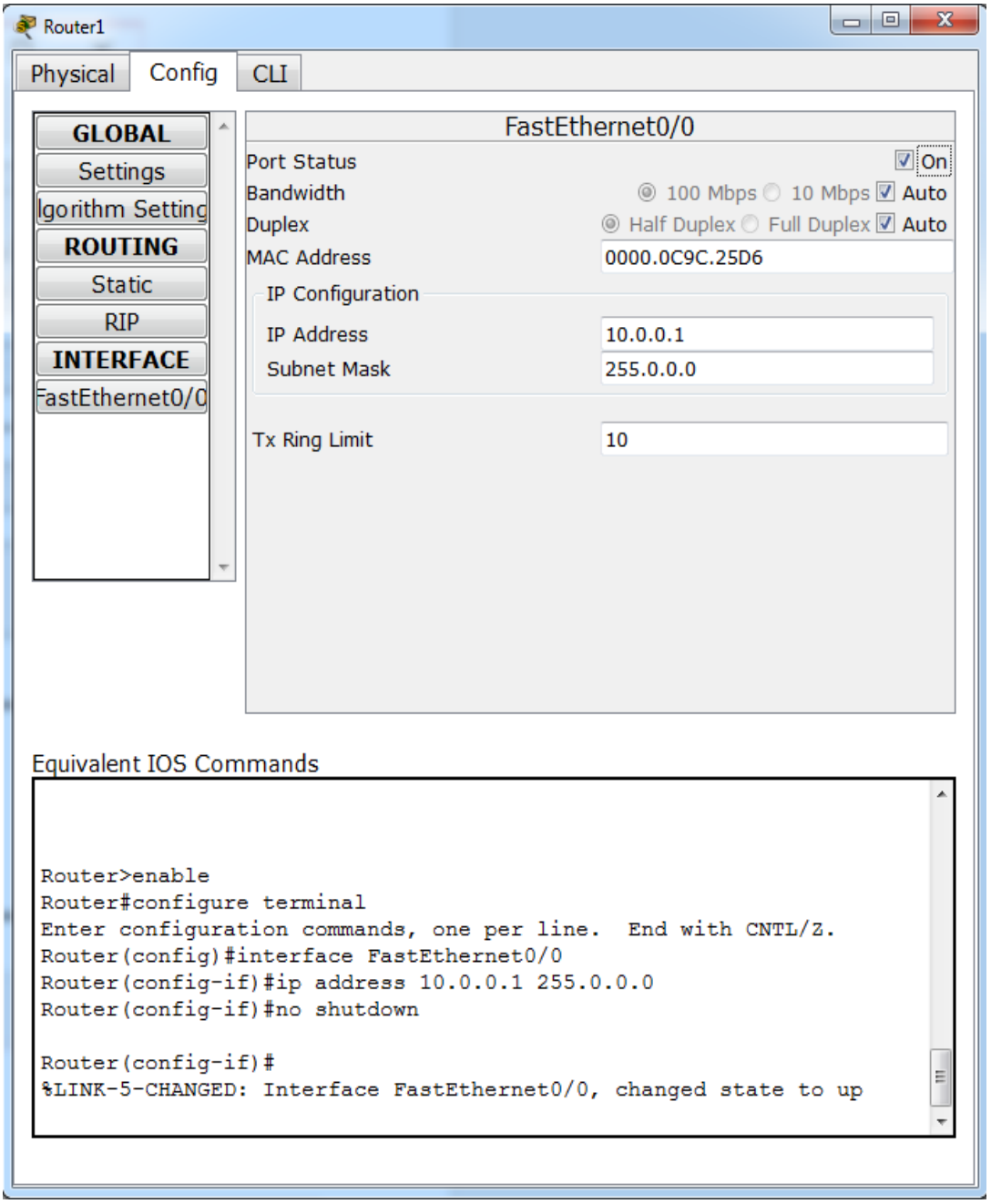
Оборудование Cisco работает под управлением собственной операционной системы — **Cisco IOS** (от англ. Internetwork Operating System — межсетевая операционная система).

Настройка оборудования выполняется через **CLI** (Command-Line Interface — интерфейс командной строки).

В **Cisco Packet Tracer** (далее сокращенно **PT**) консоль доступна через вкладку **CLI.**



Несмотря на то, что в **PT** есть подобие графического интерфейса, он сам по себе не слишком функционален. Нажимая на кнопки и вводя значения, мы на самом деле управляем устройством через **CLI**. Обратите внимание на то, что происходит в **CLI,** когда мы что-то делаем, например, с маршрутизатором. В примере мы настроили на интерфейсе *FastEthernet0/0* IP адрес **10.0.0.1** и маску подсети **255.0.0.0**, а также **включили** сам интерфейс.



Давайте разберем, что именно **PT** написал в **CLI**.

Переход в привилегированный режим:

```
Router>enable
```

Вход в режим глобальной конфигурации:

```
Router#configure terminal
```

Вход в раздел конфигурации интерфейса *FastEthernet0/0*:

```
Router(config)interface FastEthernet0/0
```

Установка IP-адреса и маски подсети на интерфейсе.

```
Router(config-if)#ip address 10.0.0.1 255.0.0.0
```

Включение сетевого интерфейса.

```
Router(config-if)#no shutdown
```

При конфигурировании оборудования напрямую через CLI необходимо было бы выполнять те же консольные команды.

Основные концепции Cisco CLI

Режимы работы командной строки

Устройства Cisco имеют несколько режимов командной строки:

1. Пользовательский режим (user mode).
2. Привилегированный режим (privileged mode).
3. Режим глобальной конфигурации (global configuration mode).
4. Режим специфической конфигурации.

Понять, в каком режиме мы находимся, очень просто. Нужно посмотреть на приглашение в командной строке. Оно будет иметь следующий вид:

1. Для пользовательского режима — ***Router>.***
2. Для привилегированного режима — ***Router#.***
3. Для режима глобальной конфигурации — ***Router(config)#.***
4. Для режимов специфической конфигурации — ***Router(config-*)#***, где на месте звездочки находится название подрежима. Например, ***Router(config-if)#*** — режим настройки сетевого интерфейса.

Вместо слова Router пишется имя устройства. По умолчанию маршрутизаторы имеют имя Router, а коммутаторы — Switch, но обычно при конфигурировании эти имена меняют на более конкретные.

Пользовательский режим

В этот режим мы попадаем изначально. Здесь доступен только ограниченный перечень команд, выполнение которых не должно навредить функционированию устройства. Например, из этого режима можно посмотреть версию операционной системы командой ***show version*** или выполнить команду ***ping***.

Привилегированный режим

Для перехода в этот режим необходимо из пользовательского режима выполнить команду ***enable*** и в случае необходимости ввести пароль. После перехода нам доступен полный перечень команд и возможность перехода в режим конфигурации без пароля. Таким образом, зная пароль на вход и на переход в привилегированный режим, человек имеет полный доступ к устройству. Для перехода обратно в пользовательский режим используется команда ***disable***.

Режим глобальной конфигурации

Этот режим позволяет вносить изменения в конфигурацию устройства. Для входа в него необходимо из привилегированного режима выполнить команду ***configure terminal***. Ввод паролей в данном случае не потребуется. Быстрый выход из режима глобальной конфигурации выполняется командой ***end***.

Режимы специфической конфигурации

Они являются подрежимами режима глобальной конфигурации. Например, введя в режиме глобальной конфигурации команду ***interface FastEthernet 0/0,*** мы перейдем в подрежим настройки соответствующего интерфейса ***(config-if)***. Множество режимов специфической конфигурации соответствует множеству разных ветвей глобальной конфигурации. Выход на уровень выше выполняется командой ***exit***.

Хранение конфигурации оборудования

В устройствах Cisco имеется по меньшей мере 2 конфигурации:

1. **Running-configuration** — конфигурация, загруженная в данный момент в оперативную память устройства. Когда вы вносите изменения в оборудование, изменяется именно она.
2. **Startup-configuration** — конфигурация, которая хранится в энергонезависимой памяти устройства и будет прочитана и установлена в **running-configuration** при включении устройства.

Важно! Running-configuration НЕ сохраняется автоматически и в случае перезагрузки устройства теряется.

