Урок 1

Основы компьютерных сетей. Технология Ethernet. Часть 1

Основные концепции компьютерных сетей. Эталонная модель OSI/ISO и стек протоколов TCP/IP. Введение в технологию Ethernet. Диагностика физического уровня.

Введение

Прежде чем говорить о том, что такое компьютерная сеть, давайте попробуем разобраться, для чего мы её используем. Каждый раз, когда вы хотите передать информационное сообщение (смс, электронные письма, веб страницы, музыкальные треки, фотографии, фильмы и т.д.), вы используете сеть. Не секрет, что большинство людей не видят разницы между Интернетом и компьютерной сетью (а иногда даже и браузером и Интернетом, но мы-то не такие), на самом же деле Интернет — один из примеров компьютерных сетей, хотя довольно известный и масштабный.

Интернет – не просто какая-то отдельно взятая сеть, но Сеть сетей, она объединяет множество сетей во всех уголках мира в единую среду, позволяя обеспечивать передачу информации из одной точки земного шара в другую, хранить информацию и организовывать электронные библиотеки и архивы, не беспокоясь о том, где эти данные хранятся, обеспечивать распределенные вычисления и удаленный доступ к приложениям, который могут находиться за много километров от вашего компьютера.

Можно также встретить обозначение этой Сети сетей такими терминами, как Всемирная сеть или Глобальная сеть, но чаще всего мы просто говорим: Интернет. Технологическая основа сети Интернет (и не только) – стек протоколов TCP/IP, который является одной из основных тем большинства наших занятий. Что это такое, мы рассмотрим чуть позже.

На основе Интернета работает Всемирная паутина (World Wide Web или WWW, ее тоже не стоит путать с Интернетом, хотя она и доступна для нас всех через Интернет) и множество других служб, о которых мы с вами будем говорить в этом курсе. Большинство мобильных приложений, которые предоставляют пользователям функции обмена информацией, используют сетевые технологии. Все веб-приложения размещаются на web-серверах и доступ к ним пользователи получают через компьютерные сети. Сложные информационные системы, которые установлены во всех крупных предприятиях, зависят от работоспособности сети. Знание работы компьютерных сетей позволит не просто написать программу или сервер, который позволит обмениваться пользователям информацией, но выбрать правильную архитектуру и протоколы, которые будут справляться со всеми потребностями пользователей, вовремя доставлять нужную информацию и гарантировать её достоверность.

Данный курс подробно рассматривает сетевые технологии, используемые на самом разном уровне: от физических основ построения сетей до механизмов взаимодействия программного обеспечения с сетью. Так или иначе сетевые проблемы решают самые разные специалисты. Программисты, разрабатывающие веб-приложения и мобильные приложения, системные администраторы, настраивающие сетевое программное обеспечение, сетевые инженеры, обслуживающие телекоммуникационное оборудование. Это очень разные профессии, но каждая из них вносит важный вклад в функционирование компьютерных сетей, как единого целого. При этом у работников каждой из этих трех профессий, как правило, имеются существенные пробелы в понимании других частей сетевых технологий, решаемых их коллегами. Данный курс призван устранить эти пробелы и позволить понимать сетевые технологии на высоком уровне. Программа курса будет полезна программистам, системным администраторам и даже начинающим сетевым инженерам.

Зачем программисту знать, как работают сетевые технологии

Современное общество все больше зависит от работы информационных и инфокоммуникационных систем. В дальнейшем эти тенденции будут только усиливаться, а появление электронной коммерции

и Интернета вещей (IoT, Internet of Things) – это наглядный пример использования информационными системами сетевых возможностей. Если раньше можно было писать десктопные приложения, не задумываясь о сетевых технологиях, сейчас все большее распространение получают мобильные и веб-приложения, вычислительные машины становятся не устройствами «сами по себе», а частью инфраструктуры, где ряд задач решается не на вашем компьютере или смартфоне, а на удаленном сервере. Таким образом в программировании так или иначе придется иметь дело с сетью.

Можно ли заниматься программированием, даже веб-программированием, без понимания работы сетей? Конечно, можно. Для многих это не является проблемой. Но можно ли при этом считать себя высококвалифицированным программистом, который не просто использует инструменты, которые ему предоставили, но понимает их работу. Если большинство задач позволяют не вникать в стек TCP/IP, существует целый круг задач, в которых могут всплывать неожиданные и не вполне очевидные проблемы. Почему приложение не работает? Клиент настроен правильно. Сервер настроен правильно. Необходимо выполнять диагностику, анализировать входящий и исходящий трафик. Может быть проблема в шлюзе провайдера или организации? Может быть там установлен прозрачный прокси-сервер, с которым ваше приложение не совместимо. А может организация вообще пытается контролировать и изменять трафик, из-за чего приложение становится неработоспособным. Такие вещи могут показаться неочевидными, но специалисты, которые успешно справляются с такими задачами, всегда могут претендовать на хорошие должности и оклады, а самое главное, гордиться интересной и не самой легкой работой.

Еще примеры. Безопасно ли пользователи передают данные на сервер. Не все догадываются, что протокол FTP в чистом виде использовать для передачи данных на сервер — дурной тон. С другой стороны, FTP все еще используется для хранения общедоступных архивов. Почему? А как оградить свое приложение (блог, форум, чат) от назойливых посетителей. Как забанить злодея? По MAC-адресу? По IP-адресу? По куки (cookies)?

Ну и конечно же, продиагностировать проблемы в собственной сети. Настроить домашний роутер, определить, почему не работает сеть — все это тоже не последние вещи.

Знание сетевых технологий, понимание работы стека протоколов TCP/IP, клиент серверной архитектуры и протоколов прикладного уровня – это основа для разработки сетевых приложений. Не важно, что нужно написать: веб-приложение, игру или сложную банковскую систему – все эти программы используют компьютерные сети для получения и передачи данных.

Разрабатывая приложение, архитектор или программист всегда должен ответить на ряд вопросов, от которых зависит то, как будут использованы сетевые сервисы:

- масштабирование приложения;
- производительность приложения;
- безопасность приложения.

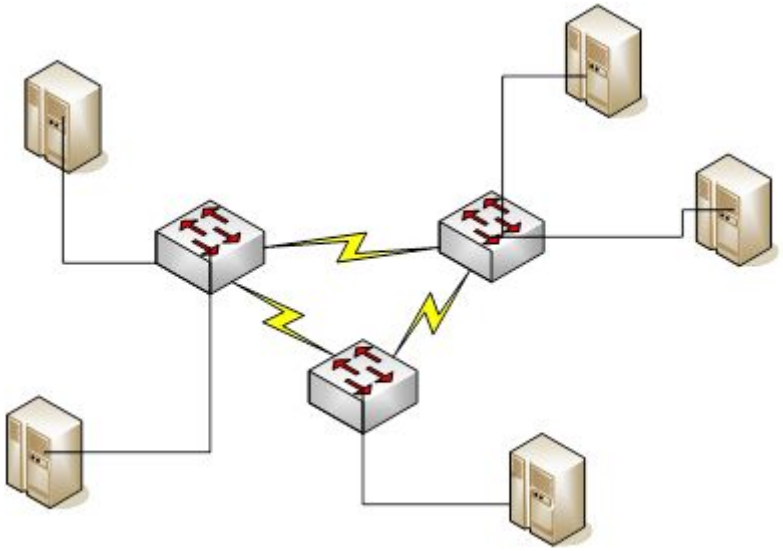
Все перечисленные вопросы в том числе зависят и от архитектуры сети и используемых сетевых протоколов.

Какие технологии и протоколы лучше использовать, мы узнаем в течение нашего курса. Для того чтобы разобраться, как работают сети и из чего они состоят, мы начнём с основных определений. Заучивать эти определения не надо, гораздо важнее понимать их.

Определения

Коммуникационная сеть – система, состоящая из объектов, называемых пунктами (узлами) сети и осуществляющих функции генерации, преобразования, хранения и потребления некоторого продукта,

а также линий передачи (связей, коммуникаций, соединений), осуществляющих передачу продукта между пунктами.



Информационно-вычислительная сеть (компьютерная сеть) – коммуникационная сеть, в которой продуктом генерирования, переработки, хранения и использования является информация, а узлами сети – вычислительное оборудование.

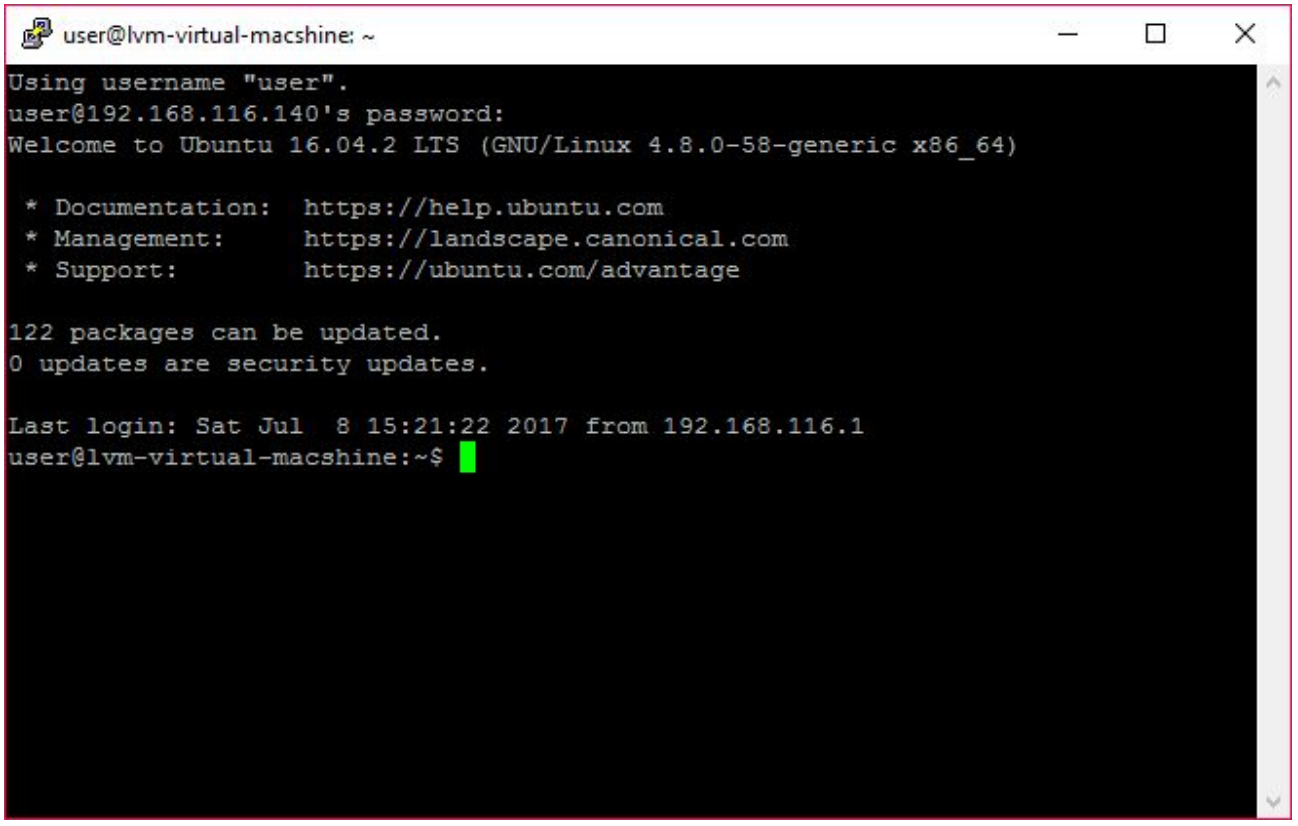
Согласно современным представлениям информация считается нематериальной, а то, что содержится в структуре объектов, принято называть данными.

В работе с государственными органами и госзаказчиком часто фигурируют термины «Информационно-коммуникационная сеть» и «Информационно-коммуникационная сеть Интернет». Согласно глоссарию, опубликованному на сайте Министерства связи и массовых комуникаций (http://minsvyaz.ru/ru/documents/3464/) Информационно-коммуникационная сеть — совокупность технических средств для передачи и обработки информации.

Терминал/terminal – это оконечное устройство сети. Например, компьютер, телевизор, радиостанция (устройство для приема и/или передачи радиоволн), стационарный, мобильный или даже IP-телефон, факсовый аппарат.

Компьютерный терминал/computer terminal – устройство ввода/вывода, рабочее место на многопользовательских ЭВМ, монитор с клавиатурой.

Примеры терминальных устройств: телетайп, терминал, консоль (устройство), терминальный сервер, тонкий клиент. Примеры терминальных программ: консоль (текстовый интерфейс), эмулятор терминала, xterm, telnet, ssh, putty. Терминалом может выступать и полноценный компьютер (как правило, с меньшими вычислительными возможностями), играющий роль оконечного устройства для другой, более мощной ЭВМ (тонкий клиент, толстый клиент).



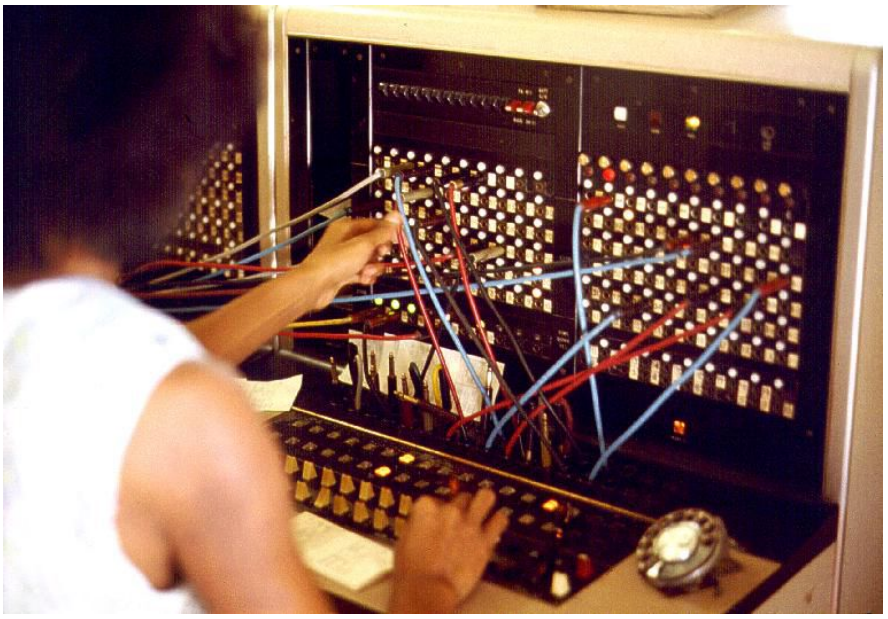
Putty — программный терминал.

Телетайп из 80-х, использовался для работы на PDP 11-05

Канал связи/connection channel – совокупность линий связи (радио, радиорелейных, спутниковых, кабельных, оптических) и оборудования передачи (усилители, преобразователи, мультиплексоры, коммутаторы и т.д.), обеспечивающая передачу нормализованного сигнала между 2 или более точками.

Линия связи/connection line – совокупность технических устройств и физической среды, обеспечивающая передачу электрических сигналов от передатчика к приемнику.

Коммутатор/switch – телефонная станция, пакетный переключатель, устройство адресного распределения сигналов пользователей, в малых или специализированных сетях может отсутствовать.

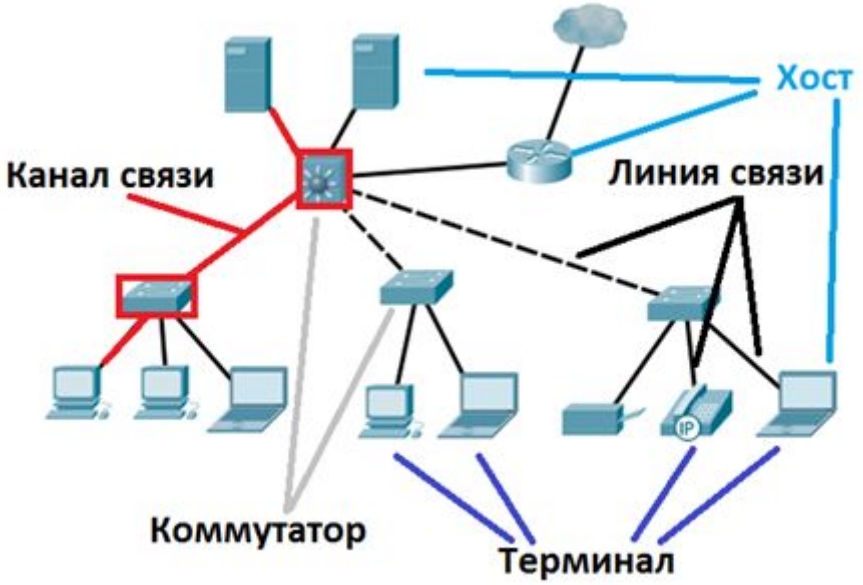


Ручной телефонный коммутатор.

Автор: Joseph A. Carr - http://www.JoeTourist.net/, Attribution, https://commons.wikimedia.org/w/index.php?curid=5169771

Современный сетевой коммутатор — Gigabit Ethernet Switch

Хост/host – любое устройство в сети, предоставляющее сервисы формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определённое на этих интерфейсах. В более частном случае под хостом могут понимать любой компьютер, сервер, маршрутизатор, подключенный к локальной или глобальной сети.



Глобальная сеть Интернет

История Интернет

Хронология важнейших событий из истории развития компьютерных сетей:

Этап	Время
Первые глобальные связи компьютеров, первые эксперименты с пакетными сетями	Конец 60-х
Начало передач по телефонным сетям голоса в цифровой форме	Конец 60-х
Появление больших интегральных схем, первые мини-компьютеры, первые нестандартные локальные сети	Начало 70-х
Создание сетевой архитектуры IBM SNA и протокола X.25	1974
Появление персональных компьютеров, создание Интернета в современном виде, установка на всех узлах стека TCP/IP	Начало 80-х
Появление стандартных технологий локальных сетей (Ethernet — 1980 г., Token Ring*, FDDI* — 1985 г.)	Середина 80-х
Начало коммерческого использования Интернета	Конец 80-х
Изобретение Web	1991

*- морально устаревшие технологии, проигравшие Ethernet .

Службы сети Интернет

Наиболее популярные службы сети Интернет:

- World Wide Web;
- Электронная почта;
- Телеконференции;
- Электронный журнал;
- Чат;
- ICQ;
- FTP - серверы и SFTP-серверы;
- Torrent;
- Поисковые системы;
- Интернет-магазины;
- Интернет-аукционы;
- Интернет-радио;
- Интернет-телевидение;
- IP-телефония;
- Веб-форумы;
- Блоги;
- Видеохостинги;

- Фотохостинги;
- Социальные сети;
- Вики-проекты;
- Электронные платёжные системы.

Основы сетевых технологий

Виды межсетевого обмена

По способу взаимодействия службы в сети делят на два: «клиент-сервер» (client-server) и «равный с равным» (peer-to-peer/p2p).

Модели «клиент-сервер» и «равный с равным» могут использоваться одновременно. Также подобные виды взаимодействия выделяют и для сетей. Сети, построенные по принципу «равный с равным», называют также одноранговыми сетями, в которых все компьютеры имеют одинаковый статус - ранг. Сети, в которых есть специализированные узлы, предоставляющие остальным сервисы (сервера), называют многоранговыми.

Давайте более подробно разберем прозвучавшие определения сервер и клиент.

Сервер/server (программное обеспечение) — программное обеспечение, принимающее запросы от клиентов.

Сервер/server (аппаратное обеспечение) — компьютер (или специальное компьютерное оборудование), выделенный и/или специализированный для выполнения определенных сервисных функций.

Выделенный/dedicated сервер — это сервер, занимающийся только сетевыми задачами.

Невыделенный сервер может помимо обслуживания сети выполнять и другие задачи. Любой персональный компьютер может выступать в роли невыделенного сервера.

Клиент/client — аппаратный или программный компонент вычислительной системы, посылающий запросы серверу. Клиентом называется абонент сети, который только использует сетевые ресурсы, но сам свои ресурсы в сеть не отдаёт, то есть сеть его обслуживает, а он ей только пользуется. Компьютер-клиент также часто называют рабочей станцией. Каждый компьютер может быть одновременно как клиентом, так и сервером.

Под сервером и клиентом часто понимают также не сами компьютеры, а работающие на них программные приложения. В этом случае то приложение, которое только отдает ресурс в сеть, является сервером, а то приложение, которое только пользуется сетевыми ресурсами — клиентом.

ЭВМ, которая предоставляет некие услуги, обычно именуется сервером, ЭВМ, которая запрашивает эти услуги, выступает в роли клиента. Терминология клиент-сервер применяется как к аппаратному обеспечению (ваш ноутбук будет выступать клиентом, а компьютер-сервер в серверной стойке в дата-центре — соответственно — сервером), так и к программному (например, если веб-сервером выступает Apache или Nginx, роль клиента на вашем компьютере будет играть Google Chrome или, например, Mozilla Firefox).

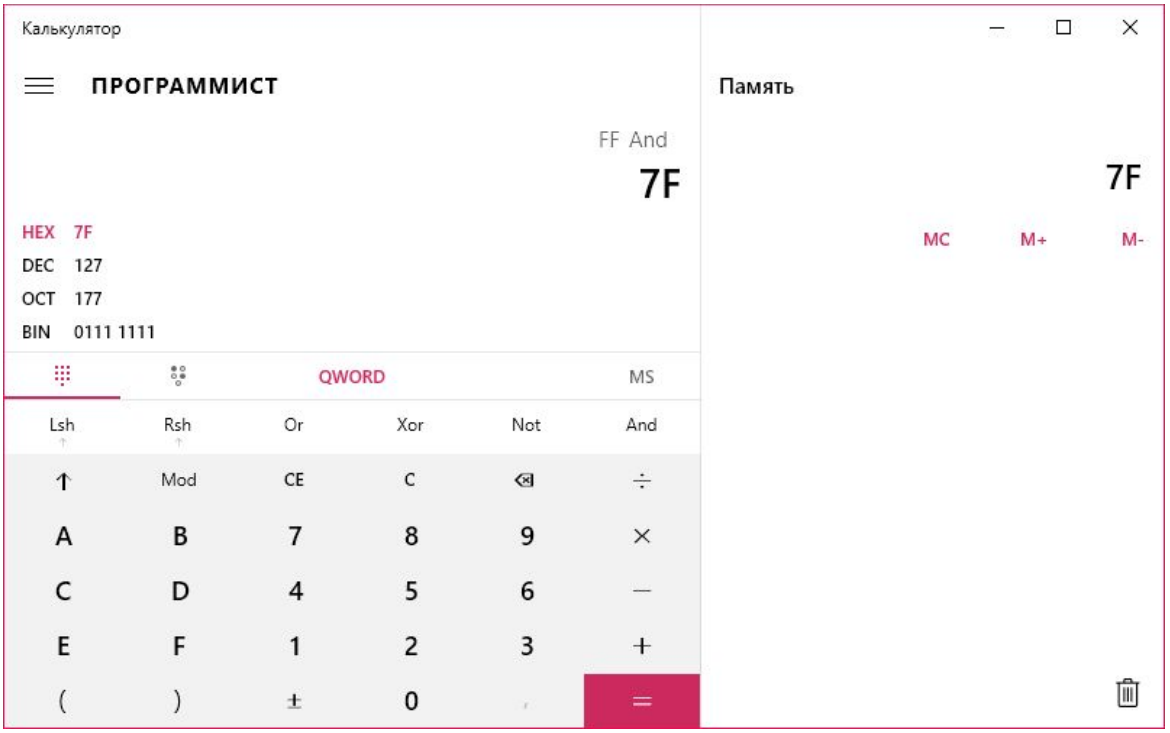
Отметим, что в силу специфики стека TCP/IP даже при p2p взаимодействии, как минимум одно приложение должно ожидать соединение, а другое подключаться. При этом одно и то же приложение в разное время или даже одновременно может выступать и в роли сервера (для одних клиентов), и в роли клиента (подключаясь к другим серверам). Это тоже распространенное явление.

Также следует заметить, не все оборудование является клиентом или сервером. Так концентраторы, коммутатору, маршрутизаторы, шлюзы не являются ни клиентом, не сервером, хотя современные устройства (маршрутизаторы, домашние WiFi-роутеры и т.д.) также могут предоставлять серверные услуги.

Какой минимальный набор знаний нужен, чтобы двигаться дальше

Кто-то уже знает, что такое IP-адрес, TCP- и UDP-порты, доменные имена и команда ping. Без хотя бы самого простого знакомства с этими вещами двигаться дальше невозможно.

В отличие от людей, компьютеры работают не с символьными именами, а только с числами. Более того, это мы привыкли записывать адреса в десятичной и шестнадцатиричной системе, сами же компьютеры и сетевые устройства работают только в двоичной системе. Если вы еще не знакомы с такими вещами, как двоичная и шестнадцатеричная система счисления, перевод из одной системы в другую, логические операции И (AND), ИЛИ (OR), НЕ (NOT), ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR), крайне рекомендуется изучить, это будет полезно не только в изучении сетей, но и вообще необходимо знать каждому программисту.



Компьютеры, работающие в сети, взаимодействуют благодаря IP-адресам (IP расшифровывается просто Internet Protocol — Интернет-Протокол). IP-адрес позволяет глобально идентифицировать ЭВМ в сети, где бы компьютер находился. Если вы находитесь в Москве, и ваш компьютер с IP-адресом 1.2.3.4 подключен к сети Интернет, вы можете установить связь с компьютером, находящимся, скажем в Лос-Анджелесе, и имеющим IP-адрес, например 130.140.105.1

IPv4-адрес (как правило, используются адреса IP версии 4, хотя постепенно и осуществляется переход на IPv6) состоит из 4-х чисел (байт), именуемых октетами, и записываемых, как правило, в десятичной системе счисления. Есть специальные IP-адреса (адреса сетей и широковещательные адреса), но большинство IP-адресов служат для идентификации хостов, то есть неких машин (не важно, в роли клиентов или серверов они выступают). Важно знать, что каждый октет IP-адреса может иметь значения от 0 и до 255, не больше, все потому, что IP-адрес на самом деле состоит из бит, и если его перевести в шестнадацтеричные числы, мы получим как раз те же самые 4 октета.

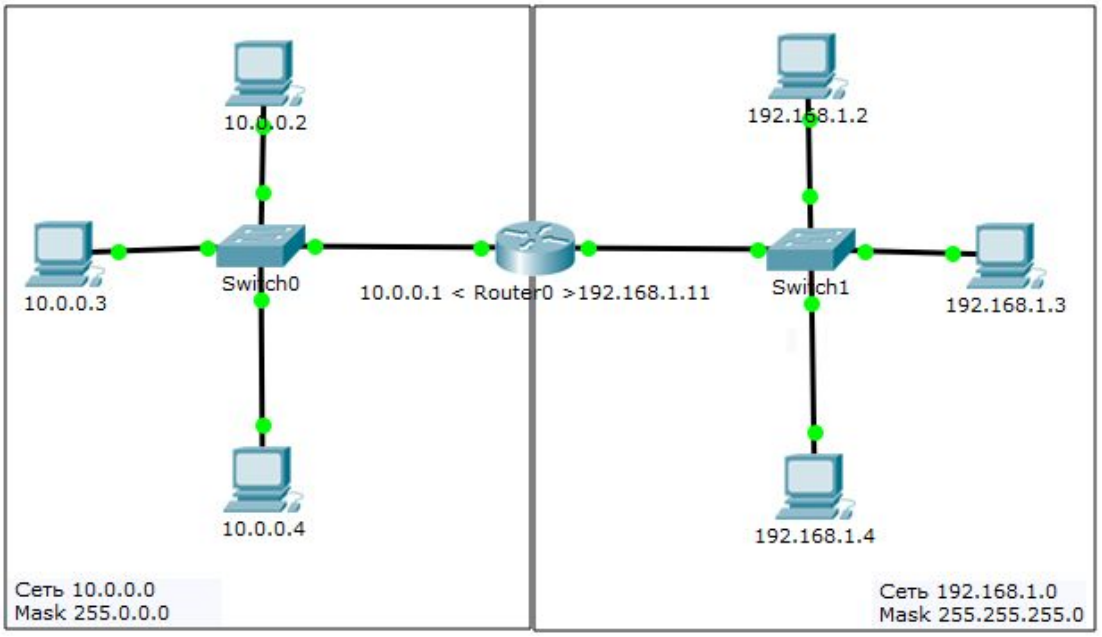
Посмотрим, как выглядит адрес 64.233.164.139 в шестнадцатеричной и двоичной системе счисления.

В десятичной системе счисления	64.	233.	164.	139
В шестнадцатеричной системе счисления	40	E9	A4	8B
В двоичной системе счисления	0100 0000	1110 1001	1010 0100	1000 1011

Вы можете посмотреть, как выглядит любой другой IP-адрес, используя (для Windows) калькулятор Calc или аналогичный, перейдя в режим “Программист”.

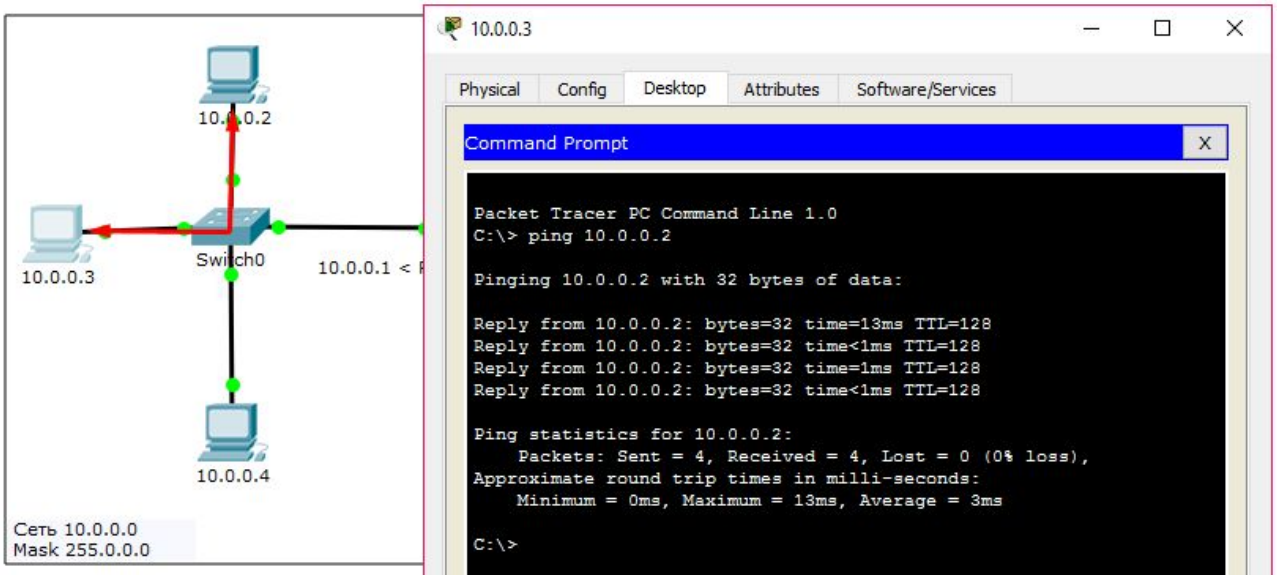
Таким образом, адрес 64.233.164.139 в шестнадцатеричной системе счисленяи выглядит как 40.E9.A4.8b и в двоичной системе счисления как 0100 0000.1110 1001.1010 0100.1000 1011. (Точки и пробелы вставлены только для удобства записи)

В разных сетях IP-адреса выглядят по-разному. Не вдаваясь в подробности, можно отметить, что IP-адрес содержит часть, идентифицирующую сеть, и часть, идентифицирующую хост. Таким образом, чтобы компьютеры могли работать, в одной сети они будут начинаться одинаково, и наоборот, разные сети будут иметь отличающиеся в первых октетах числа.



На картинке мы видим две сети. Компьютеры объединяются в сеть коммутатором (свитчем). Одна сеть содержит IP-адреса, начинающиеся с 10.0.0., вторая – с 192.168.1. Маршрутизатор (роутер) объединяет обе сети, и обладает двумя IP-адресами (существуют случаи, когда это не так, но мы рассматриваем самый простой пример). Обратите внимание на маску. Если мы возьмем маску сети и адрес, мы можем проверить, принадлежит он этой сети или нет. Если в маске октет 255, значит октет оставляем, если 0, заменяем на 0. В результате мы получаем адрес сети. (На самом деле система более сложная, это простой случай).

В рамках одной сети можно обмениваться информацией между компьютерами если они соединены в одну сеть и имеют адреса в одной сети.



Проверка выполняется в командной строке (и в Windows и в GNU/Linux и в командной строке компьютера в Cisco Packet Tracer) командой ping.

Чтобы запустить командную строку в Cisco Packet Tracer, необходимо кликнуть на компьютер, кликнуть на вкладку Desktop (рабочий стол) и кликнуть на иконку Command Prompt (командная

строка). Закрыть командную строку можно нажав на крестик напротив синей полоски с надписью «Command Prompt».

В Windows можно нажать Win+R, ввести CMD и нажать Enter.

В Windows и командной строке компьютера в Cisco Packet Tracer

```
C:\> ping 10.0.0.3
```

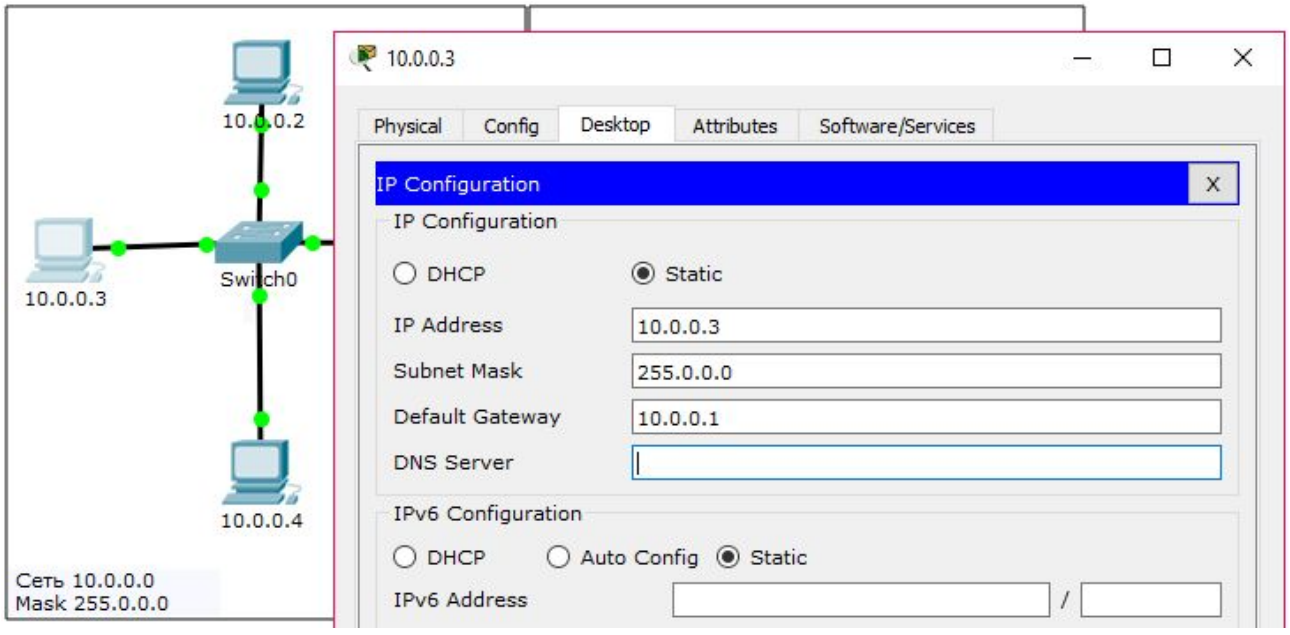
В терминале GNU/Linux или Mac OS X:

```
user@host:~$ ping 10.0.0.3
```

В Linux в текстовый терминал можно перейти нажав Ctrl-Alt-F1 (..F6, возврат в графическую среду Alt-F7), либо запустив в графической среде эмулятор терминала сочетанием Ctrl-Shift-T.

Вы пишете адрес, с которым хотите проверить связь, и ваш компьютер отправляет по очереди специальные пронумерованные сообщения (ICMP-пакеты, ICMP — Internet Control Message Protocol — протокол межсетевых управляющих сообщений), а проверяемый компьютер отправляет их назад. Если сообщение дошло до компьютера (не было потеряно) и вернулось, то отображаются данные об ответе и затраченном времени.

То есть для связи с компьютерами в одной сети достаточно, чтобы IP-адреса принадлежали одной сети, а сами компьютеры были подключены в сеть. Этого недостаточно для отправки сообщений в другую сеть (например на адрес 192.168.1.2). Для этого на каждом компьютере должен быть настроен не только IP-адрес, но и еще и адрес шлюза. У шлюза два адреса, поэтому для компьютеров в сети 10.0.0.0 адресом шлюза будет указываться 10.0.0.1, а для сети 192.168.1.0 — соответственно, 192.168.1.1.

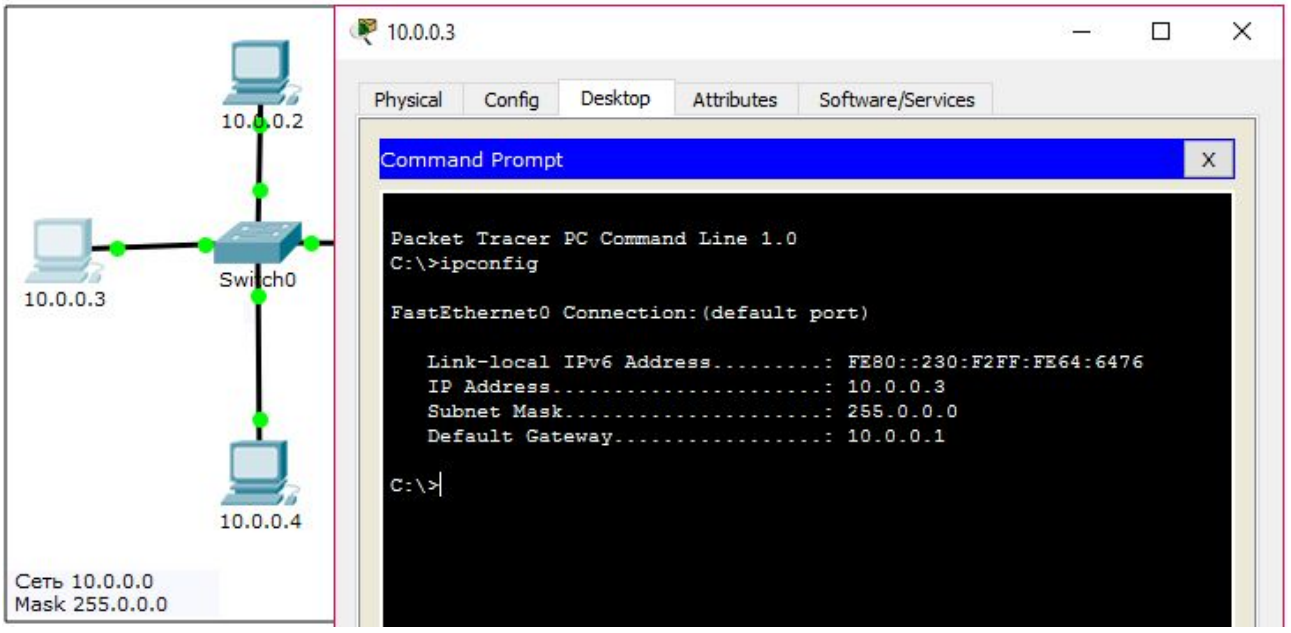


Чтобы посмотреть/изменить настройки IP-конфигурации компьютера в Cisco Packet Tracer, необходимо кликнуть на компьютер.

Те же настройки посмотреть (и даже поменять) можно в командной строке с помощью команды ipconfig (в Windows и в командной строке компьютера в CiscoPacket Tracer), либо команды ifconfig (в GNU/Linux и других UNIX-подобных системах).