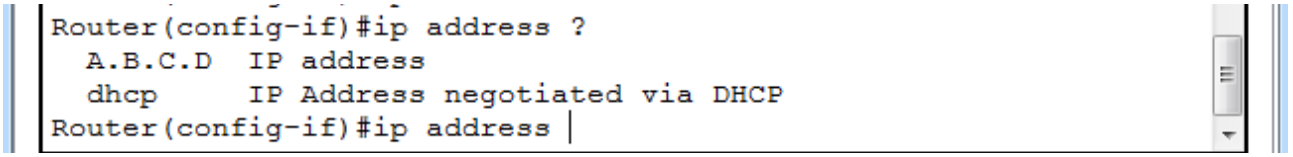
Основные команды для работы с конфигурациями:

- ***show running-config*** отображает текущую рабочую конфигурацию;
- ***show startup-config*** отображает текущую стартовую конфигурацию;
- ***write*** выполняет запись **running-configuration** в **startup-configuration;**
- ***erase startup-config*** удаляет **startup-config**; если в таком состоянии перезагрузить устройство, оно загрузится с настройками по умолчанию.

Общие методы работы с CLI

Получение справки

Знак вопроса можно использовать почти в любой момент, чтобы **получить справку** о возможных командах:



Автозавершение команд

Нажатие клавиши Tab выполняет автозавершение текущей написанной команды, если не возникает неоднозначности.

Выполнение команд из режима конфигурации

Если на устройстве работает Cisco IOS 12.2(8) или выше, вы имеете возможность использовать **do** для запуска привилегированных команд из конфигурационного режима. Другими словами, вы можете выполнить команду **show** или любую другую во время конфигурирования устройства.

Для выполнения действия необходимо добавить **do** перед необходимой командой.

Например:

```
router(config)# do show interface f0/0
```

или:

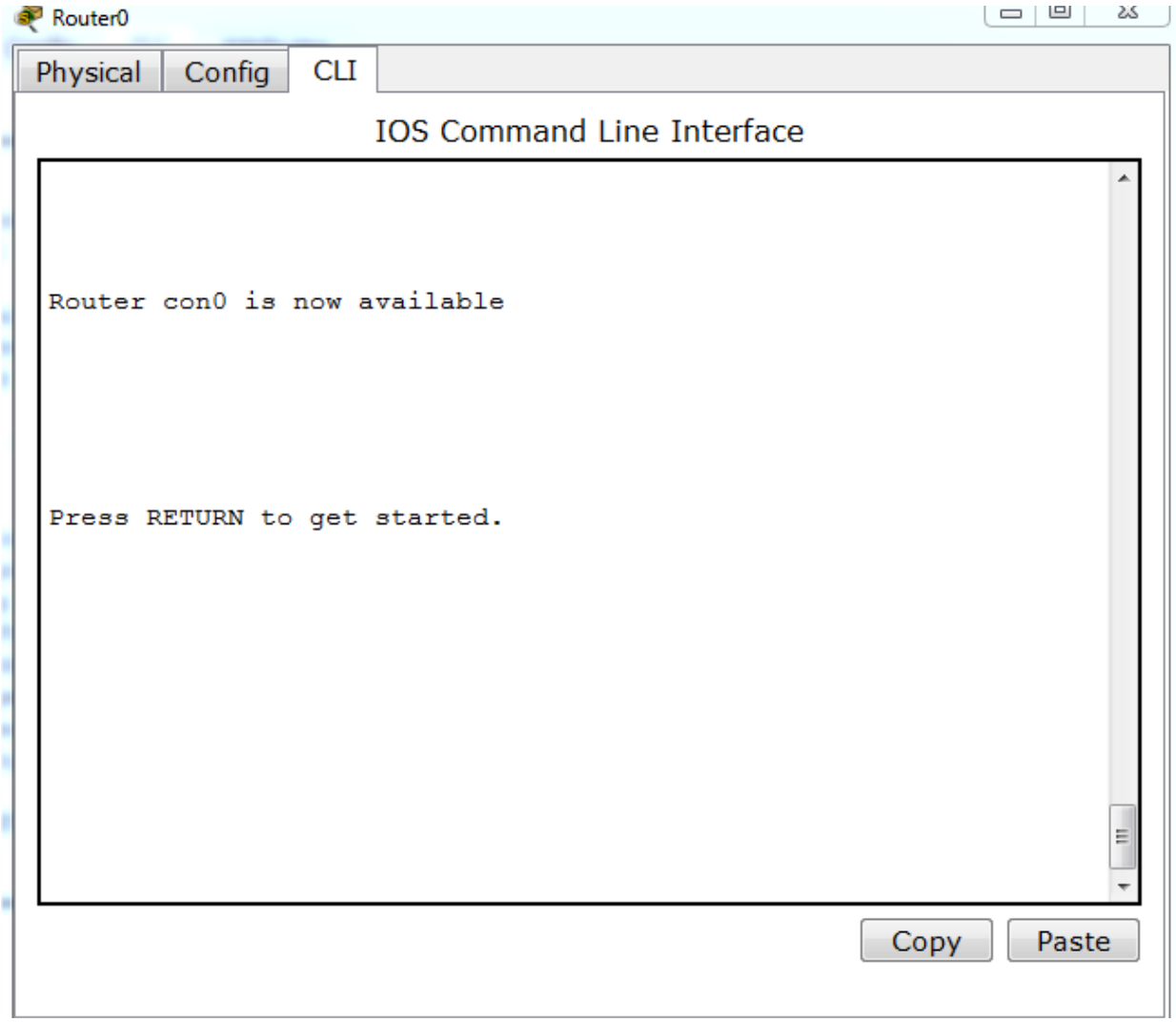
```
switch(config-if)# do show run
```

Так же можно использовать команды **ping**, **debug** и пр.

Сокращение команд

Все команды можно сокращать, если это не вызывает неоднозначности. Пример: **configure terminal** сокращается до **conf t**.

Пример ручной конфигурации сетевого интерфейса с помощью CLI




```
Router>enable
Router#configure terminal
Router(config)#interface fa1/0
```

Нажатие клавиши Tab приводит к автозавершению команды, если это возможно:

```
Router(config-if)#ip addr<tab>
```

В результате:

```
Router(config-if)ip address
```

Назначаем IP-адрес и маску подсети:

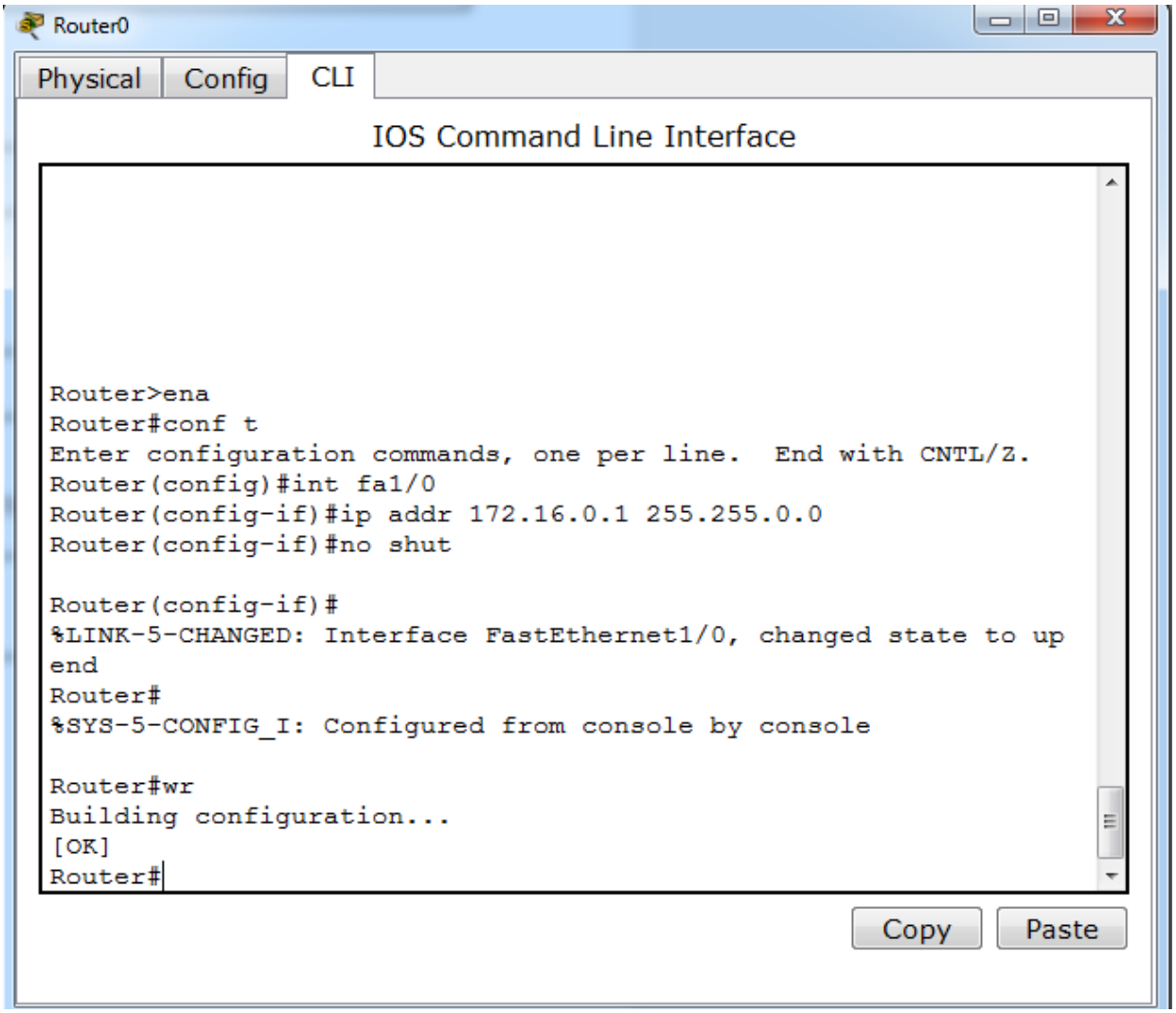
```
Router(config-if)ip address 172.16.0.1 255.255.0.0
```

Включим сетевой интерфейс (по умолчанию он выключен):

```
Router(config-if)no shutdown
```

Выходим из режима конфигурации и сохраняем текущий конфиг в стартовый:

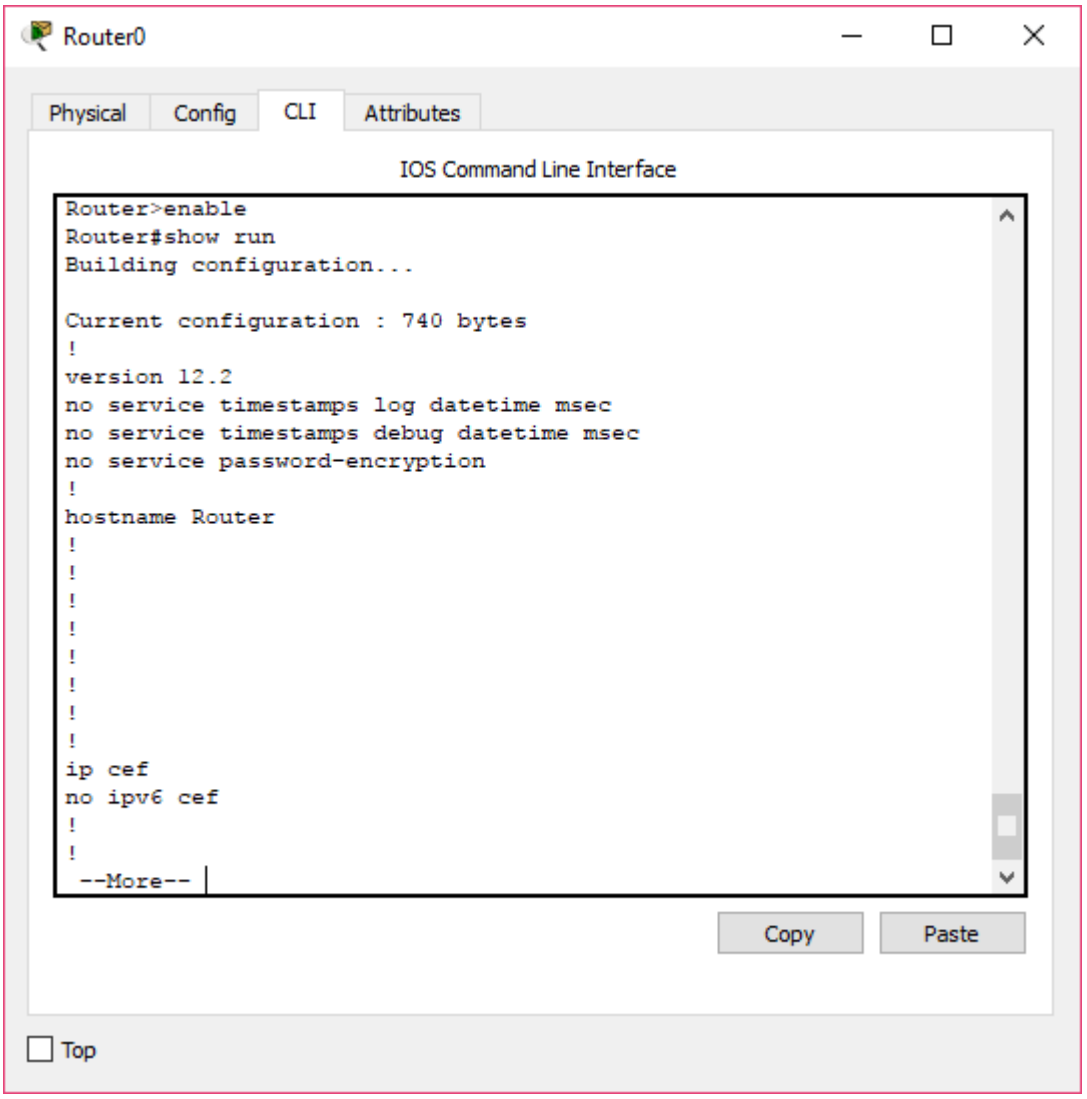
```
Router(config-if)end
Router#write
```



Мы сделали всё то же самое, что умеет делать графический интерфейс, и немножко больше (сохранили конфигурацию).

Для отображения текущей рабочей конфигурации введите:

```
Router#show run
```



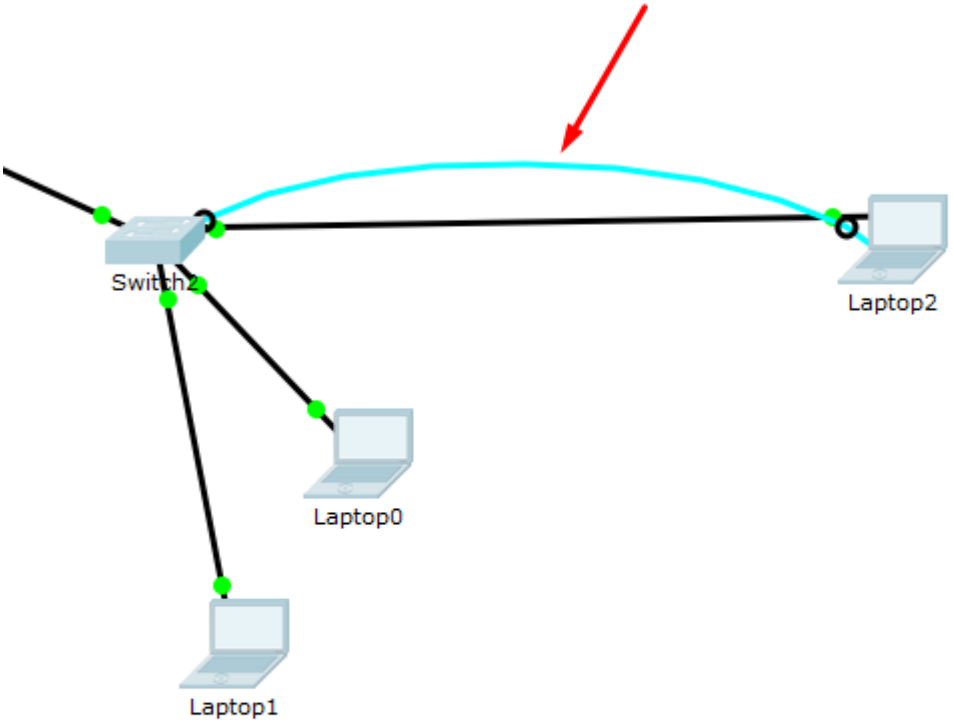
Попробуйте самостоятельно:

```
Router#show ?
```