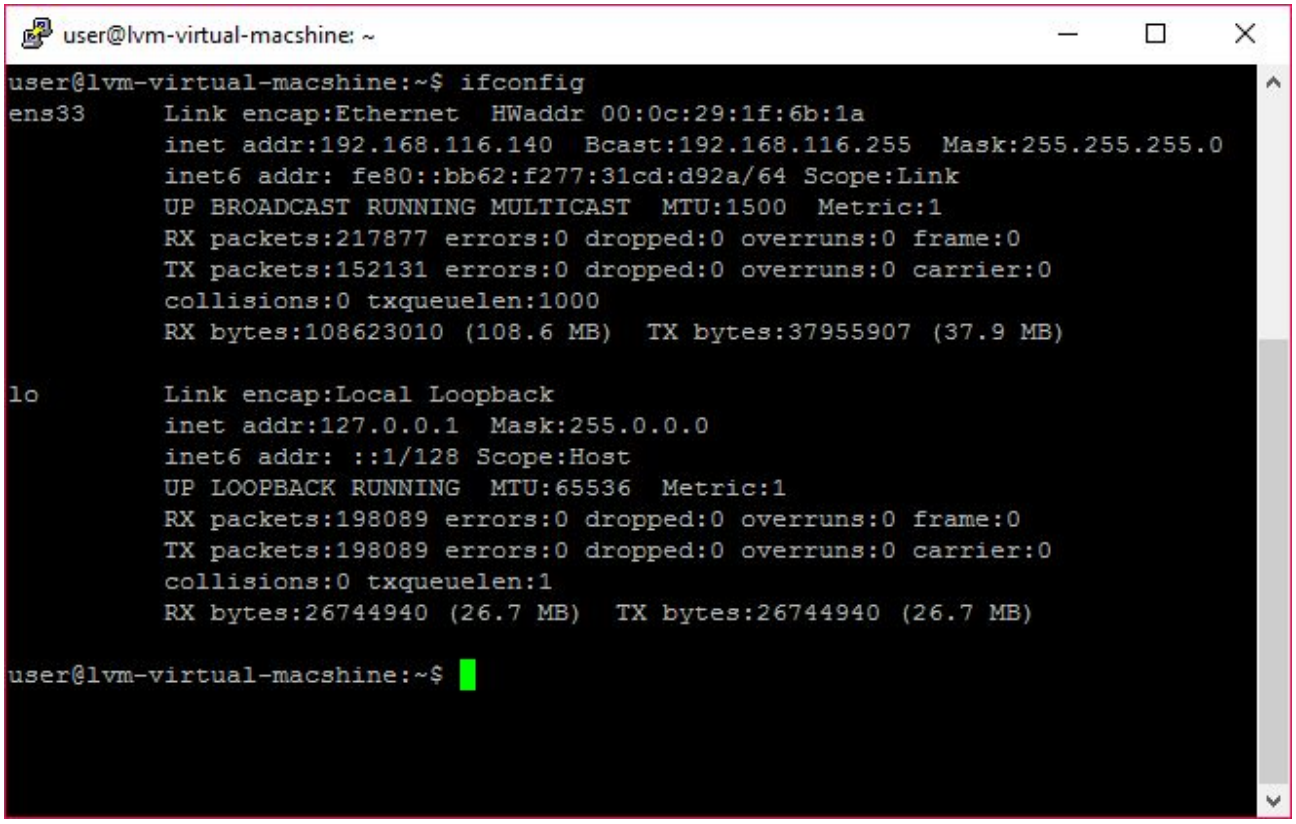
В Windows и командной строке компьютера в Cisco Packet Tracer

```
C:\> ipconfig
```

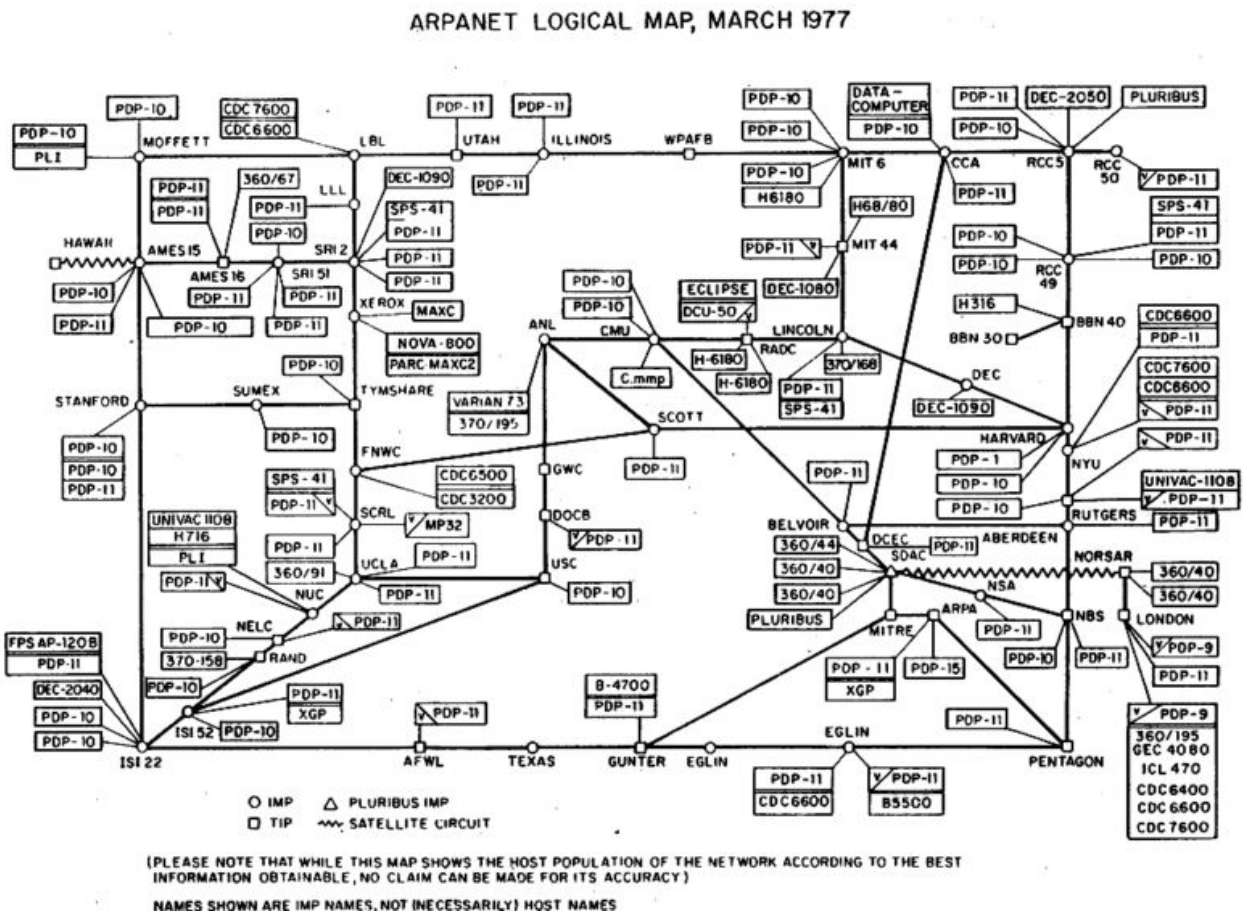


В терминале GNU/Linux или Mac OS X:

```
user@host:~$ ifconfig
```

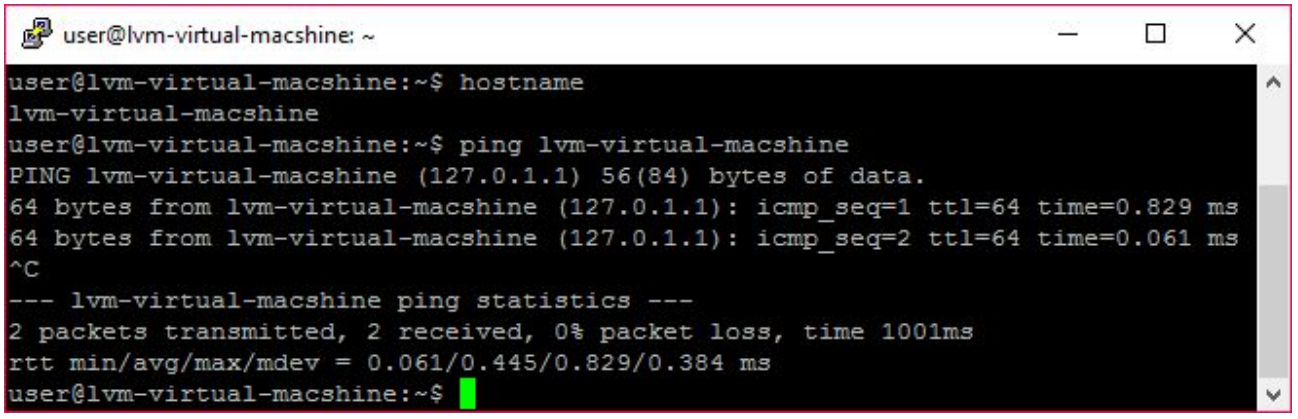


Чтобы посмотреть сетевые параметры в графическом режиме в Windows (в зависимости от версии), необходимо кликнуть на значок сети, перейти в центр управления сетями и общим доступом и кликнуть на подключение, например, Ethernet. Далее выбираем сведения (посмотреть) или свойства (чтобы изменить). Без должного опыта менять на рабочем компьютере крайне не рекомендуется, так как можно сделать сеть не рабочей.



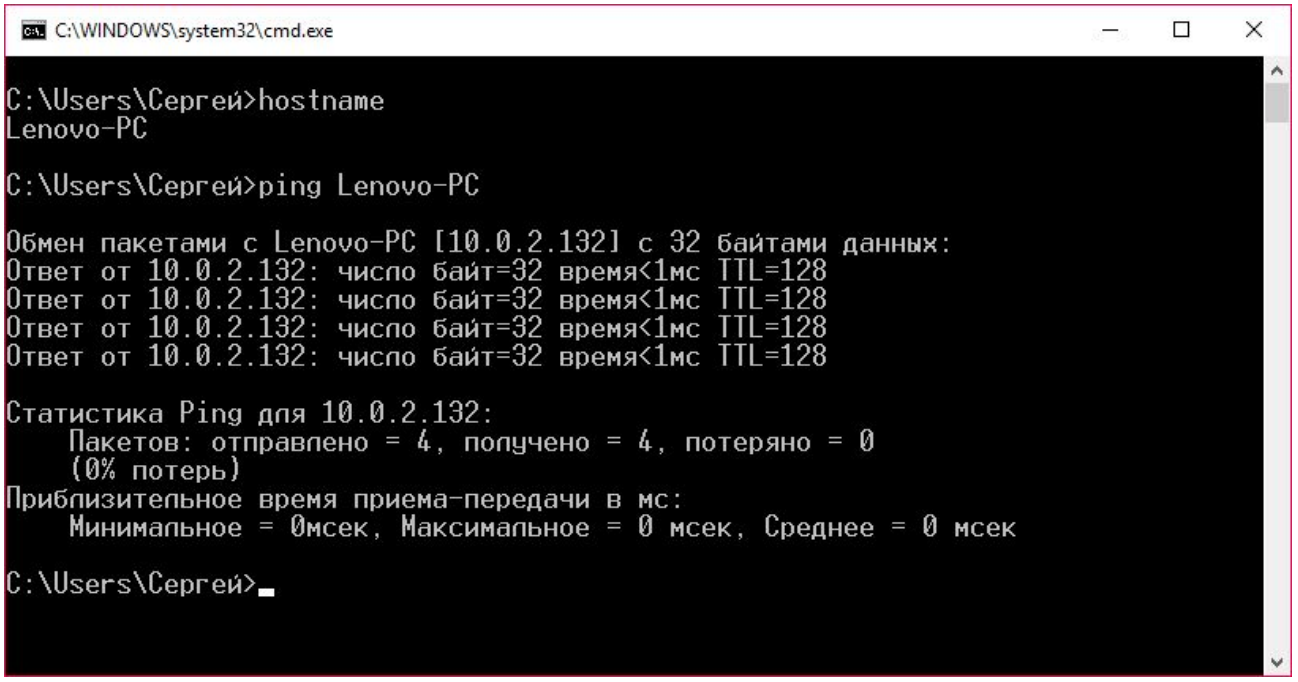
На картинке сеть 1977 года. IP-протокол, известный сейчас как IPv4 появился в 1980 году. Из этой сети вырос весь современный Интернет, и та маленькая сеть не сравнится в масштабах с существующей. Но даже то количество компьютеров, которое есть на схем, довольно велико, чтобы оперировать вручную IP-адресами (или иными адресами сетевого уровня). Поэтому появился такой термин как имя хоста — hostname. IP-адресу можно сопоставить некое символьное имя, имя компьютера. В командной строке Windows и Linux hostname покажет имя вашего компьютера. Его можно использовать вместо IP-адреса. В GNU/Linux:

```
user@host: ~ $ hostname
user@host: ~ $ ping lvm-virtual-macshine
```



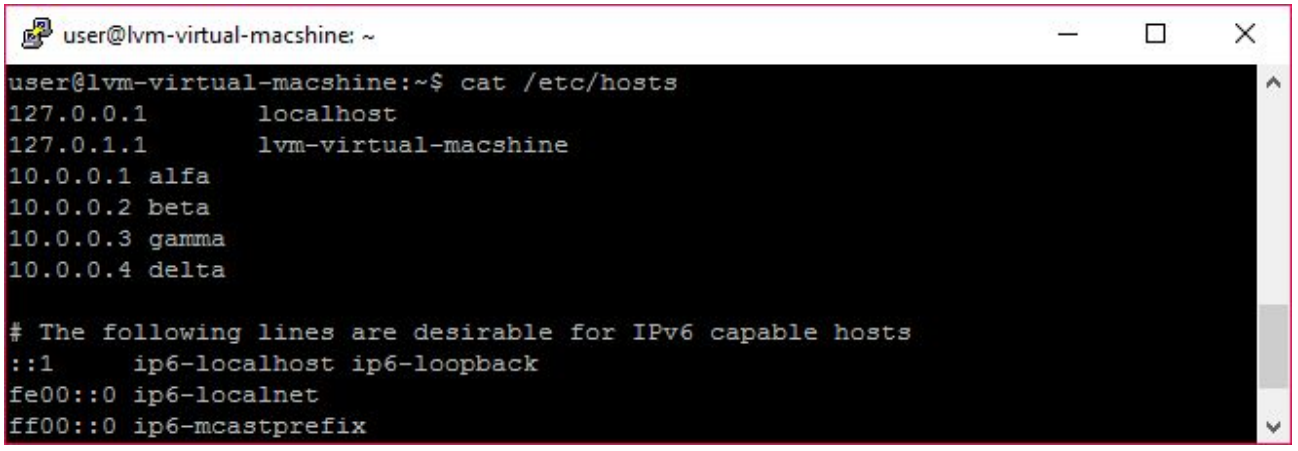
В Windows:

```
C:\>hostname
Lenovo-PC
C:\>ping Lenovo-PC
```

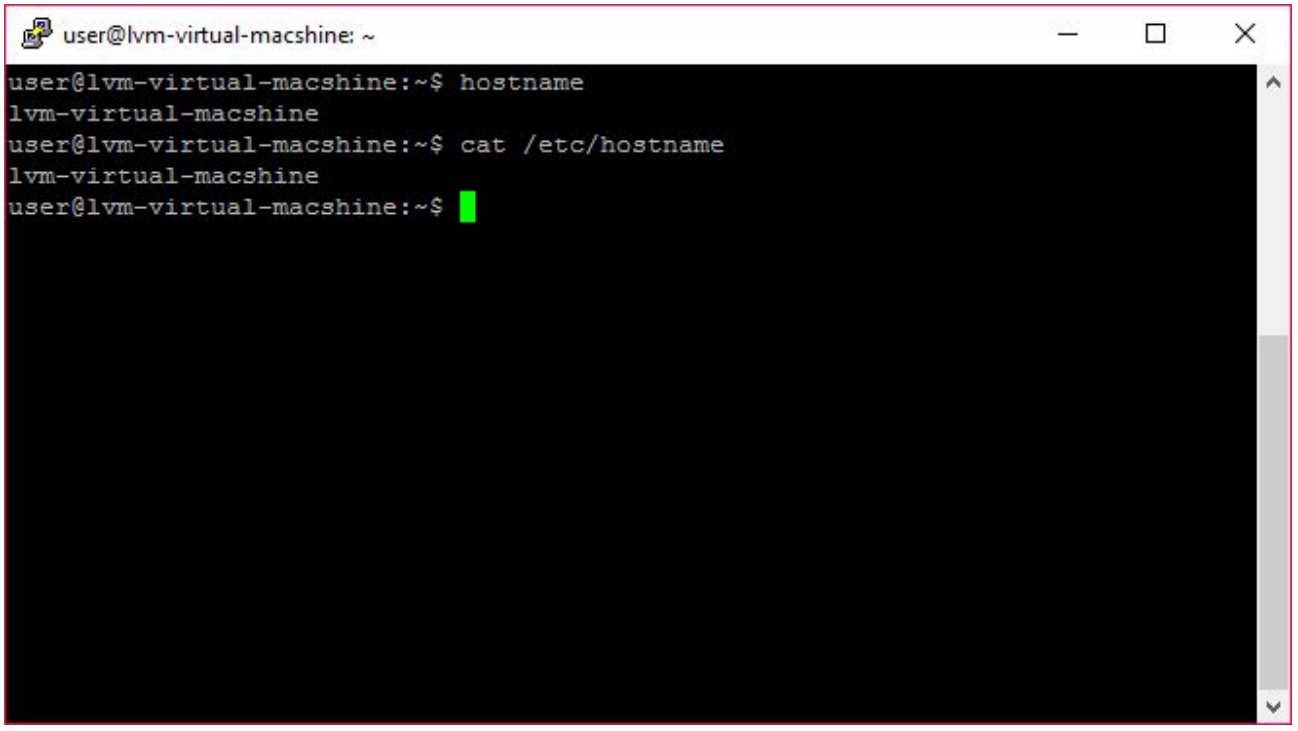


Но компьютеры между собой соединяются используя IP-адреса. Когда вместо IP-адреса мы пишем имя хоста, утилита ping сама преобразует имя хоста в IP-адрес, что мы и видим на рисунках — ответ приходит от указанного IP-адреса. Но откуда компьютер знает, какой адрес.

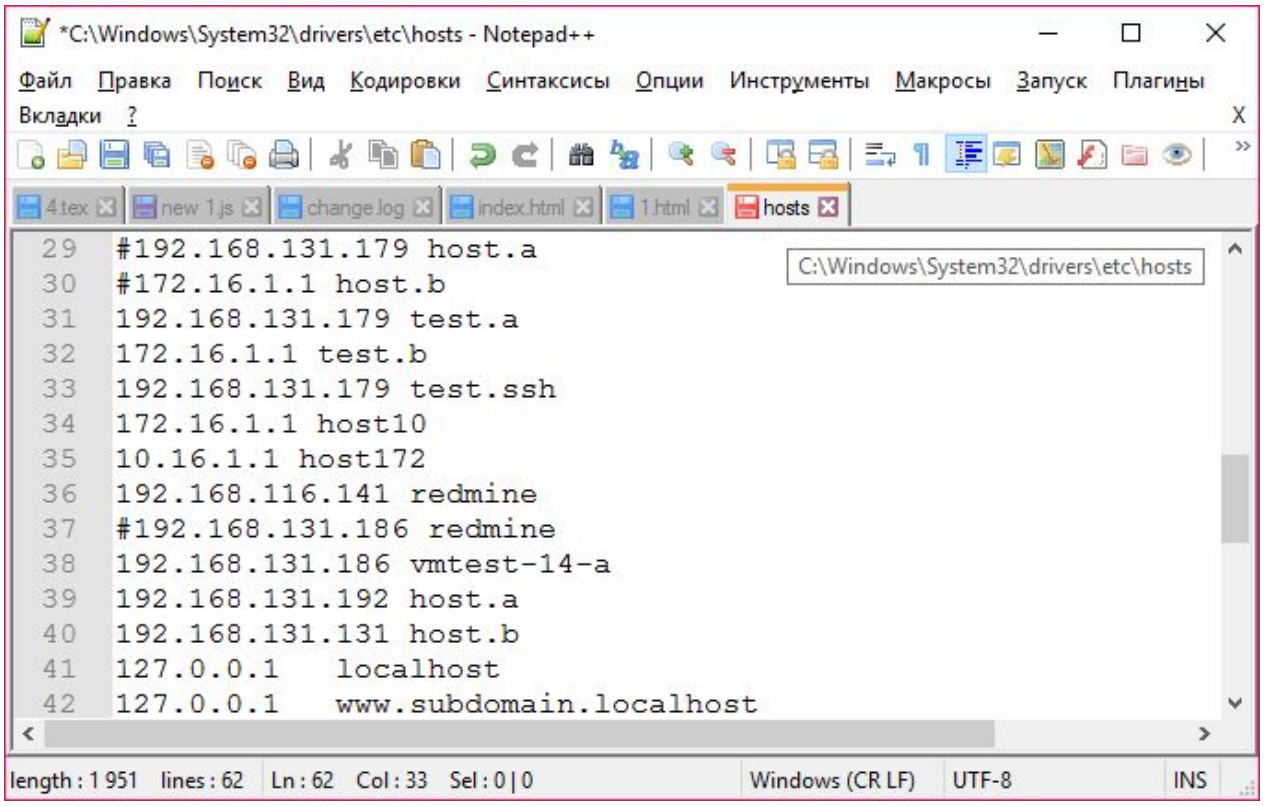
В то время уже распространились UNIX-подобные системы (по большей части UNIX BSD), а в подходе unix-way конфигурационные файлы хранятся в файлах (в директории /etc). Поэтому на всех компьютерах сети появился файл /etc/hosts, который содержал соответствия IP-адрес – имя хоста. Если в системе появлялся новый хост (новый компьютер) необходимо было изменять файл /etc/hosts на всех машинах (команда cat в Linux позволяет вывести содержимое файла на консоль).