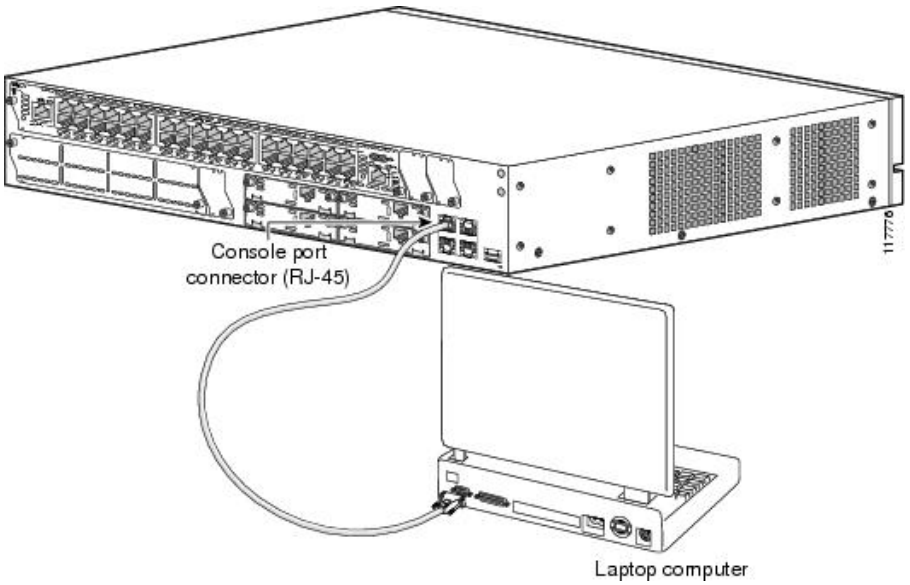
Изучите предложенные команды.

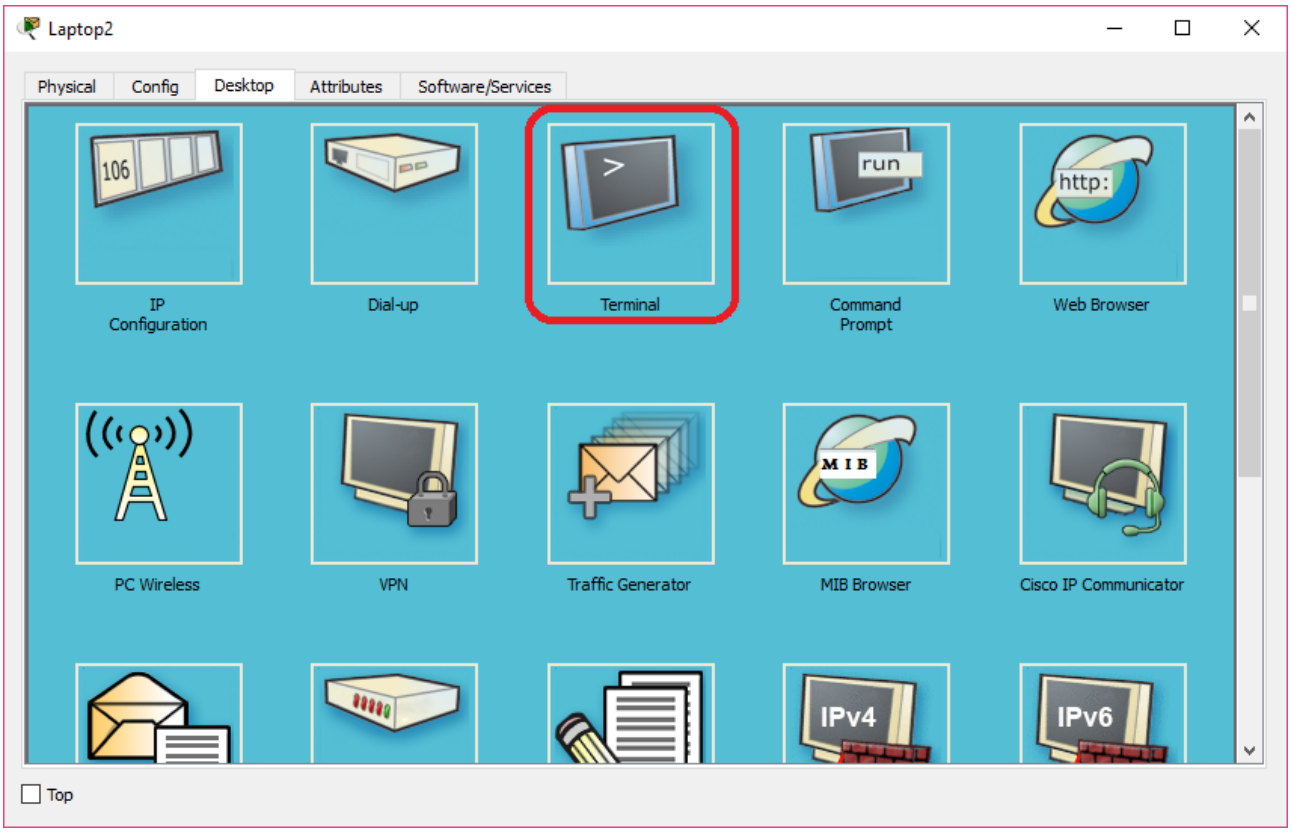
Настройка удаленного доступа к коммутатору/маршрутизатору через telnet

Настроим коммутатор для доступа через telnet, подключив синий консольный кабель (это мы в Cisco Packet Tracer видим терминал, а на практике надо подключить терминал через консольный кабель). После настроек можно будет подключаться к устройству по сети.