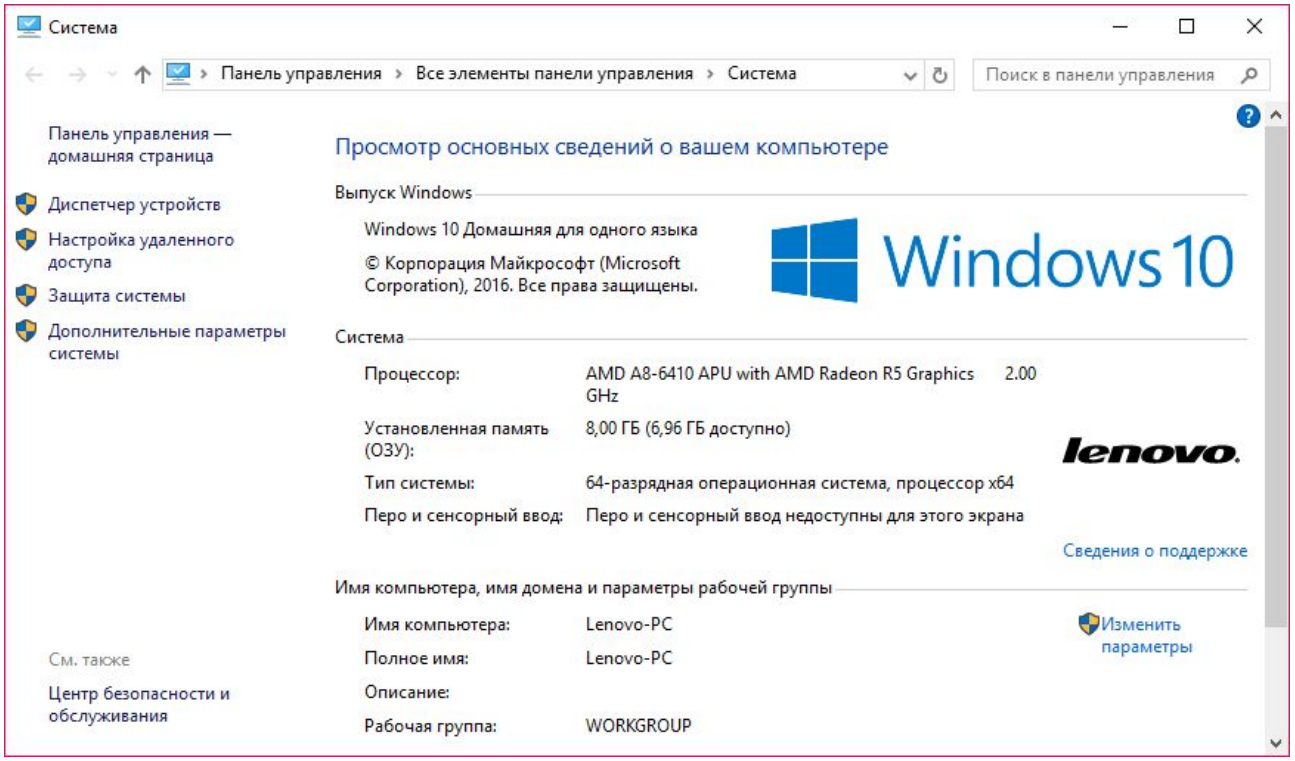
Имя же текущей машины также хранилось в файле /etc/hostname



Список соответствий IP-адрес и хост в Windows хранится идентично. Обычно это файл C:\Windows\System32\drivers\etc\hosts



Имя же компьютера можно посмотреть/изменить в свойствах “Мой компьютер”/””Этот компьютер” или в панели управления.



Такой способ обмена информацией о именах хостов (с помощью ручной правки файла /etc/hosts или копирования его вручную с машины на машину) оказался не сильно удобным. А когда сеть разрослась, это стало физически невозможно. Поэтому разработали систему доменных имен.

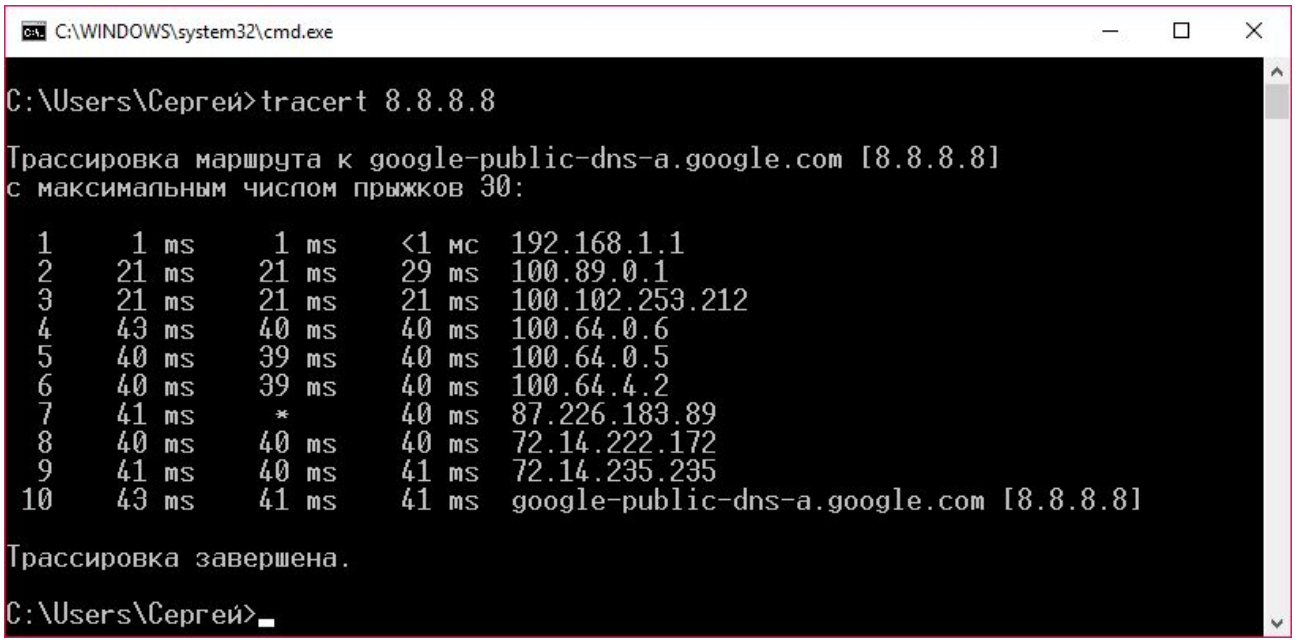
Доменные имена похожи на имена хостов, но построены по иерархическому принципу. Были созданы несколько зон верхнего уровня, .com — для коммерческих сайтов, .org — для некоммерческих, .edu — для образовательных, .mil — для военных. Организации получали домены в этих зонах, например, .mit.edu — зона Массачусетского технологического института, .caltech.edu — доменная зона Калифорнийского технологического института. При этом администратор каждой зоны уже сам выделяет поддомены для соответствующих сервисов и служб. Так веб-сайт Массачусетского технологического института находится на домене web.mit.edu, а сайт Калифорнийского технологического института на домене www.mit.edu

DNS — Domain Name System — система доменных имен и одноименный протокол обеспечивает иерархическое распределенное хранение доменных имен в сети Интернет и возможность поиска соответствия IP-адреса для нужного домена. Для конечного пользователя, чтобы можно было не просто иметь доступ к компьютерам в сети Интернет-по IP-адресам, а используя доменные имена, в настройках TCP/IP соединения также необходимо указать один или два DNS-сервера, которые могут обеспечить поиск и кеширование IP-адреса для указанного клиентом доменного имени. Такие DNS-сервера называют кеширующими. Как правило, в настройках указывается адрес DNS-сервера провайдера, либо используются публичные DNS-сервера (от Google, с IP-адресами 8.8.8.8 и 8.8.4.4, от Яндекс, с именами 77.88.8.8, 77.88.8.1).

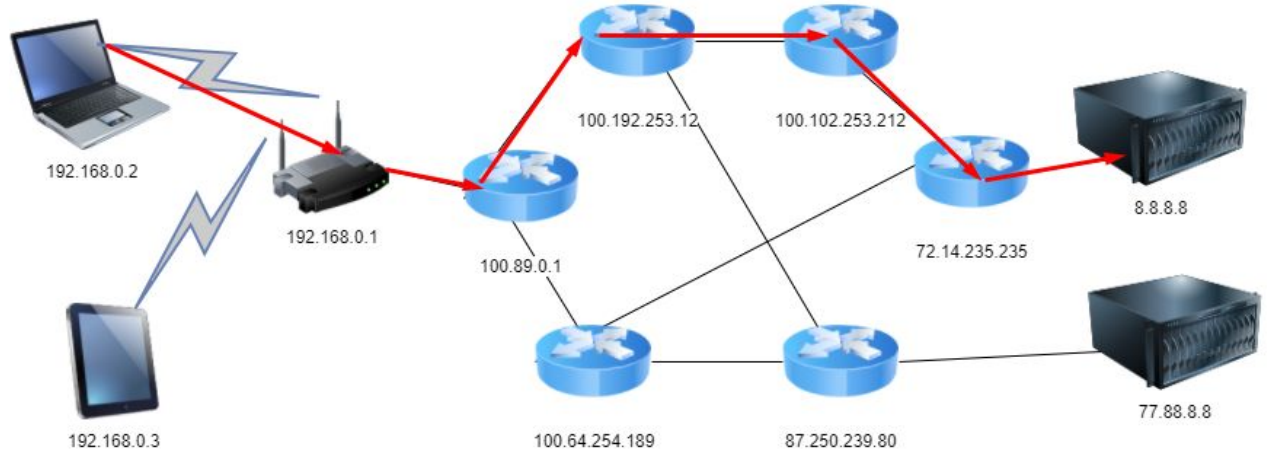
Таким образом, когда мы в браузере вбиваем имя ресурса, например, geekbrains.ru, google.com или yandex.ru, сначала браузер выясняет, что за IP-адрес соответствует данному имени, и, уже используя IP-адрес, подключается к соответствующему серверу в сети Интернет.

Так как Интернет — Сеть сетей, каждое сообщение, следуя от одного хоста (если только они не в одной сети), как правило, проходит не один шлюз, а несколько. При этом на каждом таком промежуточном узле осуществляется процесс определения дальнейшего маршрута, исходя из IP-адреса места назначения. Этот процесс называется маршрутизацией. И если для отправителя и

получателя все выглядит так, как они общаются напрямую (что мы видим из команды ping), на самом деле ситуация более сложная. Примерно ее можно посмотреть с помощью команды tracert (в Windows и командной строке компьютера в Cisco Packet Tracer) или traceroute (в GNU/Linux).



Мы видим, что между компьютером, с которого выполнялась трассировка маршрута, и узлом, связь до которого мы проверяем, находится целых 9 маршрутизаторов (на самом деле точное число сказать невозможно, и позже мы поймем это), при том самый первый — это тот самый шлюз, который указан в настройках tcp/ip-соединения.



Пример движения пакетов от хоста 192.168.0.2 до 8.8.8.8 (очень упрощенная схема).

Использование команды tracert в Windows и командной строке компьютера в Cisco Packet Tracer

```
C:\> tracert 5.255.255.55
```

или

```
C:\> tracert yandex.ru
```

Использование команды traceroute в UNIX-подобных системах:

```
user@host: ~ $ traceroute 5.255.255.55
```

или

```
user@host: ~ $ traceroute yandex.ru
```

Так мы можем использовать в качестве адреса хоста назначение как IP-адрес, так и его доменное имя (имя хоста). Более подробно как работает трассировка маршрута, мы разберем на следующих занятиях.

Для пересылки пакета маршрут выясняется исходя из IP-адреса места назначения.

Не все адреса могут маршрутизироваться. Например, пакет, направленный на адрес 127.0.0.1, никогда не покинет машину. Он будет доставлен другому приложению, находящемуся на данной машине. Стоит отметить, что это тоже нормальный способ использования, например, PHP-скрипт выполняемый на сервере, может обращаться к приложению mysql, находящемуся на той же машине, используя адрес 127.0.0.1. Такой адрес называется локальная петля. На самом деле для таких целей может использоваться любой IP-адрес, начинающийся с 127. Для адреса 127.0.0.1 также имеется имя хоста по умолчанию, localhost.

Поэтому команды

```
ping 127.0.0.1
```

и

```
ping localhost
```

идентичны.

Любое обращение на localhost — обращение на собственную машину.

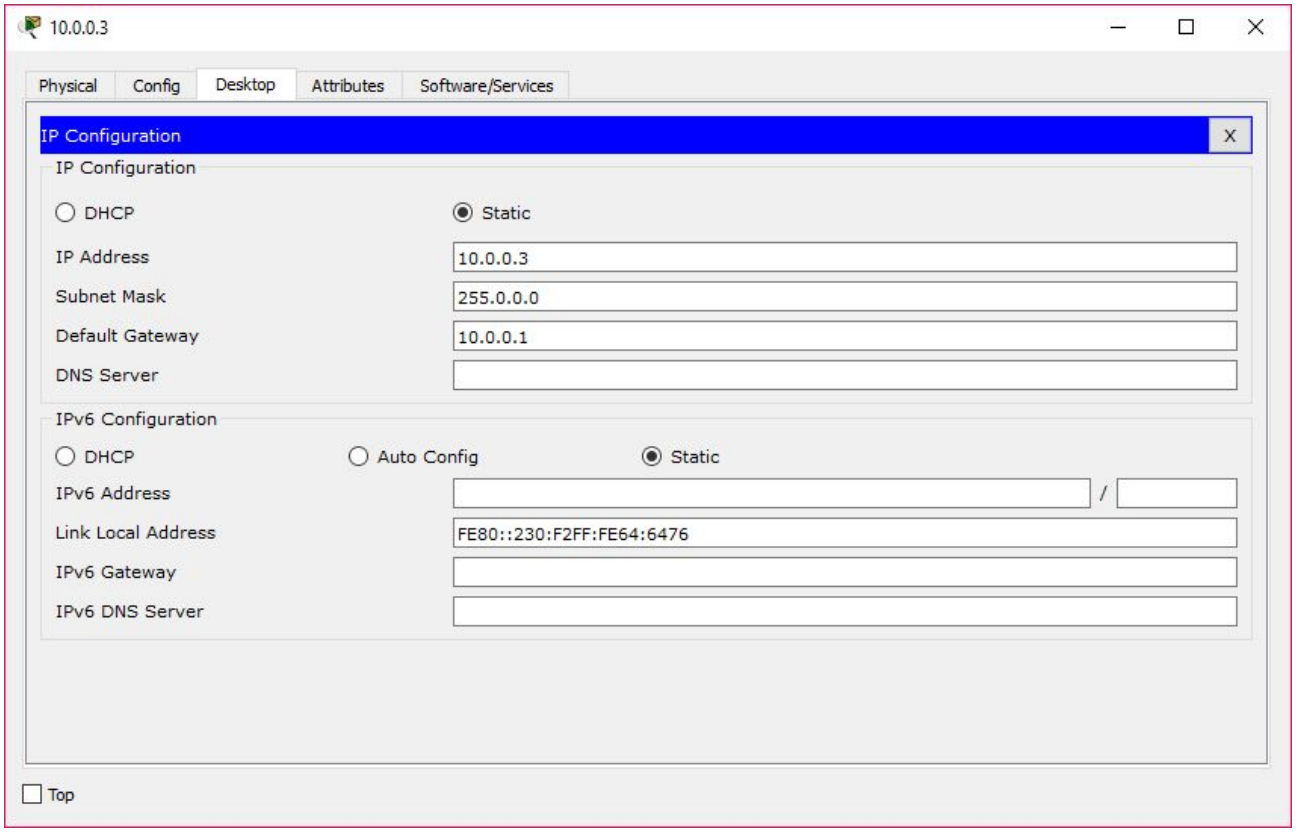
Есть также адреса, которые используются для локальных сетей. Вы можете объединить несколько машин в локальную сеть и использовать адреса из диапазонов таких сетей. Например, 192.168.1.X или 10.X.X.X и машины смогут обмениваться между собой информацией. Но нельзя обратиться на машину с адресом 192.168.1.1 из сети Интернет. Более того, и обратное не было бы верным без специальных средств. Если вы посмотрите настройки TCP/IP-соединения, то можете обнаружить, что, скорее всего, у вас тоже используется адрес из такого диапазона. Но как тогда осуществляется выход в сеть?

Если вы зайдете на какой-нибудь сайт вроде myip.ru, Вы увидите, что адрес, под которым вы видны указанному серверу, не совпадает с вашим собственным IP-адресом, указанном в настройках TCP/IP соединения. Это означает, что Ваш провайдер маскирует ваши адреса, подменяя их своим внешним IP-адресом, запоминая, с какого компьютера был осуществлен какой запрос, и заменяя IP-адрес

получателя со своего на ваш при прохождении ответа. Обладая «серым» IP-адресом при использовании механизма трансляции адресов (NAT), вы можете обращаться к другим серверам, но внешние машины не смогут инициировать соединение к вашей машине, как к серверу.

И в Cisco Packet Tracer и в настройках TCP/IP-соединения вы можете увидеть, что есть два способа настроить параметры TCP/IP (IP-адрес и маска Вашего компьютера, IP-адрес шлюза и DNS-сервер) — прописать вручную (статически) или поставить галочку у параметра — настроить динамически. Во втором случае компьютер сам узнает у сети нужные параметры, если в сети имеется соответствующий сервер (обычно он находится на той же машине, что и шлюз, но не обязательно).

Такой сервер называется DHCP-сервером, а DHCP — Dynamic Host Configuration Protocol — протокол динамической конфигурации узла. Мы его еще будем изучать.



Хост настроен статически. DNS-сервер не указан. Если выбрать DHCP, хост попытается сам найти в сети DHCP-сервер и узнать у него необходимые настройки.

Следующий вопрос, который может возникнуть, каким образом какие сервисы определяют, какое приложение на какой запрос должно реагировать? Ведь один и тот же сервер может отдавать и веб-страницы, и файлы по протоколу http, и почту по протоколу SMTP. Более того, если вы администрируете этот сервер, наверняка вам понадобится доступ к нему по протоколу VNC или ssh.

Соответственно протокол — некий набор правил, который определяет, как то или иное приложение (сервер или клиент) будет взаимодействовать с аналогичным приложением (клиентом или сервером по сети). Для веб-содержимого используются протоколы HTTP и HTTPS (шифрованный HTTP), для работы с файлами FTP и FTPS (шифрованный FTP), для администрирования шифрованный SSH и SFTP (надстройка над SSH, реализующая схожий с FTP доступ к файлам).

Теперь осталось понять, каким образом сервер понимает, какому приложению следует отдать тот или иной пакет. Для этого используются порты. Порт - это число от 0 до 65535, которое используется для идентификации приложения. Существует 65535 TCP-портов, служащих для надежных соединений (с

гарантированной доставкой), и 65535 UDP-портов, для которых надёжное соединение не требуется (без гарантированной доставки). Одно приложение может использовать несколько портов. Например, веб-сервер обычно использует 80 TCP-порт для установки незашифрованного соединения по протоколу HTTP и 443 TCP-порт для установки шифрованного соединения по протоколу HTTPS. Для удалённого администрирования по протоколу SSH (и его составной частью SFTP) используется 22 TCP-порт. Соответственно, если вы купите у хостинга не просто shared-хостинг, а VPS (Virtual Private Server — почти как настоящий сервер в серверной, только в виртуализованной среде), и будете его администрировать (с помощью клиента ssh в Linux или Mac OS X, либо с помощью Putty в Windows) вам понадобится указать: доменное имя (или IP-адрес) вашего сервера, номер порта, по которому запущен на сервере сервер ssh (обычно 22), ваш логин и пароль от операционной системы. Кстати сама система DNS для преобразования доменных имен в IP-адреса (и не только) использует UDP-порт с номером 53.

Вот фактически мы и рассмотрели упрощенно модель TCP/IP.

Она состоит всего из четырех уровней:

- 4 уровень — прикладные протоколы (DNS — 53 UDP порт, HTTP — 80 TCP порт, HTTPS — 443 TCP порт, SSH и SFTP — 22 TCP порт). Реализуют набор правил взаимодействия приложения-клиента и приложения-сервера.
- 3 уровень — транспортные протоколы (UDP — протоколы без подтверждений, TCP — протоколы с установкой соединения и надежной доставкой). Именно транспортные протоколы в заголовках содержат номера портов, позволяя идентифицировать приложения.
- 2 уровень — межсетевой — протокол IP. Заголовок содержит IP-адрес отправителя и IP-адрес получателя, позволяя идентифицировать машину-отправителя и машину-получателя и осуществлять доставку сообщений между разными сетями.
- 1 уровень — уровень сетевых интерфейсов — реализуют доступ к физической среде передачи информации. В качестве примеров можно привести Ethernet и Wi-Fi.