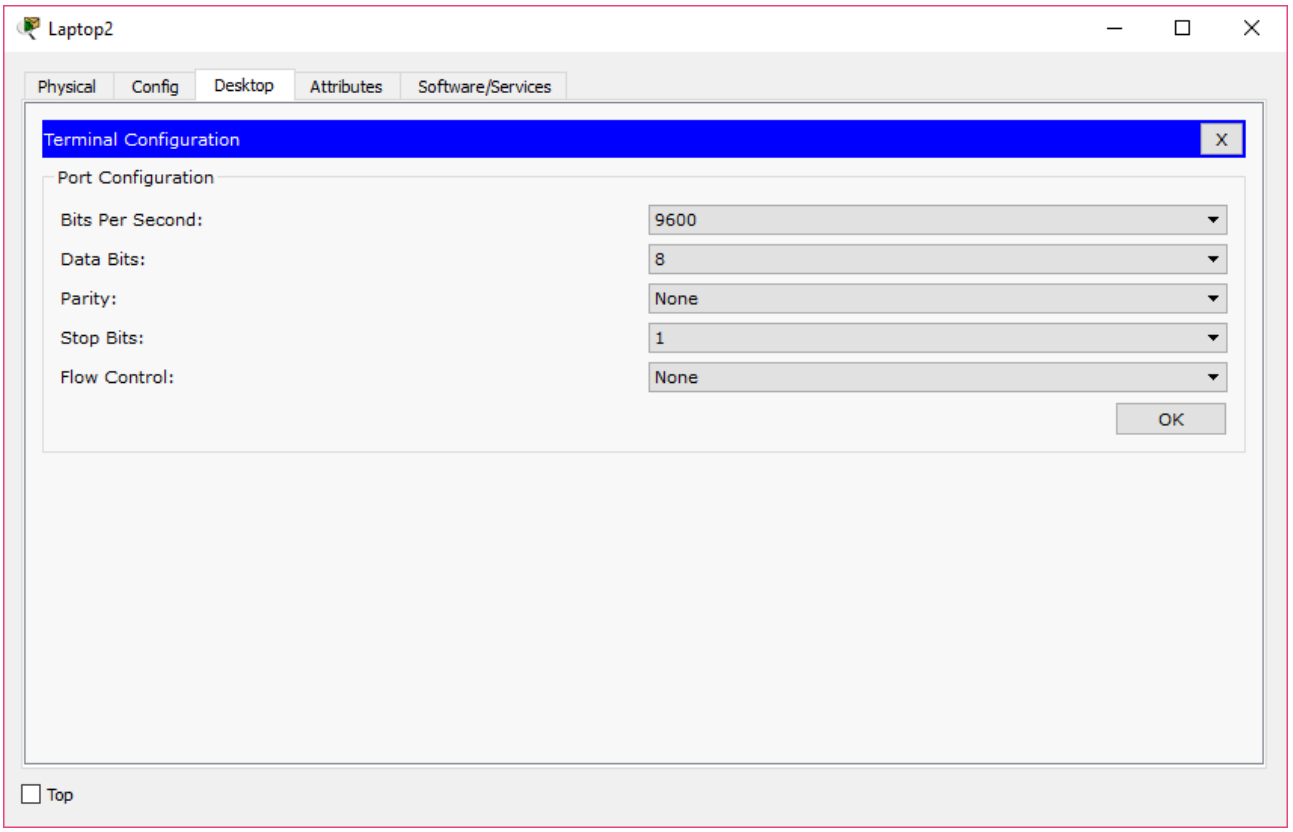


Нужно включить консольный кабель в консольный порт коммутатора и в порт RS-232 (COM-порт, стандартный последовательный порт для работы с терминалом) компьютера.



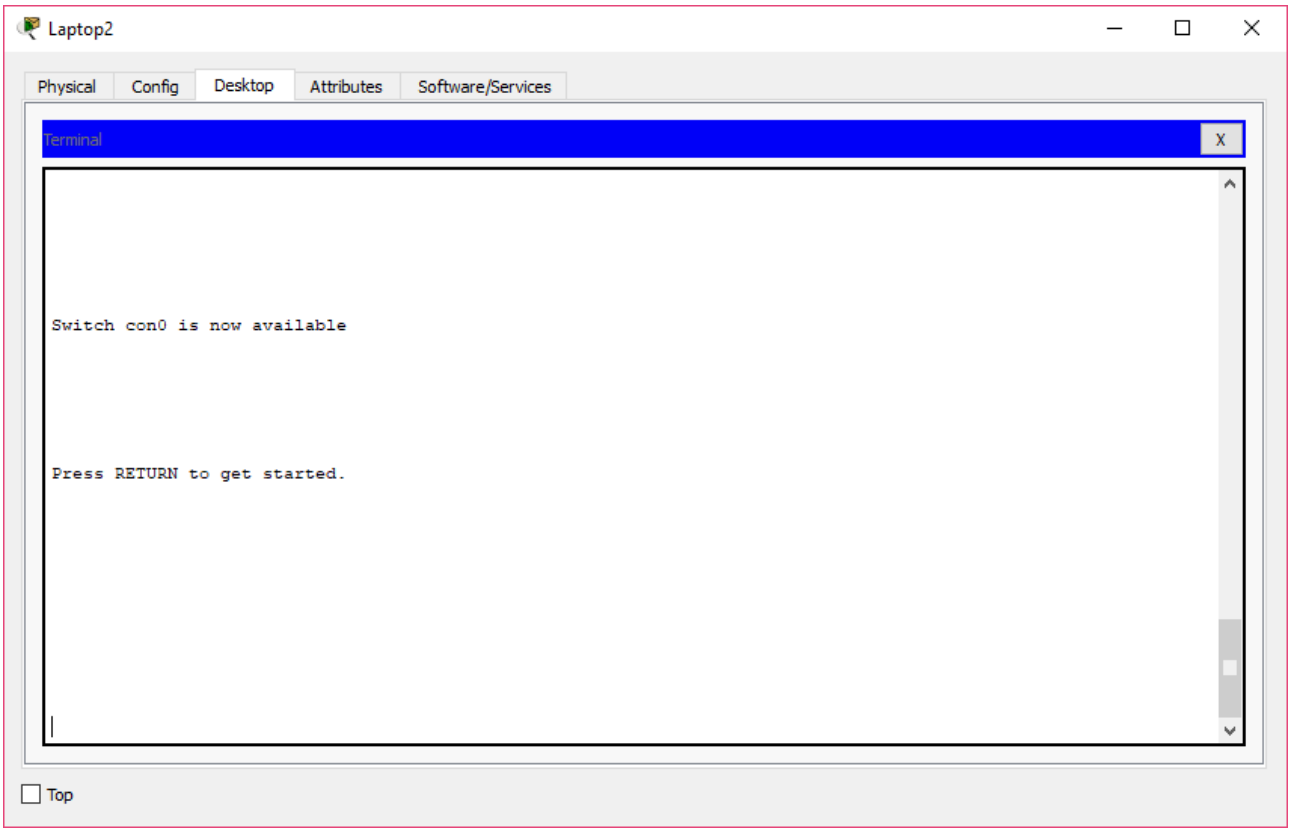


Кликаем терминал:



Стандартные настройки: 9600-8-N-1-N.

Подключаемся:



Мы видим ту же самую консоль, что и через Cisco Packet Tracer, только теперь ближе к реальности.

Команды можно сокращать:

```
Switch>ena
Switch#conf t
```

Задаем пароль для привилегированного режима (enable):

```
Switch(config)#ena pass qwr
```

Настроим пять терминальных линий:

```
Switch(config)#line vty 0 4
```

Зададим пароль:

```
Switch(config-line)passint asd
```

Настроим сетевой интерфейс для доступа по сети. Он будет находиться в первом vlan и иметь IP-адрес 10.0.0.1 с маской 255.0.0.0.

```
Router>ena
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa1/0
Router(config-if)#ip addr 172.16.0.1 255.255.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up
end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#wr
Building configuration...
```

```
Switch(config-line)int vlan 1
Switch(config-if)ip addr 10.0.0.1 255.0.wr0.0
```

Включим интерфейс:

```
Switch(config-if)no shut
```

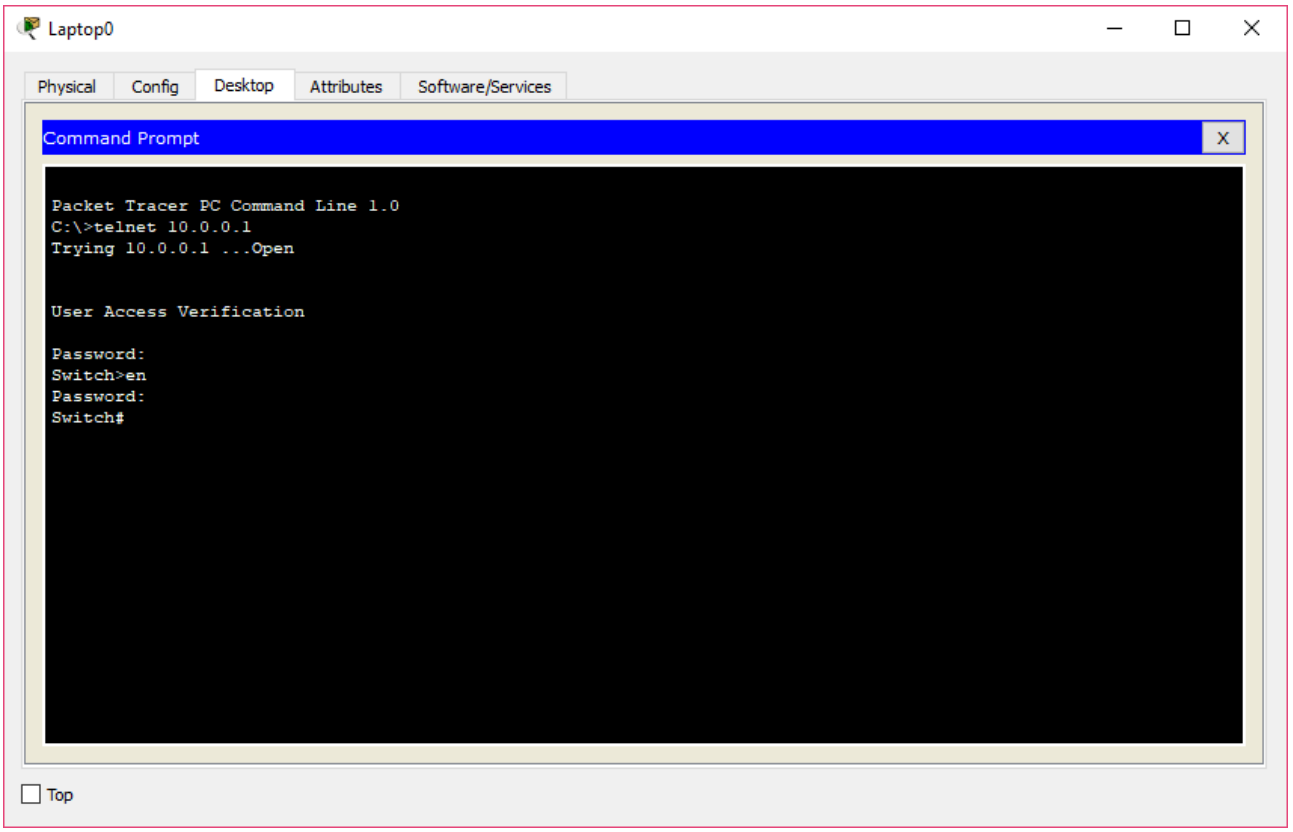
Выходим из режима конфигурации:

```
Switch#end
```