Модель OSI/ISO более сложная, состоит из 7 уровней и не имеет однозначной трактовки применительно к используемым в сети Интернет-протоколам, мы её рассмотрим в дальнейшем.

Сеть как открытая система

Для того чтобы сетевые устройства различных сетевых производителей могли передавать информацию и пользователь мог использовать любой гаджет, существуют общепринятые наборы протоколов и сетевых технологий. Прежде чем переходить к определениям, давайте попробуем разобраться, зачем они нужны. Сетевая передача информации – очень сложная задача. В 70-х годах прошлого века люди активно начали создавать различные сетевые технологии в большинстве своем не совместимые между собой, но решающие схожие задачи. Для того чтобы это упорядочить, были разработаны сетевые модели OSI и DOD, которые декомпозируют сложную задачу на более простые уровни. Каждый используемый сейчас протокол занимается своей задачей и взаимодействует с другими уровнями согласно принятым интерфейсным соглашениям.

Определения

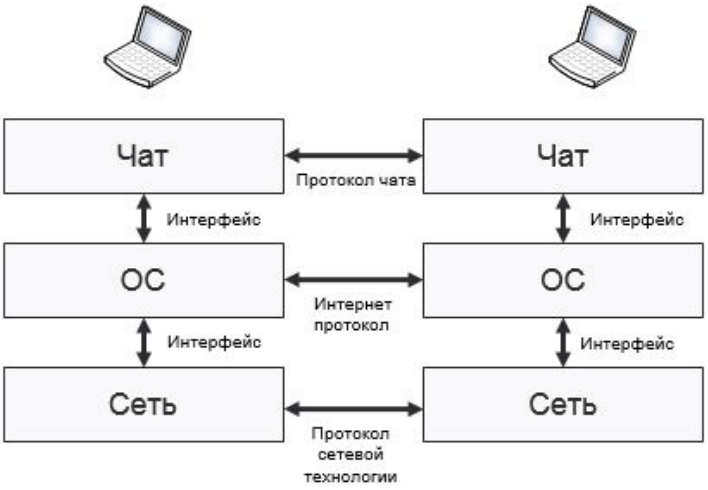
Заучивать определения не нужно, главное — понимать их.

Протокол – это набор семантических и синтаксических правил, определяющий поведение функциональных блоков сети при передаче данных. Некоторые из основных используемых протоколов на данный момент: Ethernet, Wi-Fi, TCP, UDP, HTTP, SSH, SFTP, SMTP.

Сетевые технологии – это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств, достаточный для построения локальной вычислительной сети. Сетевые технологии называют базовыми технологиями или сетевыми архитектурами локальных сетей.

Открытая система (OSI) – это система, использующая соответствующие международные стандарты. По определению комитета IEEE открытая система – это система, реализующая открытые спецификации (стандарты) на интерфейсы, службы и форматы данных.

Декомпозиция задачи или системы - это разбиение её на подзадачи или модули. В задачи декомпозиции входит определение функций модуля и способов взаимодействия с другими компонентами системы. В результате мы получаем простую структуру с возможностью замены отдельных модулей.



На рисунке приведён пример декомпозиции работы сетевого чата. Мы открываем программу, например, mIRC, после чего она подключается к серверу, используя операционную систему. Операционная система, используя сетевой контроллер, обращается в сетевую инфраструктуру и отсылает запрос программы через сеть на другой ПК. На принимающем устройстве происходит обратный процесс. Взаимодействие, которое происходит внутри системы называется интерфейсным, а взаимодействие между системами называется протокольным. Так протокол IRC обращается к системным TCP/IP, а сформированный пакет передаётся в сетевую карту согласно стандартам технологии Ethernet. Сетевое взаимодействие между ПК осуществляется по протоколу сети Ethernet.

Сетевые модели

Существуют две популярные многоуровневые сетевые модели OSI/ISO (ЭМВОС) и TCP/IP (DOD).

Идея многоуровневой модели и стека, как её реализации заключается в том, что логика работы сетевых механизмов изолируется на каждом конкретном уровне, таким образом, что каждый уровень на одной машине работает с аналогичным другой машине, не задумываясь о реализации нижестоящих уровней. Каждый уровень имеет интерфейс на уровень ниже, обращаясь к которому, он реализует собственные возможности. Соответственно, при необходимости организации надежной доставки, приложение открывает TCP-сокет, указывает IP-адрес и порт сервера, и в случае успешного подключения, имеет надежный способ передачи в обе стороны, который использует для реализации прикладного протокола. Соответственно, на прикладном уровне можно не беспокоиться о тройном рукопожатии, управлении окном, ретрансмиссиях и дупликациях. Напротив, на транспортном уровне, например, при реализации протокола TCP, уже не имеет значения, какой прикладной протокол

используется выше, а с другой стороны, и какие протоколы будут использоваться ниже. TCP хорошо работает и через Ethernet и через WiFi, если взять канальный уровень.

Сетевая модель OSI (Open Systems Interconnection Basic Reference Model) — базовая эталонная модель взаимодействия открытых систем, сокр. ЭМВОС; 1978 г) — абстрактная сетевая модель для коммуникаций и разработки сетевых протоколов. Является стандартом ISO (почему часто пишется OSI/ISO) и ГОСТ(ГОСТ Р ИСО/МЭК 7498-1-99).

Модель OSI/ISO является академической попыткой создания универсальной модели взаимодействия систем. На практике модель потерпела поражение перед стеком TCP/IP, ставшим практической реализацией стека сетевых технологий. Тем не менее, модель OSI/ISO иногда бывает удобна для описаний, и нижние 4 уровня применяются для описания сетевых устройств.

OSI/ISO расшифровывается как Open Systems Interconnection Basic Reference model (OSI model), является стандартом ISO: стандарт ISO/IEC 7498-1. Также имеется отечественное соответствие данному стандарту, ЭМВОС – эталонная модель взаимодействия открытых систем, также есть отечественный стандарт ГОСТ Р ИСО/МЭК 7498-1-99). Разработана в 1978 году и состоит из 7 уровней, которые имеют частичное соотношение с действительно применяемыми технологиями.

Данная модель не вполне соответствует используемому стеку технологий TCP/IP, но часто применяется:

- в преподавании сетевых технологий (учебниках, таких как Компьютерные сети Э. Таненбаума и у Олифера);
- государственным заказчиком и государственными структурами/ведомствами;
- нижние уровни (L1–L4) используются производителями сетевого оборудования и сетевыми инженерами (например Cisco);
- не всегда модель TCP/IP удобно использовать, когда речь идет о шифрованных протоколах;
- работодатели на собеседовании могут задавать вопросы о модели OSI/ISO.

Несмотря на то, что стек TCP/IP является стандартом, используемым в современных сетях, модель OSI/ISO также необходимо знать, хотя бы на уровне названия и назначения семи уровней модели OSI/ISO (ЭМВОС).

Рассмотрим назначение уровней:

- Прикладной уровень предоставляет интерфейс к сетевым услугам для прикладного программного обеспечения. Сам же запрашивает интерфейс у уровня представления.
- Уровень представления предназначен для перекодирования и преобразования форматов данных. Запрашивает интерфейс у сеансового уровня.
- Сеансовый уровень необходим для установки и управления сеансами связи. Запрашивает интерфейс у транспортного уровня.
- Транспортный уровень предназначен для передачи данных от отправителя к получателю, и предоставляет возможность управления надежностью передачи. Запрашивает интерфейс у сетевого уровня.
- Сетевой уровень предназначен для глобальной логической адресации узлов и осуществления маршрутизации. Сетевой уровень запрашивает интерфейс у канального уровня.
- Канальный уровень предназначен для взаимодействия устройств через физическую среду передачи, принимая нужные кадры или фреймы (так именуются, блоки данных вместе с заголовками на канальном уровне), отбрасывая ненужные, и осуществляя контроль целостности.
- Физический уровень представлен физическими способами передачи информации. Витая пара (две пары, четыре пары), коаксиал, оптоволокно, радиоканал и т.д.

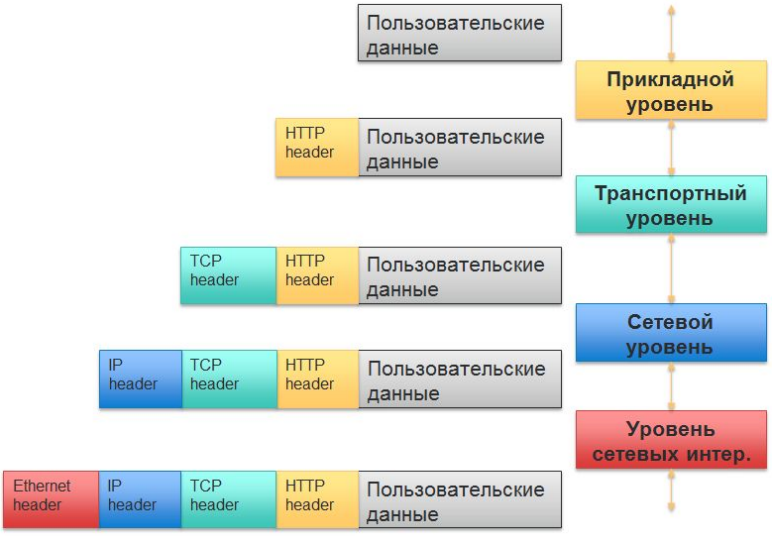
Протоколы TCP/IP не имеют однозначного сопоставления с моделью OSI/ISO.

Иногда протоколы маршрутизации RIP и BGP относятся к прикладному уровню, но чаще, к сетевому. Протокол DNS относится к прикладному, но иногда его относят к сеансовому уровню. Протоколы SSL и TLS чаще всего к уровню представления, но иногда и к сеансовому уровню, аналогично транспортный протокол TCP также обладает сеансовыми чертами. Протокол ARP относят либо к сетевому либо к канальному уровню (на самом деле он имеет промежуточное положение). Консенсус присутствует относительно протоколов канального уровня (IEEE 802.3 Ethernet, IEEE 802;11 Wi-Fi, ITU G.992.1 ADSL, ITU G 991.1 HDSL), основных протоколов сетевого уровня (IP, ICMP, IGMP), протоколов транспортного уровня (TCP, UDP), большинство протоколов относятся к прикладному уровню (FTP/FTPS, SFTP/SSH, HTTP/HTTPS, SMTP, POP3, IMAP4 и т.д. и т.п.). Туннелирующие протоколы часто относят к сеансовому уровню, хотя их местоположение в стеке OSI/ISO не может быть строго определенным из-за дублирования уровней модели OSI/ISO внутри тоннеля и во вне его (PPTP, L2TP, OpenVPN).

Примерно схему расположения протоколов в модели OSI/ISO можно указать следующим образом:

Уровень	Название	Примеры
7 уровень	Прикладной уровень	SOAP, FTP(S), SFTP, POP3(S), DHCP, HTTP(S), SMTP, SSH, IMAP4(S), BOOTP
6 уровень	Уровень представления	ASCII, GZIP, MPEG, XDR, Byte Order, SSL, TLS
5 уровень	Сеансовый уровень	PPTP, L2TP
4 уровень	Транспортный уровень	TCP, UDP, DCCP, SCTP, SPX
3 уровень	Сетевой уровень	IPv4, ICMP, IPv6, ICMPv6, IPX, ARP, RARP
2 уровень	Канальный уровень	Wi-Fi, Ethernet, xDSL
1 уровень	Физический уровень	радиоканал, оптоволокно, витая пара

Вторая модель, которая будет нам более полезна – TCP/IP. Модель DOD или TCP/IP — модель сетевого взаимодействия, разработанная Министерством обороны США, практической реализацией которой является стек протоколов TCP/IP.



Эта модель появилась до модели OSI и независимо от нее при разработке ARPANET, и быстро завоевала популярность благодаря своей простоте. Модель OSI оказалась громоздкой и слишком долго разрабатывалась, поэтому основной используемой сейчас является TCP/IP. Декомпозиция подразумевает интерфейсное и межпротокольное взаимодействие. Оно организовывается с помощью инструмента инкапсуляции.

PDU (Protocol Data Unit) — протокольный блок данных — название блока данных, независимо от используемого уровня модели OSI.

Инкапсуляция — механизм языка программирования, ограничивающий доступ к составляющим объект компонентам (методам и свойствам), делает их приватными, то есть доступными только внутри объекта.

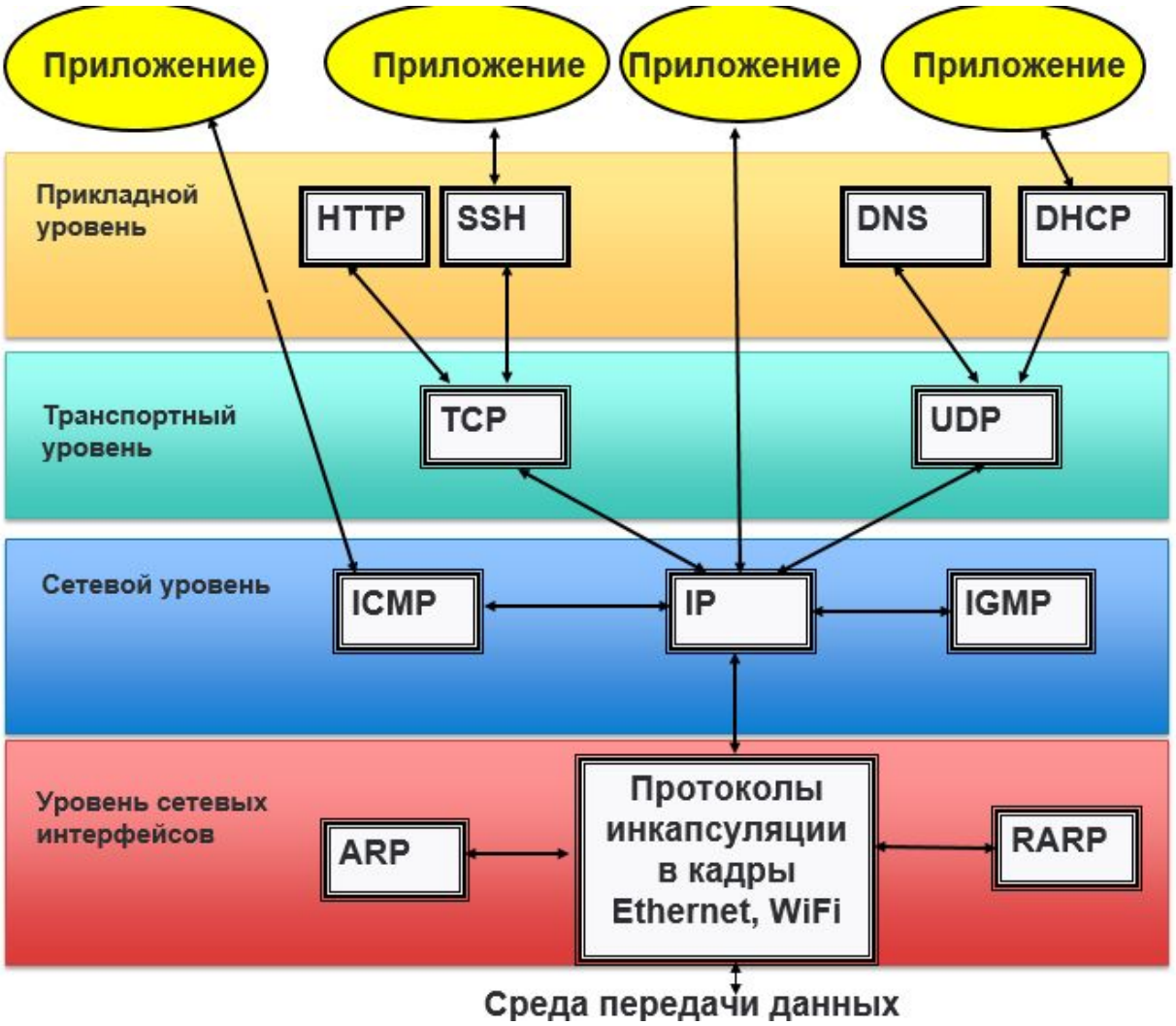
В сетевых технологиях инкапсуляция означает помещение PDU протокола вышестоящего уровня в поле данных PDU протокола нижестоящего уровня, «обертывание» его более низкоуровневым заголовком, упаковка его в PDU нижестоящего уровня. Это можно сравнить с матрешкой, как каждый раз меньшую матрешку (более высокий уровень) вкладывают в оболочку более крупной матрешки (более низкий уровень). Обратный процесс называется декапсуляцией. Подобный механизм позволяет сосредоточиться на протоколах одного уровня, не затрагивая механизмы работы других протоколов. Так разработчики и инженеры сетевого оборудования могут не зависеть от способов взаимодействия прикладных протоколов (используемых приложениями). И наоборот, прикладные протоколы смогут работать через самые разные протоколы нижних уровней (например, как через Wi-Fi, так и через Ethernet). Таким образом обеспечивается прозрачность сетевой передачи.

Стек TCP/IP

Разберем подробнее стек TCP/IP.

Стек протоколов TCP/IP — набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет. Название TCP/IP происходит из двух наиважнейших протоколов семейства — Transmission Control Protocol (TCP) и Internet Protocol (IP), которые были разработаны и описаны первыми в данном стандарте. Также изредка упоминается как модель DOD в связи с историческим происхождением от сети ARPANET из 1970-х годов (под управлением DARPA, Министерства обороны США).

Уровни протоколов TCP/IP расположены по принципу стека (англ. stack, стопка) — это означает, что протокол, располагающийся на уровне выше, работает «поверх» нижнего, используя механизмы инкапсуляции. Например, протокол TCP работает поверх протокола IP.



В самом низу находятся уровень сетевых интерфейсов, объединяющий физический и канальный уровень модели OSI. Пример: интерфейс Ethernet, описывающий передачу данных по коаксиальному кабелю или витой паре. Протоколы этих уровней обычно реализуются на аппаратном уровне, например в сетевой карте компьютера.

Выше идёт межсетевой уровень, соответствующий сетевому уровню модели OSI, и также часто называемый сетевым. На этом уровне работает протокол IP (Internet Protocol), описывающий структуру сети и доставку пакетов. На межсетевом уровне решает вопрос идентификации хостов и маршрутизации. Основной протокол IP (Internet Protocol). Данные снабжаются заголовком, где указываются IP-адреса получателя и отправителя. На сетевом уровне обычно говорят об IP-пакетах или IP-дейтаграммах (реже). Это синонимы. Также к сетевому уровню относятся вспомогательные протоколы, такие как ICMP (для идентификации о сетевых проблемах), IGMP (управление группами широковещательных рассылок, например для потокового вещания IPTV), ARP (преобразование IP-адресов в MAC-адреса), RARP (расширение протокола ARP для назначения машине IP-адреса по ее MAC-адресу, был вытеснен протоколом DHCP), протоколы маршрутизации. При этом ICMP вкладывается в IP-пакет. А некоторые протоколы маршрутизации технически могут быть выполнены как протоколы прикладного уровня (использовать интерфейс транспортного уровня TCP, UDP), тем не менее входить в механизм сетевого уровня. Протокол ARP работает сразу поверх канального уровня и служит для поиска MAC-адресов для искомых IP-адресов своей сети (или шлюза). Протоколы ARP и RARP занимают промежуточное положение между двумя нижними уровнями, иногда их относят к уровню сетевых интерфейсов (канальному уровню).

Ещё выше — транспортный уровень, где находятся протокол TCP (Transmission Control Protocol), использующийся для передачи данных с гарантированной доставкой, и протокол UDP (User Datagram Protocol) — протокол для передачи данных с негарантированной доставкой . Эти протоколы обычно реализуются на уровне операционной системы. На транспортном уровне решается вопрос идентификации приложений и надежной/ненадежной доставки. И TCP и UDP в заголовке содержат двухбайтный порт получателя и порт отправителя. Существует два набора портов, TCP-Порты (в системе именуются STREAM) и UDP-порты (DGRAM). Именно по номеру порта операционная система понимает, какому из работающих приложений должны быть доставлены полученные системой данные.

На самом верху находится множество протоколов прикладного уровня, выполняющих конкретные прикладные задачи. Обычно они программируются в отдельных приложениях. Большинство протоколов, используемых приложениями, являются протоколами прикладного уровня. HTTP, FTP, SSH/SFTP, SMTP, POP3, IMAP4, XMPP, а также DNS, DHCP, NTP, SNMP — являются прикладными протоколами. По большей части любое приложение может реализовать свой протокол, и он будет протоколом прикладного уровня. Прикладные протоколы используют тот или иной протокол транспортного уровня, в зависимости от решаемых задач. Если нужна надёжная передача файлов большого объема (архив, приложения) либо механизм сессии (как ssh), используется TCP. Если нужна отправка коротких сообщений или мультимедиа-потока, не критичного к потерям пакетов — UDP.