Сохраняем текущую конфигурацию:

```
Switch#wr
```

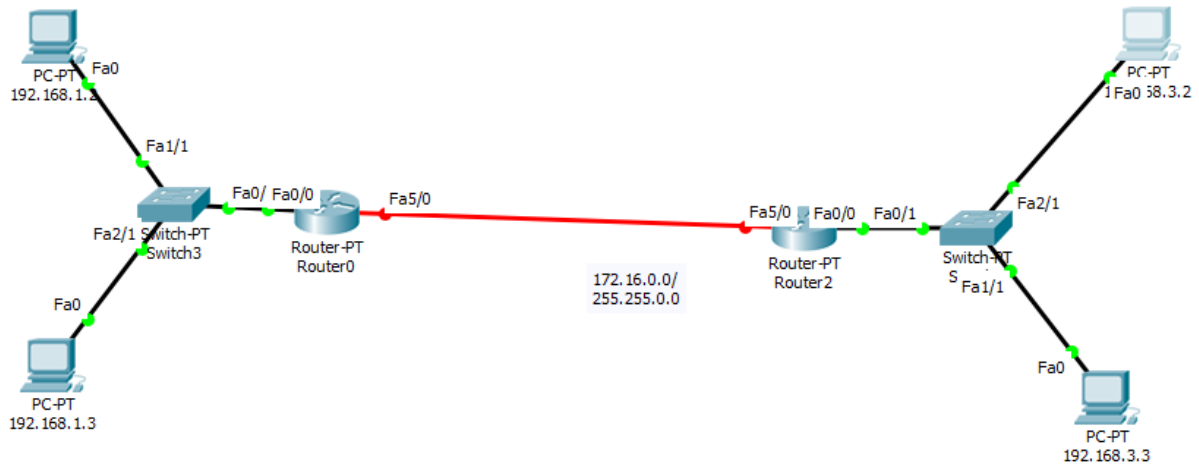
Теперь мы можем с любого другого компьютера в сети 10.0.0.0 (компьютер подключен к коммутатору и имеет IP-адрес, например 10.0.0.200, и маску сети 255.0.0.0) подключиться к устройству.



Вводим пароль, переходим в enable, снова вводим пароль. Можно настраивать.

Настройка статической маршрутизации в CLI

Рассмотрим вот такую схему:



Первое, что нужно настроить, — это сетевые интерфейсы на роутерах

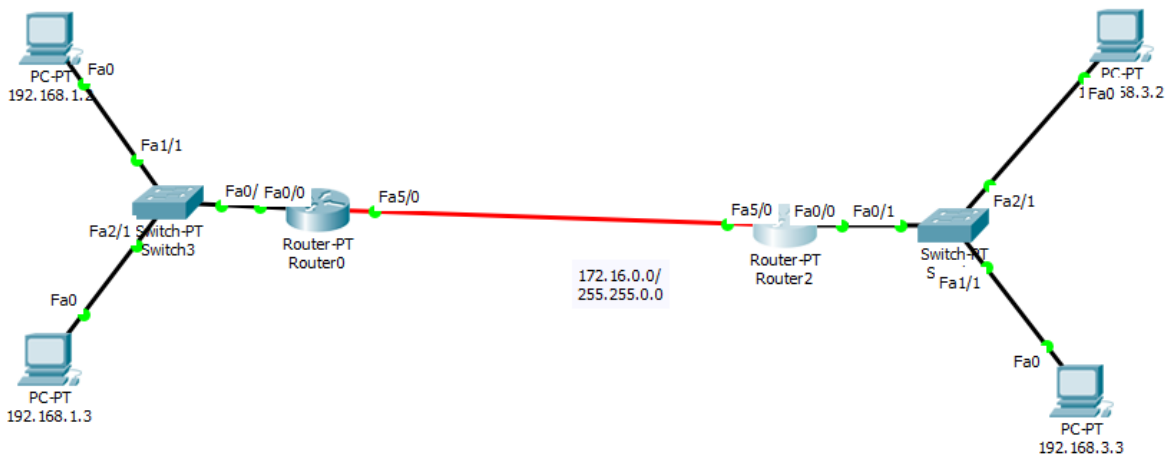
Router-0:

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa5/0
Router(config-if)#ip addr 172.16.0.1 255.255.0.0
Router(config-if)#no shut
```

На Router-1:

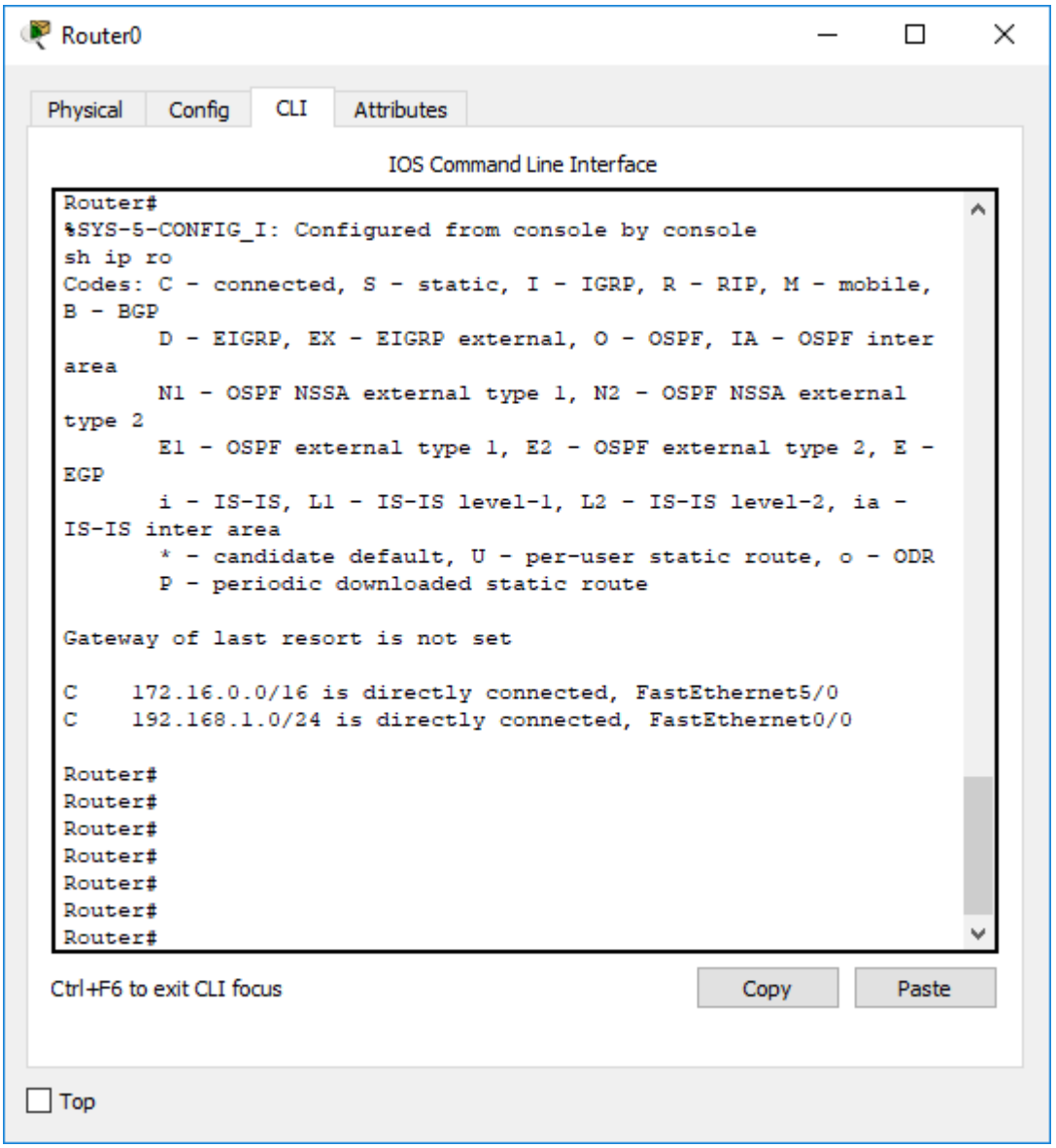
```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int fa5/0
Router(config-if)#ip addr 172.16.0.2 255.255.0.0
Router(config-if)#no shut
```

Завелось:

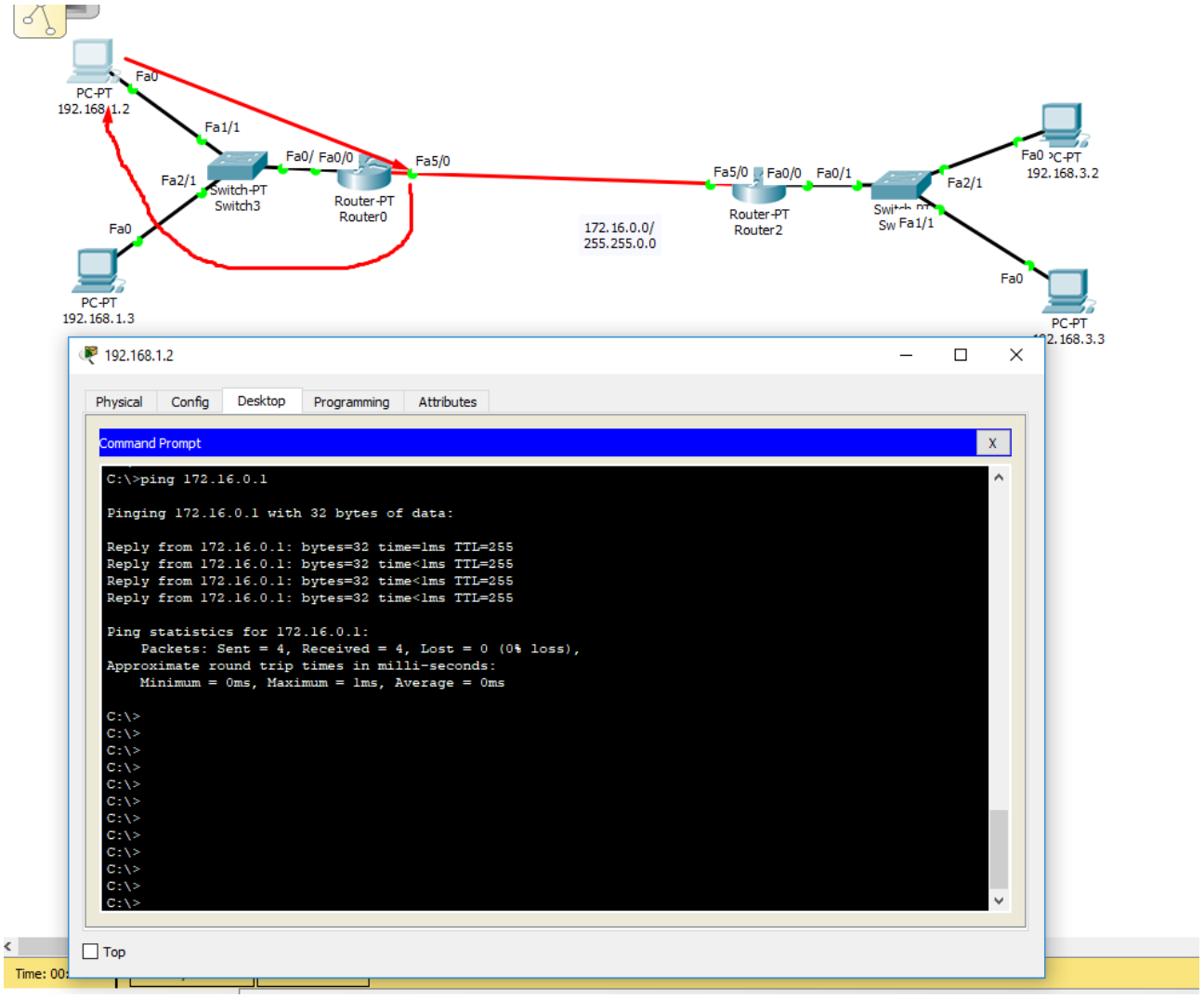


Но каждый компьютер знает, куда отправлять пакеты: если это в той же сети, то в сетевой интерфейс (делается ARP-запрос), либо маршрут по умолчанию — на роутер. У нас две сети, и у каждой есть свой роутер по умолчанию, к которому она непосредственно подключена.

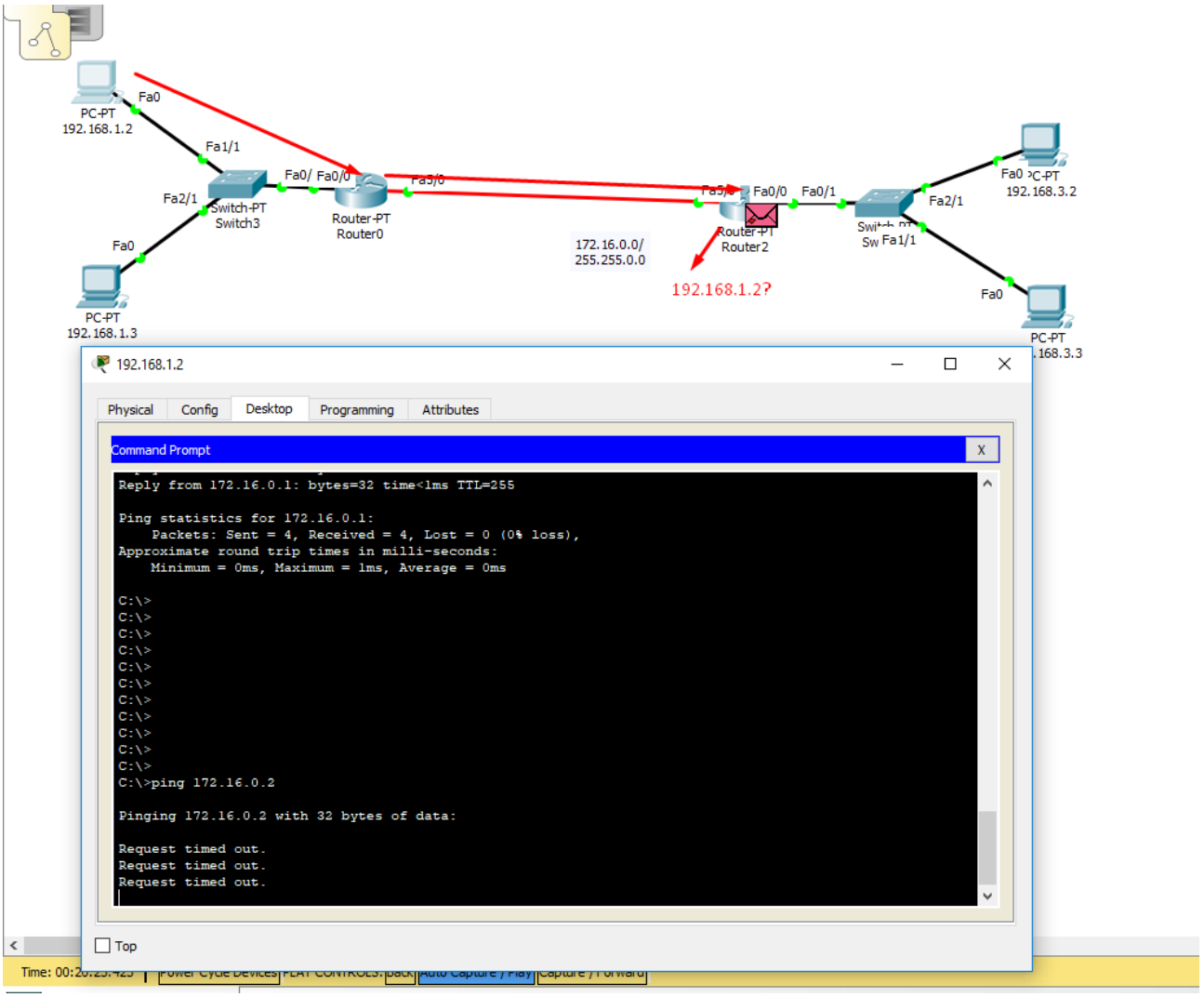
Например, Router-0:



При этом 192.168.1.2 может запинговать 172.16.0.1…



…но не может пинговать 172.16.0.2:



Причем сообщение доходит до узла, но Router-1 не знает, за каким маршрутизатором находится сеть 192.168.1.0.

Поэтому сообщаем Router-0, что сеть 192.168.3.0/25.255.255.0 находится за Router-3. При этом указываем уже доступный адрес Router-3, а именно 172.16.0.2. Этот адрес уже должен пинговаться роутером.

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 192.168.3.0 255.255.255.0 172.16.0.2
Router(config)#ex
```

Проверяем:

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

```
       P - periodic downloaded static route

Gateway of last resort is not set

C    172.16.0.0/16 is directly connected, FastEthernet5/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.3.0/24 [1/0] via 172.16.0.2
```

Теперь сообщаем Router-3, что сеть 192.168.1.0/25.255.255.0 находится за Router-0. При этом указываем уже доступный для Router-3 адрес Router-0, а именно 172.16.0.1. Этот адрес тоже уже должен пинговаться роутером.

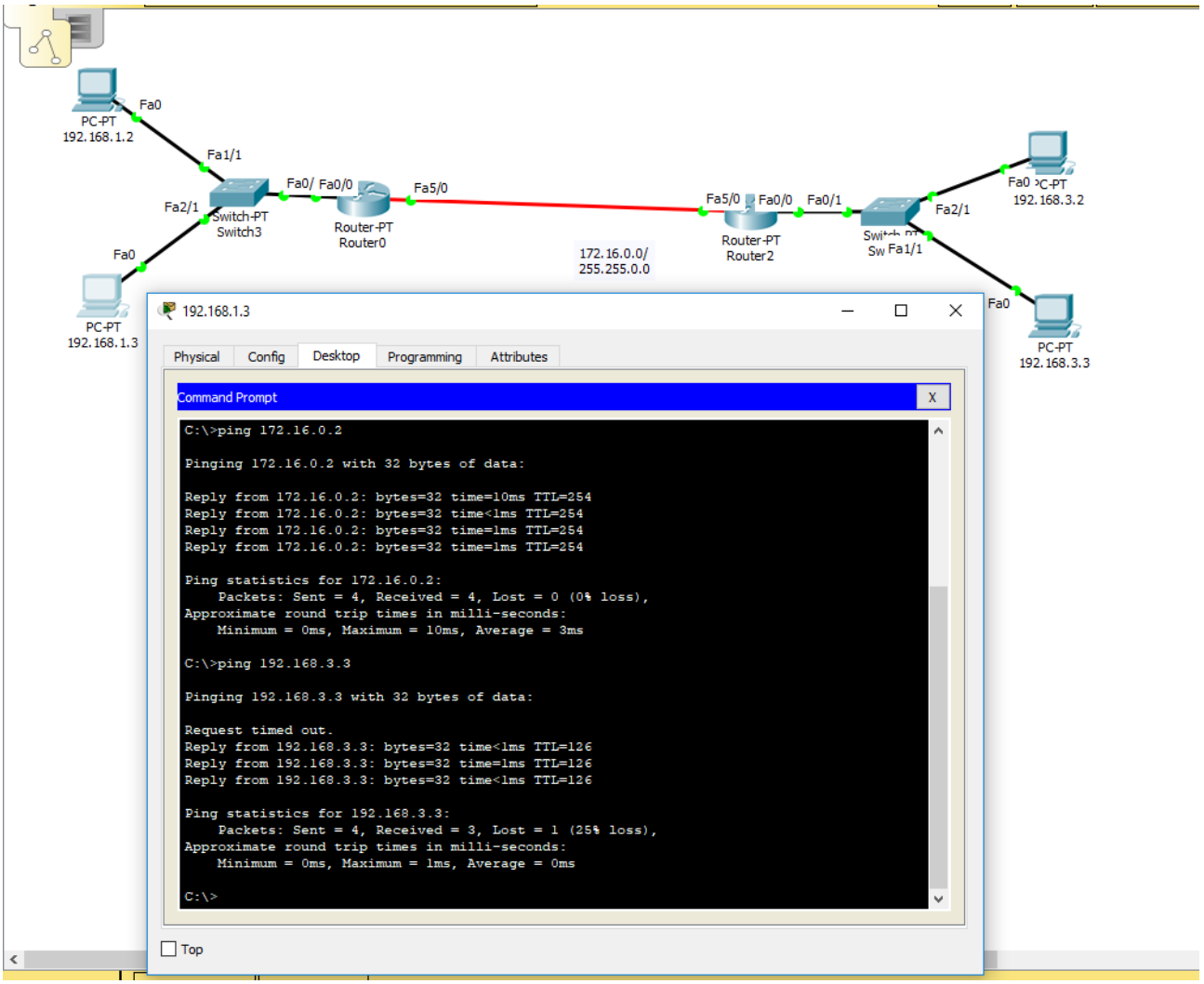
```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 192.168.1.0 255.255.255.0 172.16.0.1
Router(config)#ex
```

Проверяем:

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

Gateway of last resort is not set

C 172.16.0.0/16 is directly connected, FastEthernet5/0
S 192.168.1.0/24 [1/0] via 172.16.0.1
C 192.168.3.0/24 is directly connected, FastEthernet0/0

Router#
```

Теперь каждый роутер знает, за каким роутером находится искомая сеть.

Значит, теерь всё пингуется — как сам роутер (благодаря тому, что сообщение может быть отправлено назад, Router-2 знает сеть 192.168.1.0/255.255.255.0), так и рабочая станция за Router-2 с Router-1 (и обратно).



Практическое задание

Скачать файл **Lesson4Homework.pkt (внимание, он же будет фигурировать и в последующих домашних заданиях!).**

Здесь три корпоративные сети трех офисов одной организации соединены с помощью оптоволоконных линий (точка-точка). Необходимо настроить связь так, чтобы любые два компьютера организации могли связаться друг с другом.

Подпишите работу.

Настройте статическую маршрутизацию. Проверьте, что любые два компьютера из разных сетей попарно пингуются.

Приложите для каждого из роутеров настройки (можно вывод команды **show run**) и, кроме того, вывод **sh ip ro** (таблицу маршрутизации).

Приложите файл .pkt и в комментарии (либо отдельным .docx или .pdf) перечень настроек. Скриншот прилагать не требуется.

Дополнительные материалы

1. https://tools.ietf.org/html/rfc6598.
2. http://just-networks.ru/seti-tcp-ip/protokol-ipv4.
3. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. — СПб.: Питер, 2012. — 960 с. (Глава 5. Сетевой уровень.)

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. http://just-networks.ru/seti-tcp-ip/protokol-ipv4.
2. https://tools.ietf.org/html/rfc6598.