IP — протокол, лежащий в основе Интернета, его название так и расшифровывается: Internet Protocol.

В настоящее время используются следующие две версии протокола IP:

IPv6 — активно внедряемая в работу с 2010 года (текущая версия спецификации опубликована в декабре 1998); IP-адрес имеет разрядность 128 бит и записывается в виде восьми 16-битных полей, с использованием шестнадцатеричной системы счисления и с возможностью сокращения двух и более последовательных нулевых полей до ::; пример: 2001:db8:42::1337:cafe;

IPv4 — «классическая» (1981 г.); IP-адрес имеет разрядность 32 бита и записывается в виде четырех десятичных чисел в диапазоне 0 … 255 через точку; пример: 192.0.2.34.

Каждый узел может напрямую связаться только с узлами своей сети (например, подключенными к тому же сегменту Ethernet), для определения которых используется адрес сети — часть IP-адреса, определяемая маской сети). Связь с узлами других сетей осуществляется через промежуточные узлы — маршрутизаторы.

1. Уровень сетевых интерфейсов

Данный уровень модель DOD объединяет в себе 2 нижних уровня модели OSI. Давайте разберём каждый из них отдельно, но запомним, что все их функции относятся к уровню сетевых интерфейсов.

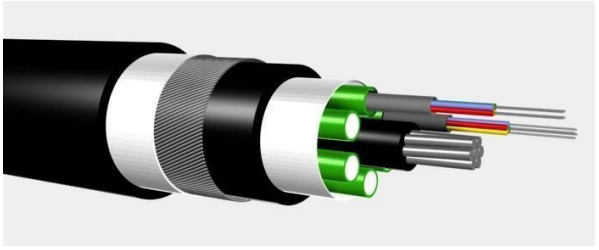
1. Физический уровень (OSI)

Основные задачи уровня:

- передача информации по линиям связи (провода, радио эфир, оптические волокна);
- формирование физической природы сигнала (определение напряжения сигнала, его частоты, длительности и т.д.);
- кодирование и скремблирование передаваемой информации;
- синхронизация работы сетевых интерфейсов приемника и передатчика;

- моделирование полезного сигнала в вид, пригодный для передачи по каналу связи.

Данный уровень реализуется аппаратно, к нему относят провода, коннекторы, сетевые интерфейсы, повторители и прочее оборудование, которое занимается просто передачей сигнала по линии связи.



2. Канальный уровень (OSI)

К данному уровню относятся сетевые технологии, например, Ethernet, Wi-Fi, 4G и т.д.

Протоколы и технологии, обеспечивающие сетевую передачу на данном уровне: STP, VLAN, LACP и т.д. Об особенностях работы сетевых технологий и протоколов мы поговорим на следующих занятиях.

Основные задачи, решаемые протоколами и технологиями на этом сетевом уровне:

- обеспечивается достоверность передачи данных между сетевыми узлами с различными сетевыми топологиями;
- управление доступом к разделяемой среде и решение проблемы коллизий;
- управление средой передачи;
- адресация сетевых узлов;
- разбиение данных сетевого уровня на кадры;
- проверка целостности кадра, путем формирования и проверки контрольной суммы.

Данный уровень реализуется программно-аппаратно и представлен коммутаторами, точками доступа.

2. Сетевой уровень

Данный уровень по функционалу и названию совпадает с моделью OSI.

Основные задачи решаемые на данном уровне:

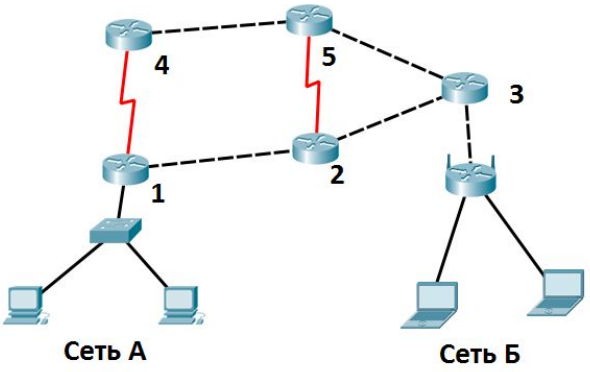
- адресация сетей и узлов в них;
- передача пакетов с данными между узлами в различных сетях;
- нахождение оптимальных маршрутов между сетями;
- предоставление заданного качества обслуживания для информационных потоков.

“Сеть” - совокупность сетевых узлов, подключенных с применением одной сетевой технологии и использующих единую адресацию.

Маршрут - последовательность переходов пакета между узлами\маршрутизаторами на пути к узлу назначения.

К сетевому уровню относятся: маршрутизируемый протокол IP (устаревший протокол IPX) и маршрутизирующие протоколы: RIP, OSPF, BGP.

Основное оборудование: маршрутизаторы, коммутаторы 3 уровня.



Маршруты:

1 – 2 – 3

1 – 2 – 5 – 3

1 – 4 – 5 - 3

1 – 4 – 5 – 2 – 3

3. Транспортный уровень

Совпадает названием уровнем из модели OSI, но носит расширенный функционал (захватывает сеансовый уровень).

В задачи данного уровня входят:

- сегментирование данных, полученных от протоколов прикладного уровня на дейтаграммы, для передачи по сети;
- нумерация и упорядочивание дейтаграмм;
- буферизация дейтаграмм;
- сопоставление и адресация процессов (приложение) и сетевых запросов (создание сокетов);
- управление интенсивностью передачи.

Уровень реализуется программно и представлен протоколами:

- TCP;
- UDP;
- RDP.

4. Прикладной уровень

Данный уровень включает в себя представительный и прикладной уровень модели OSI.

Основные задачи решаемые протоколами данного уровня:

- Передача запросов от клиента к серверу;
- Передача ответов от сервера к клиенту;
- Шифрование данных и идентификация абонента;
- Обеспечение работы сетевых служб.

Протоколы прикладного уровня:

HTTP, DNS, POP3, IMAP, SMTP, SNMP, Telnet,	SSH, FTP, TFTP, RDP, iSCSI, NTP; XMPP.

Сетевая технология Ethernet

Название «Ethernet» (буквально «эфирная сеть» или «среда сети») отражает первоначальный принцип работы этой технологии: всё, передаваемое одним узлом, одновременно принимается всеми остальными (то есть имеется некое сходство с радиовещанием). В настоящее время практически всегда подключение происходит через коммутаторы (switch), так что кадры, отправляемые одним узлом, доходят лишь до адресата (исключение составляют передачи на широковещательный адрес) — это повышает скорость работы и безопасность сети. На рисунке ниже приведен коммутатор.

Стандарты Ethernet определяют проводные соединения и электрические сигналы на физическом уровне, формат кадров и протоколы управления доступом к среде — на канальном уровне модели OSI. Ethernet в основном описывается стандартами IEEE группы 802.3. Ethernet стал самой распространённой технологией ЛВС в середине 1990-х годов, вытеснив такие устаревшие технологии, как ARCNET и Token ring.

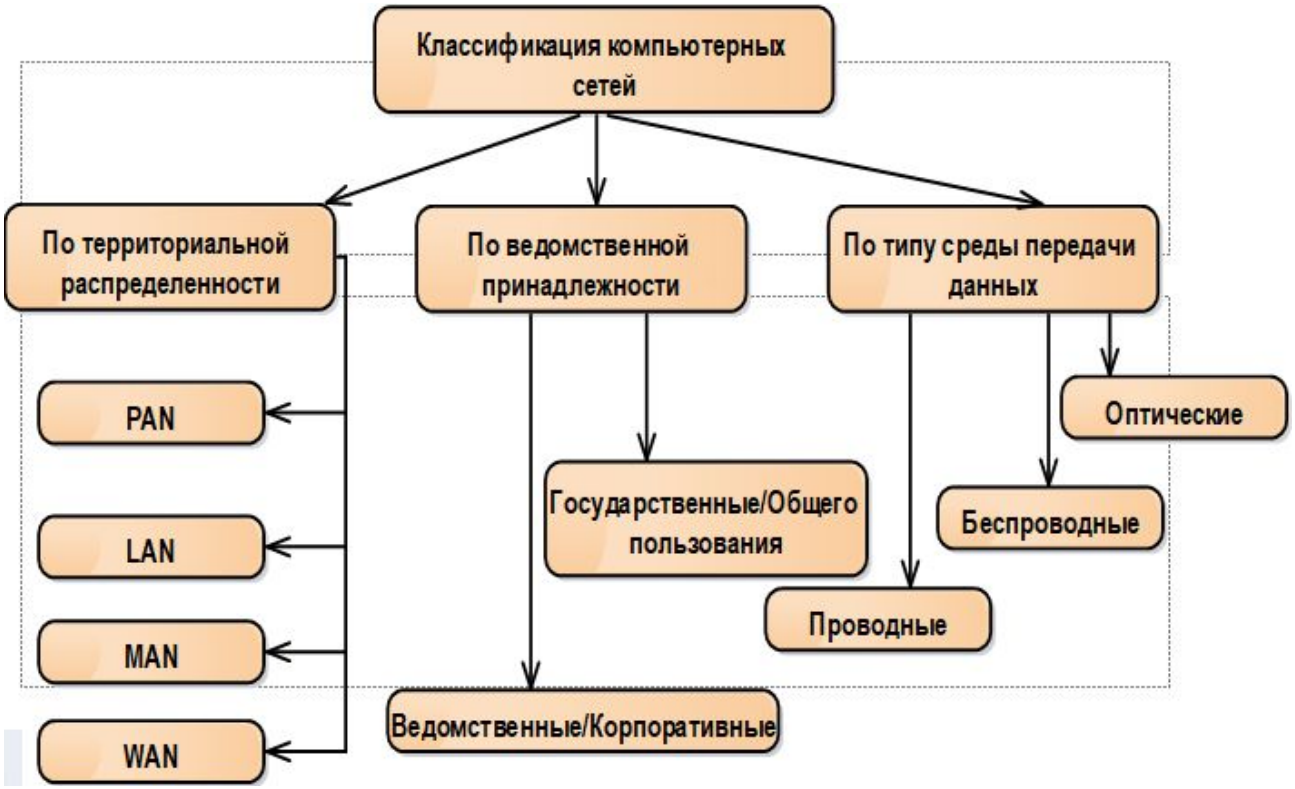
Сферы применения (подключение принтеров, рабочих станций, серверов, пользователей интернет и т.д.)

Скорость технологии Ethernet/FastEthernet/GigabitEthernet со скоростями 10/100/1000 мбит/с соответственно.

Последние принятые стандарты (10G/40G/100G Ethernet), используемые в центрах обработки данных (ЦОД) и магистральных линиях со скоростями 10/40/100 Ггб/с соответственно.

Классификация сетей

Попробуем разобраться в многообразие сетевых технологий и классифицировать их. Можно привести много разных типов классификаций, рассмотрим основные.



Классификация сетей по территориальной распространенности:

- **PAN (Personal Area Network)** — персональная сеть, предназначенная для взаимодействия различных устройств, принадлежащих одному владельцу. Одним из примеров является стандарт IEEE 802.15 WPAN / Bluetooth. WPAN — Wireless PAN. Стек Bluetooth широко применяется для соединения нескольких устройств, таких как смартфоны, SmartTV и т.д.

- **LAN (Local Area Network)** — локальные сети, имеющие замкнутую инфраструктуру до выхода на поставщиков услуг. Термин «LAN» может описывать и маленькую офисную сеть, и сеть уровня большого завода, занимающего несколько сотен гектаров. Локальные сети являются сетями закрытого типа, доступ к ним разрешен только ограниченному кругу пользователей, для которых работа в такой сети непосредственно связана с их профессиональной деятельностью. Как правило, используется немаршрутизируемые серые адреса (10.0.0.0/172.16.0.0/192.168.0.0).

- **MAN (Metropolia Area Network)** – это компьютерная сеть, которая соединяет пользователей с компьютерными ресурсами в географической области или регионе, большей, чем та, которая покрыта даже большой локальной сетью (ЛВС), но меньше, чем область, охваченная глобальной сетью (WAN). Или, если по простому, городские сети. К таковым относят стандарты IEEE 802.6 MAN на основе оптических сетей FDDI (не получил широкого распространения и считается устаревшим) и IEEE 802.16 WMAN (Wireless MAN) и WiMAX, с поддержкой ячеистой (MESH-технологии).

- **WAN (wide area network)** – Представляет собой телекоммуникационную сеть или компьютерную сеть, которая простирается на большое географическое расстояние. Широкополосные сети часто устанавливаются с помощью выделенных телекоммуникационных схем.

Классификация сетей по типу предоставляемых сервисов. На сегодня параллельно продолжают существовать:

- Сети фиксированной связи или телефонные сети общего пользования (ТФоП или PSTN);
- Сети передачи данных (телевизионные/радио сети);
- Компьютерные сети;
- Мобильные сети.

Виды топологий

Сетевая топология — это конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование (коммутаторы, маршрутизаторы), а рёбрам — физические или информационные связи между вершинами.

Сетевая топология может быть:

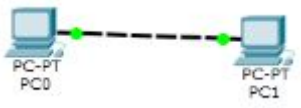
- физической — описывает реальное расположение и связи между узлами сети;
- логической — описывает хождение сигнала в рамках физической топологии;
- информационной — описывает направление потоков информации, передаваемых по сети;
- управления обменом — это принцип передачи права на пользование сетью.

Под топологией (компоновкой, конфигурацией, структурой) компьютерной сети обычно понимается физическое расположение компьютеров сети друг относительно друга и способ соединения их линиями связи. Важно отметить, что понятие топологии относится, прежде всего, к локальным сетям, в которых структуру связей можно легко проследить. В глобальных сетях структура связей обычно скрыта от пользователей и не слишком важна, так как каждый сеанс связи может производиться по собственному пути.

Топология определяет требования к оборудованию, тип используемого кабеля, допустимые и наиболее удобные методы управления обменом, надежность работы, возможности расширения сети. И хотя выбирать топологию пользователю сети приходится нечасто, знать об особенностях основных топологий, их достоинствах и недостатках надо.

Существует базовые топологии сети:

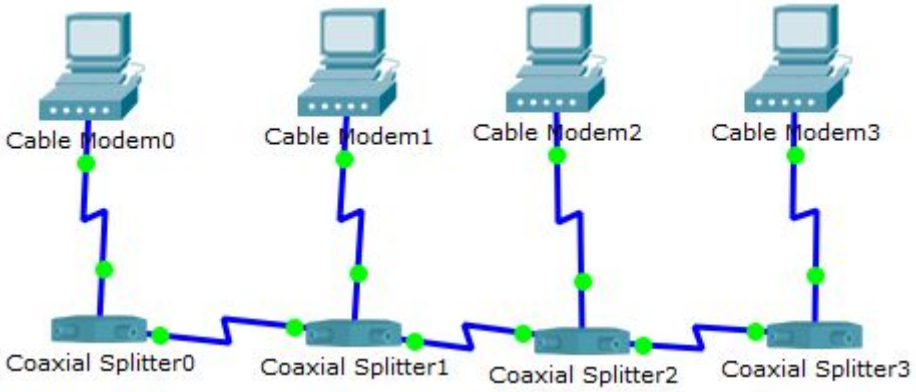
Точка-точка/point-to-point — самая простая топология с выделенной линией связи между двумя конечными точками. Примером такой топологии может быть два компьютера, соединенные кабелем непосредственно.

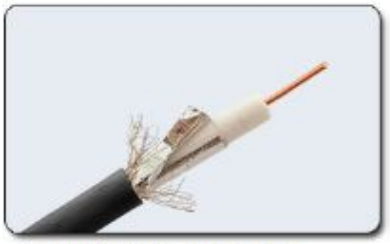


Цепь/circuit —сетевая топология, в которой все узлы сети подключены последовательно друг через друга.

Шина/bus — все компьютеры параллельно подключаются к одной линии связи. Информация от каждого компьютера одновременно передаётся всем остальным компьютерам. Каждый узел слышит

всех, и требуется решать две задачи: определить, нужно ли обработать входящее сообщение или отбросить (оно не нам), и обнаружить или предотвратить коллизии (попытку одновременной передачи сразу двумя или более узлами).



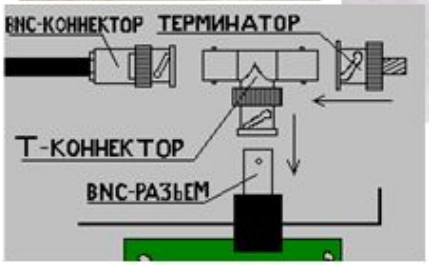


Коаксиальный кабель

Т-коннектор BNC

Т-коннектор BNC

Терминатор - терминирующий резистор 50 Ом

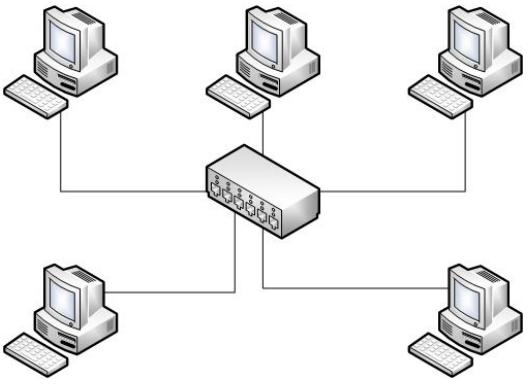


Сетевая карта с BNC-разъемом

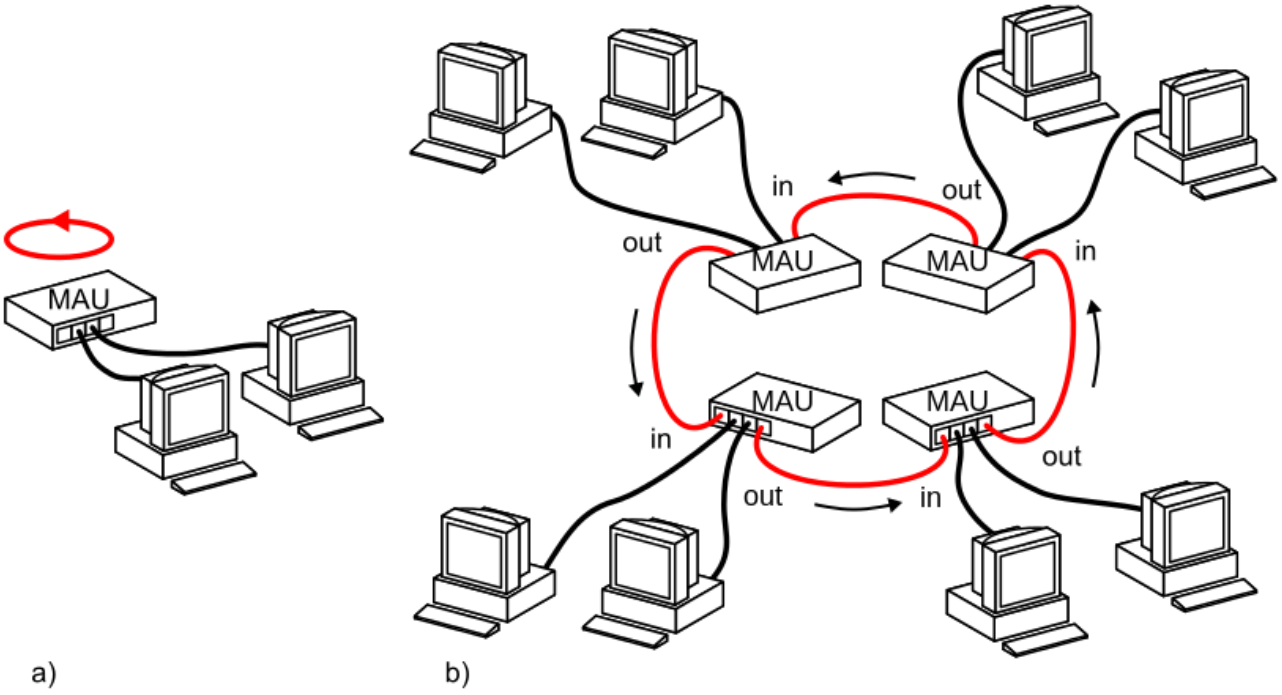
В прошлом топология шина применялась для организации сети Ethernet по коаксиальному кабелю. Но и сейчас технология шина применяется в компьютерной технике: USB-шина, PCI-шина. Также шиной будет топология сети при использовании PLC (Power Line Communications).

.

Звезда/star — к одному центральному узлу присоединяются остальные периферийные компьютеры, причем каждый из них использует отдельную линию связи. Информация от периферийного компьютера передается только центральному узлу, от центрального узла — одному или нескольким периферийным компьютерам.



Кольцо /ring — компьютеры последовательно объединены в кольцо. Передача информации в кольце всегда производится только в одном направлении. Каждый из компьютеров передает информацию только одному компьютеру, следующему в цепочке за ним, а получает информацию только от предыдущего в цепочке компьютера.

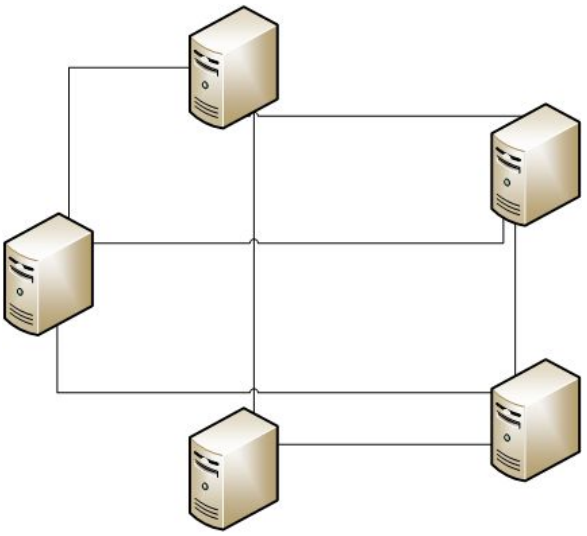


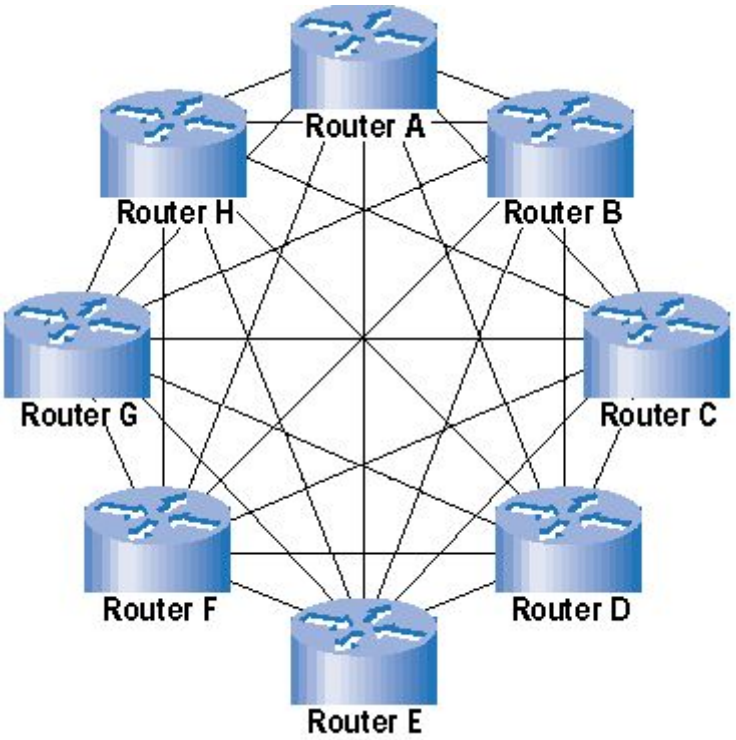
Пример кольцевой топологии на основе технологии Token Ring. Компьютеры подключаются к MAU — Media Access Unit. На рисунке A — два компьютера объединены одним MAU. На рисунке B 8 компьютеров объединены по два благодаря четырем MAU. В данном случае каждый MAU имеет 4 порта. in, два для подключения компьютеров и один out. Информация передается по кругу только в одном направлении (от in к out). MAU по очереди передают токен, тот MAU, у которого в данный момент токен, может передавать, остальные только слушать.

Источник изображения: By Andrew28913 - https://en.wikipedia.org/wiki/File:Token_ring.png, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=33943276

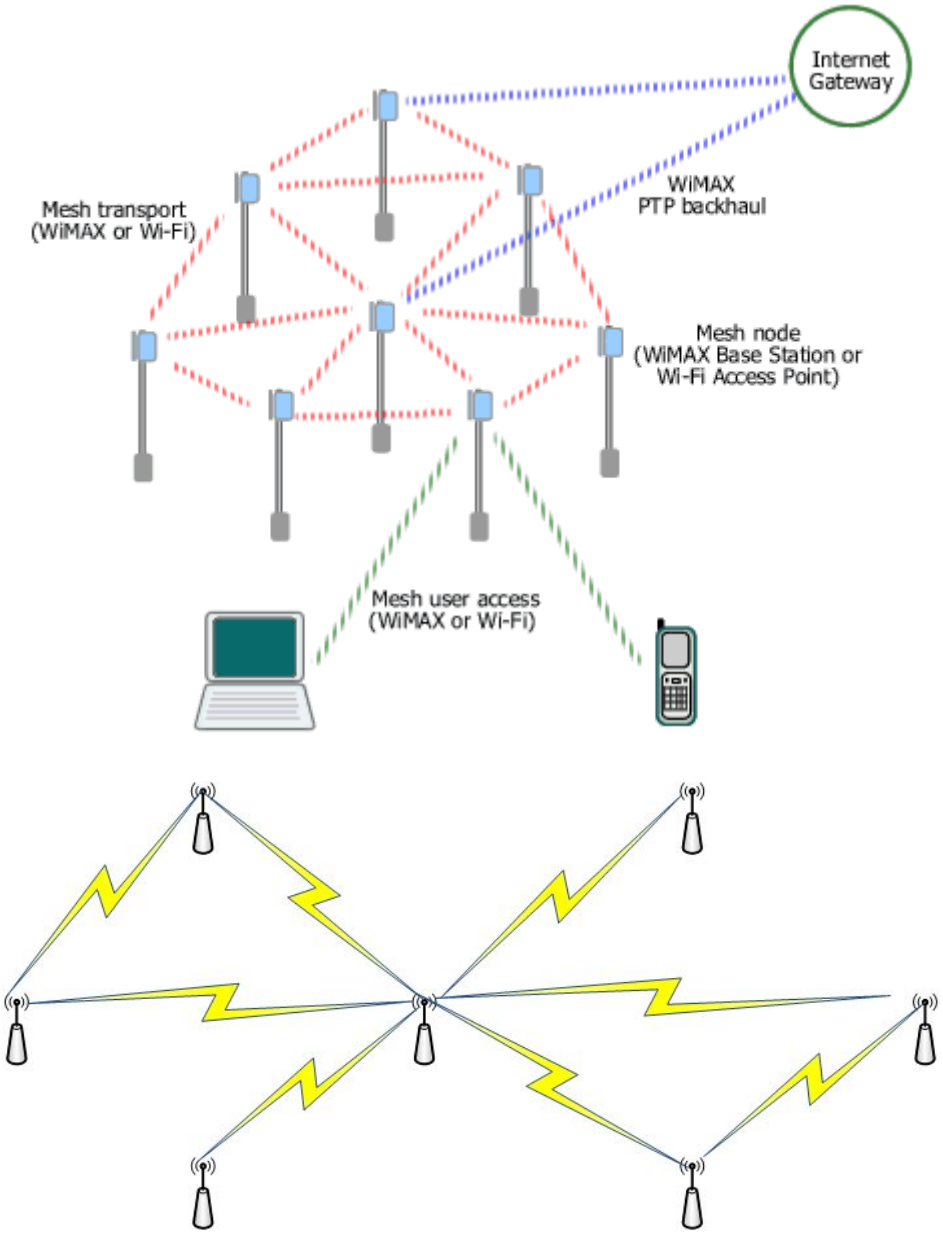
Mesh (ячеистая топология) - сетевая топология компьютерной сети, построенная на принципе ячеек, в которой рабочие станции сети соединяются друг с другом и способны принимать на себя роль коммутатора для остальных участников. Данная организация сети является достаточно сложной в настройке, однако, при такой топологии реализуется высокая отказоустойчивость. Как правило, узлы соединяются по принципу «каждый с каждым». Как правило (но не всегда) используется беспроводное оборудование. Часто применяется на массовых мероприятиях, в военном деле, в спутниковой связи. Применяются особые алгоритмы маршрутизации (ad hoc маршрутизации), такие как AODV или OLSR.

Полносвязная/fully connected mesh topology - сеть, в которой каждый компьютер непосредственно связан со всеми остальными. Однако этот вариант громоздкий и неэффективный, потому что каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров. При этом полносвязные MESH-сети имеют место быть в беспроводной ячеистой топологии, например, когда каждый радиоузел «видит» все остальные узлы.

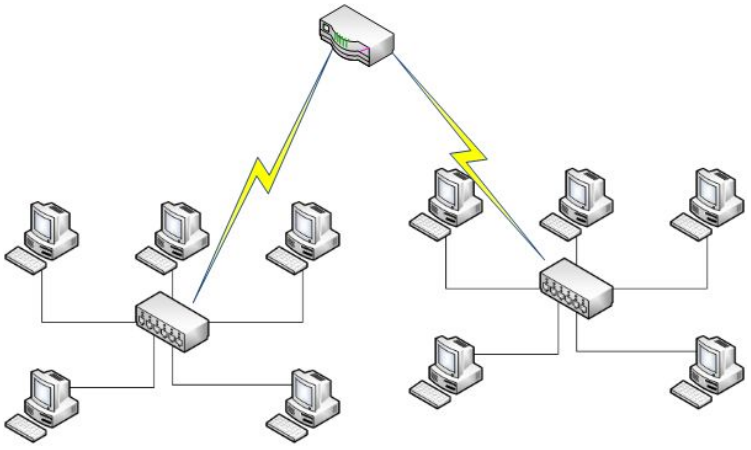




Неполносвязная (ячеистая)/mesh - топология аналогичная полносвязной, но в которой не все компьютеры соединены с каждым. Неполносвязных топологий существует несколько. В них, в отличие от полносвязных, может применяться передача данных не напрямую между компьютерами, а через дополнительные узлы.

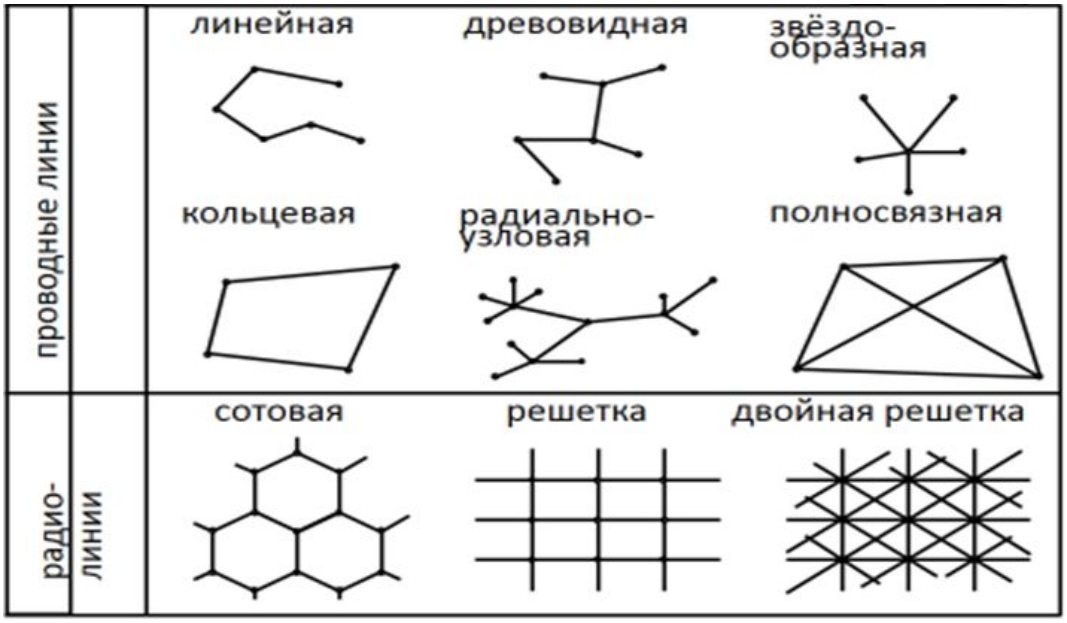


Смешанная - сетевая топология, преобладающая в крупных сетях с произвольными связями между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со смешанной топологией.



И дополнительные (производные):

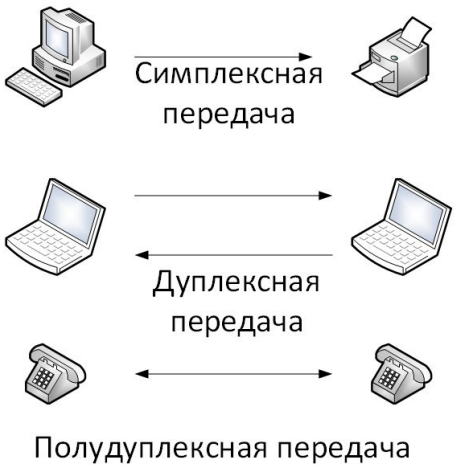
- Двойное кольцо;
- Решётка;
- Дерево;
- Fat Tree;
- Сеть Клоза;
- Снежинка.



От выбора сетевой технологии зависит много факторов:

- топология сети;
- используемое оборудование;
- стоимость создания;
- физическая надежность;
- скорость передачи данных;
- безопасность сети;
- администрирование сети;
- возможность модернизации.

Симплекс, дуплекс, полудуплекс



Симплексная передача — передача в одном направлении. Примером симплексной передачи может быть телевидение или радиовещание.

Полудуплексная передача — передача, схожая с симплексной, но с возможностью передачи информации в оба направления. При этом в один момент времени передача осуществляется только в одном направлении. Таким образом оба абонента взаимодействуют по очереди. Примером может являться разговор по рации. Чтобы перейти из режима «прием» в режим «передача»? необходимо нажать кнопку или использовать тангенту переключения прием-передача.

Дуплексная передача — способ передачи в оба направления, при этом обе стороны могут передавать информацию одновременно. Примером дуплексной передачи может являться телефонная связь.

Адресация в сети

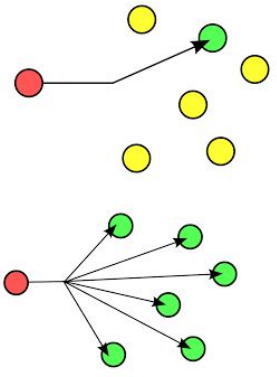
Адрес должен быть уникален в пределах сети.

Желательно автоматизировать назначение адреса.

Иерархическая структура адресации облегчит систематизирование, администрирование и маршрутизацию пакетов.

Адрес должен быть удобен для человека и машины.

Адрес должен иметь минимальное количество бит информации.

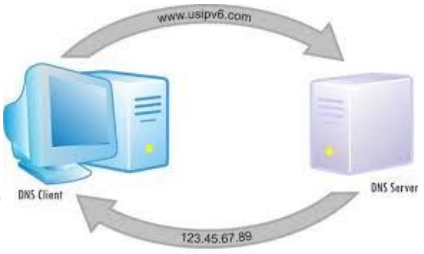


Типы сетевой адресации по количеству:

- Адресация интерфейсов устройств в сети (частные адреса),
- Адресация групп устройств (групповые адреса),
- Адресация всех устройств в сети (широковещательные или сетевые адреса).

Адреса могут быть следующих типов:

- числовые и символьные;
- аппаратные и сетевые;
- адресация может быть плоской или иерархической.



Адресация может быть связана между собой с помощью протоколов разрешения адресов (ARP, DNS).

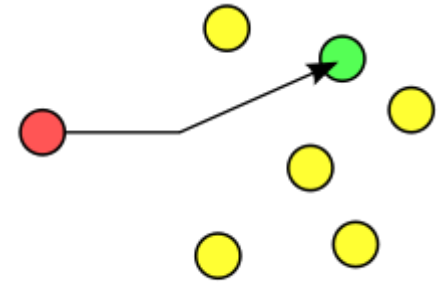
Примерами символьных адресов являются URL, доменные имена, email-адреса. Как правило, символьные адреса используются на прикладном уровне.

На транспортном уровне в качестве адресов можно отметить номера TCP-портов и UDP-портов, позволяющие идентифицировать сетевые приложения на хосте.

Примерами сетевых адресов являются 32-битные IPv4 и 128-битные IPv6-адреса.

В качестве примеров адресов канального уровня можно отметить 48-битные MAC-адреса (Media Access Control, управление доступом к среде) и 64-битные EUI-64. Такие адреса часто называют аппаратными адресами (Hardware Address).

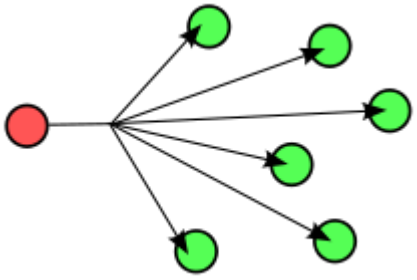
В настоящее время выделяют следующие виды адресации:



Unicast или однонаправленная (односторонняя) передача данных подразумевает под собой передачу пакетов единственному адресату. Это самый частоиспользуемый вариант передачи данных.

Пример использования: ping 192.168.1.1

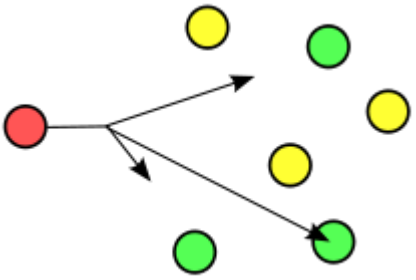
Вам ответит только один узел.



Broadcast или широковещательная передача. Broadcast сообщение получат все адреса в широковещательном домене. Применяется в ARP-запросах, PPPoE для поиска сервера, в DHCP для обнаружения сервера, а также в ICMP.

Пример: ping -b 192.168.1.255

Возможно, вам ответят все соседние узлы.



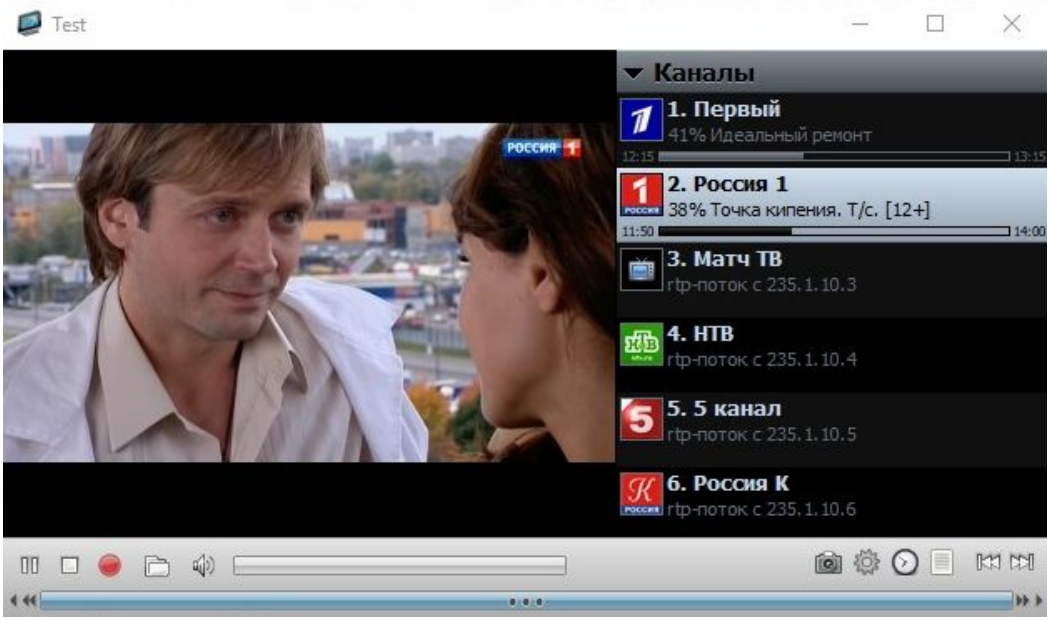
Anycast - отправка сообщения одному (как правило ближайшему) из группы получателей. Чаще всего используется в DNS, когда из нескольких равноправных серверов с одним и тем Anycast IP-адресом вам ответит только один.

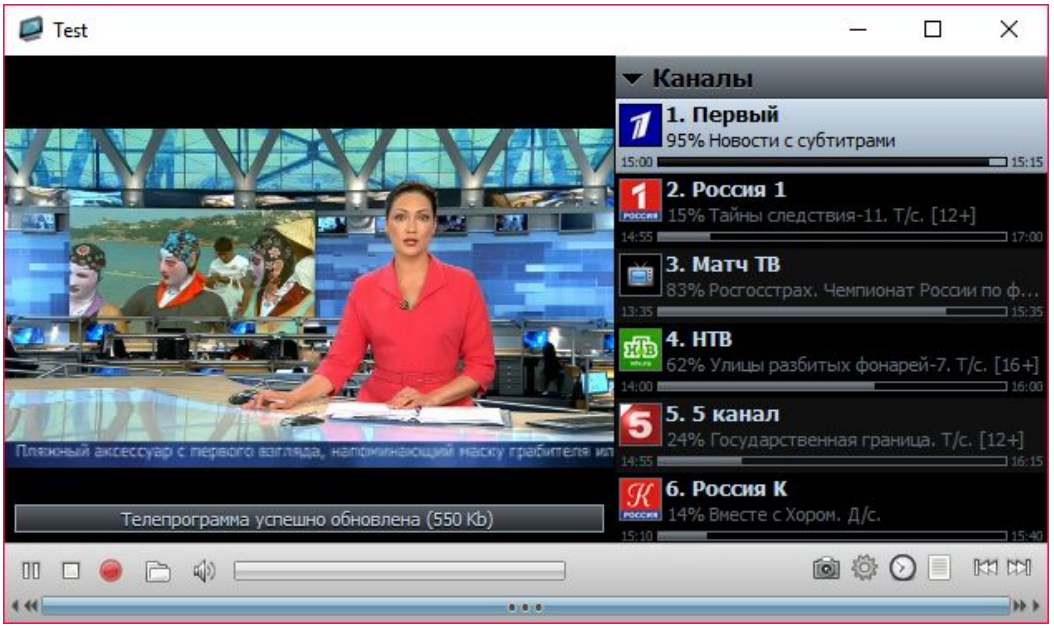
Пример: ping 8.8.8.8

Вам ответит один из множества серверов, окликающихся на 8.8.8.8 адрес.

Multicast – мультивещание, многоадресное вещание. Используется в протоколах маршрутизации и в потоковом вещании. Используются multicat IP-адреса, которые назначаются не отдельным хостам,а группам. Сервер вещает группе, если клиент хочет получать контент, он, с помощью протокола IGMP (Internet Group Management Protocol), подписывается на группу и начинает получать потоковый трафик. Несмотря на то, что сообщение шлется на multicast адрес, у компьютера остаются те же IP-адреса, что и были до подписки.

Пример:

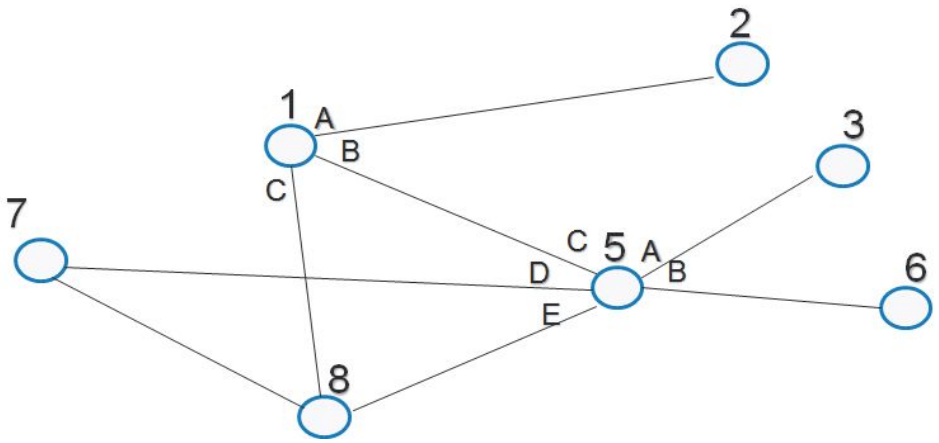




Процесс коммутации

Коммутация — процесс передачи кадров между интерфейсами узла согласно таблице коммутации.

Последовательность сетевых узлов составляющих путь от отправителя к получателю называется **сетевой маршрут**.



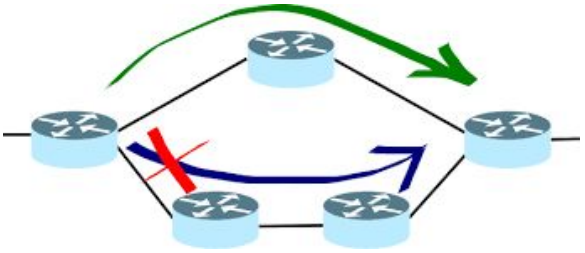
Коммутация в сети заключается в решение нескольких задач:

1. Идентификация **информационного потока** в сети, для которого необходимо проложить путь.
2. Вычисление **маршрута** для потока.
3. Передача маршрутной информации во все узлы сети.
4. **Продвижение** – определение потока и его коммутация на каждом сетевом узле.
5. **Мультиплексирование** сетевых потоков.

Информационный поток (data flow) – последовательность байт (пакетов, кадров), объединенная общими свойствами, по которым можно отделить его от остальной передаваемой информации.

Параметры для выбора сетевого маршрута:

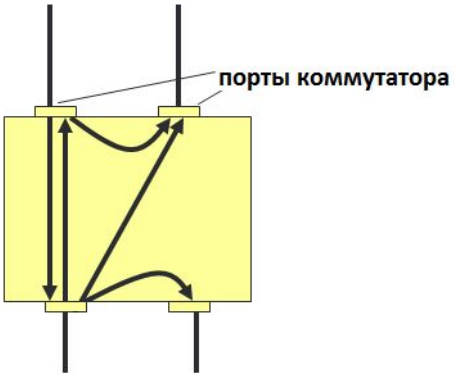
- пропускная способность;
- текущая загрузка канала связи;
- сетевая задержка;
- количество сетевых переходов;
- надежность передачи данных.



Маршрут может быть вычислен автоматически (динамическая маршрутизация) либо задан человеком (статическая маршрутизация).

Процесс коммутации включает мультиплексирование и демультиплексирование потоков

Мультиплексирование



Существует следующие типы коммутации:

- Коммутация пакетов (Ethernet, Wi-Fi);
- Коммутация ячеек (ATM);
- Коммутация каналов (SDH, PDH);
- Коммутация сообщений (телеграф).

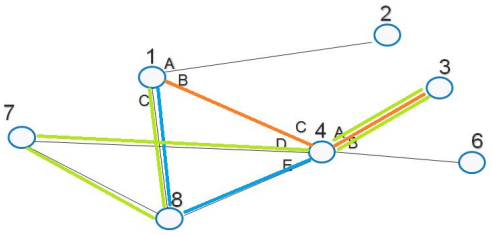
Устройство коммутатора:

- Ручная;
- Механическая;
- Автоматическая.

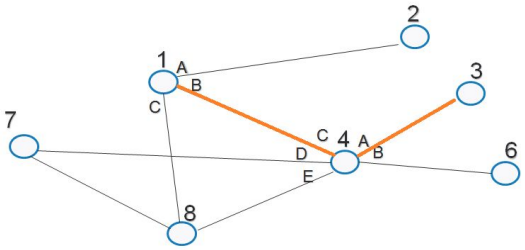
Канальная коммутация	**Пакетная коммутация**
Фиксированная скорость передачи данных	Изменяющаяся скорость передачи данных, могут быть задержки
Фиксированное количество соединений. Все линии могут быть заняты.	Всегда возможна отправка данных.
Трафик передается с одинаковой задержкой.	Трафик может передаваться с различной задержкой, либо вообще быть потерян.
Служебная информация передается только при установке соединения.	Служебная информация содержится в каждом пакете.

Коммутация пакетов в сети может производиться в двух режимах в зависимости от сетевой технологии и настроек оборудования:

1. Дейтаграмма коммутация, используется в сети Ethernet. Особенностью является обработка всех дейтаграмм/пакетов независимо друг от друга.



2. Коммутация виртуальных каналов, при передаче всех пакетов для одного информационного потока используется определенный канал в сети. Пример данной технологии MPLS.



Ethernet

Технология, построенная с использованием идей радиосвязи (Ether - эфир), изначально работала с топологией шина (через коаксиальный кабель).Первые версии Ethernet, 10Base5 (IEEE 802.3, 10 Мб/с, длина сегмента не более 500М, кабель RG-8 «толстый», не использовал Т-коннекторы, кабель “прокусывался” “вампирчиками”) и 10Base2 (IEEE 802.3a, 10 Мб/с, длина сегмента не более 200 м, кабель RG-58 «тонкий» Ethernet, использовались Т-коннекторы). В дальнейшем был осуществлен переход на витую пару, в настоящее время используется витая пара и оптический кабель. Изначально это полудуплексная технология, использующая только один проводник коаксиального кабеля, в настоящее время может использоваться несколько проводников (пар), позволяя реализовать дуплексную передачу.10Base-T — первый стандарт Ethernet, использующий витую пару и позволяющий передавать данные со скоростью до 10 Мб/с. В настоящее время часто используется стандарт 100Base-X (на витой паре 100Base-T), имеющий название Fast Ethernet и позволяющий передавать данные со скоростью до 100Мб/с. Дальнейшее развитие стандарта — гигабит Ethernet (802.3ab по витой паре, 802.ah — по оптоволокну). Существуют и более быстрые стандарты и проекты стандартов.

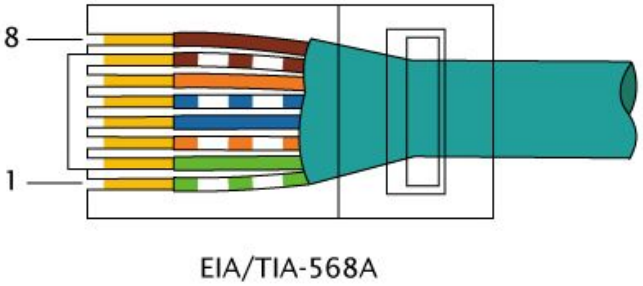
Патч корд

Коммутационный шнур/кабель или патч корд — составная часть СКС (структурированной кабельной системы). Состоит из электрического или оптоволоконного кабеля с коннекторами для подключения одного или нескольких сетевых устройств к другому или пассивному оборудованию.

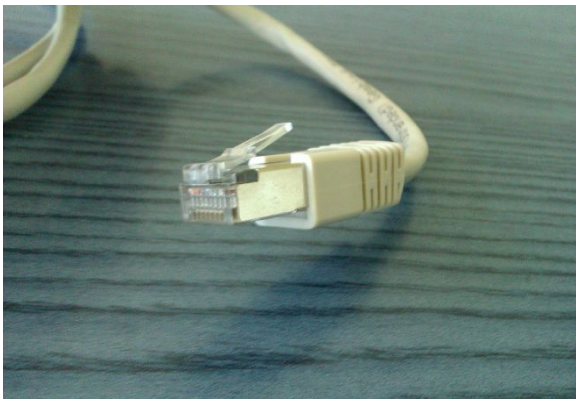
Патчкорд является пассивным сетевым оборудованием и служит для проведения информационных сигналов. Относится к физическому уровню модели OSI.

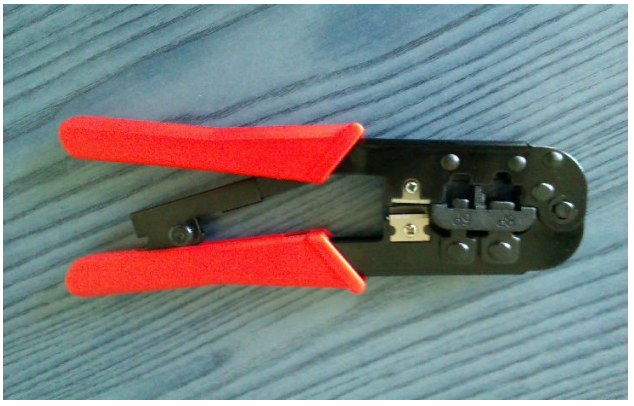
Для сетей Ethernet используют патч-корды от 1 метра (меньшего размера не рекомендуется использовать из-за переотражения сигнала и возникающих вследствие этого помех на полезный

сигнал. Существуют прямые и обратные патч корды. Прямой патч корд имеет одинаковую схему контактов (EIA/TIA-568A или EIA/TIA-568B) с двух сторон. Используется для соединения устройств типа: коммутатор – пк или коммутатор – маршрутизатор. Современное оборудование может автоматически изменять используемые контакты, поэтому вы можете использовать как прямые так и обратные патч корды. Обратные или кроссовер патч корды используются для соединения 2 компьютеров между собой или подключения компьютера к маршрутизатору, либо для соединения двух коммутаторов между собой. Имеют схему обжима (EIA/TIA-568A и EIA/TIA-568B).



EIA/TIA-568A



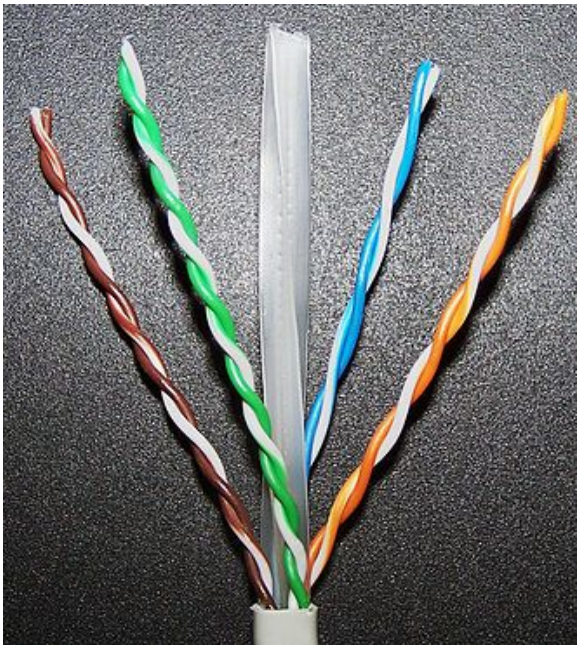


На рисунке приведен экранированный патч-корд, который имеет дополнительную фольгу в своем составе. Экранированные патч-корды стоит применять только в паре с экранированной сетевой розеткой, которая заземлена.

Для обжима коннекторов используется инструмент – кримпер. Для обжима коннектора необходимо зачистить внешнюю оболочку сетевого кабеля, выровнять жилы витой пары, составить их в необходимой последовательности (согласно схеме), затем вставить в коннектор и обжать. Для тестирования работоспособности патч кордов и сетевых розеток применяют кабельные тестеры.

Разъем, используемый в Ethernet? часто называется RJ45 (Registred Jack 45), но то название «народное», ошибочное. Настоящее название данного формата — 8P8C (8 позиций, 8 контактов).

Название «витая пара» указывает на то, что проводники (+ и –) попарно скручены (свиты) между собой для уменьшения помех и наводок.



В FastEthernet используются только две пары, оранжевая и зеленая.

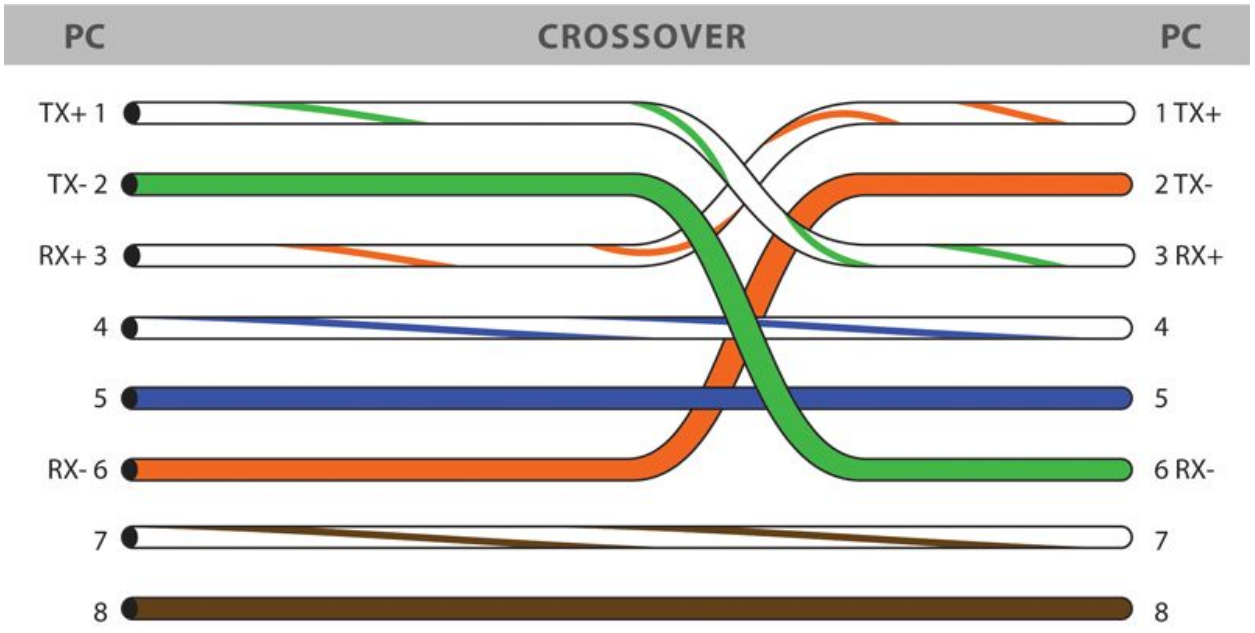
Контакт	Сигнал	Цвет	
		MDI (TIA/EIA-568-B)	MDI-X (TIA/EIA-568-A)
1	Передача +	Белый/оранжевый	Белый/зелёный
2	Передача -	Оранжевый	Зелёный
3	Приём +	Белый/зелёный	Белый/оранжевый
4	Не используется	Синий	Синий
5	Не используется	Белый/синий	Белый/синий
6	Приём -	Зелёный	Оранжевый
7	Не используется	Белый/коричневый	Белый/коричневый
8	Не используется	Коричневый	Коричневый

4 и 5 контакты (синяя пара) оставлены для совместимости с телефонными линиями. Вилку RJ-11 телефона (8P2C) с двумя контактами (использует одну пару) можно подключить в розетку 8P8C. Таким образом, если один кабель подключен и к Ethernet и к телефонной линии, подключение компьютера к розетке даст использование 1,2,3 и 6 контактов, подключение телефона 4 и 5. Для чего используются неиспользуемые пары:

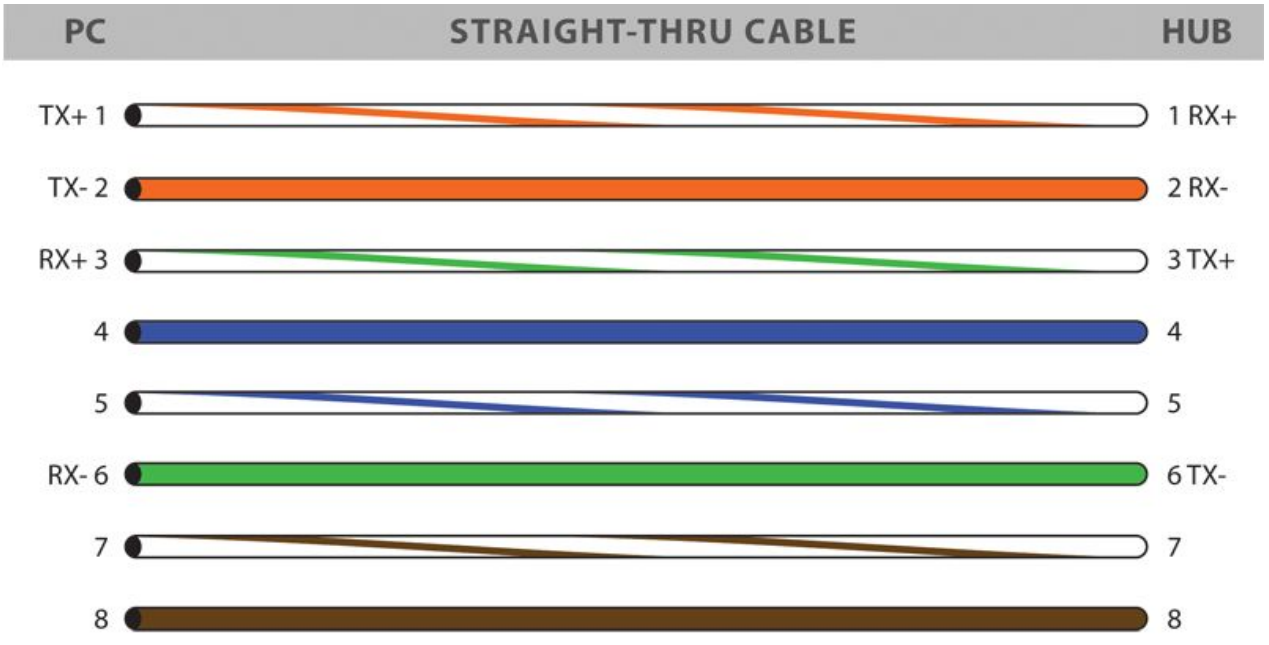
- телефон RJ-11;
- PoE (Power over Ethernet, IEEE 802.3af-2003 и IEEE 802.3at-2009, используется в частности для питания IP-камер используя только один подходящий патч-корд);
- Иногда провайдеры могут по одному кабелю подключить двух абонентов (так называемые «штаны»), две пары используются для подключения одного абонента, две для другого.
- в Гигабит Ethernet используются все четыре пары, причем в дуплексном режиме (одновременно каждая пара является приемо-передающей в один момент времени,

благодаря разности отправленного и детектируемого в это же время сигнала становится возможным получить приходящий сигнал).

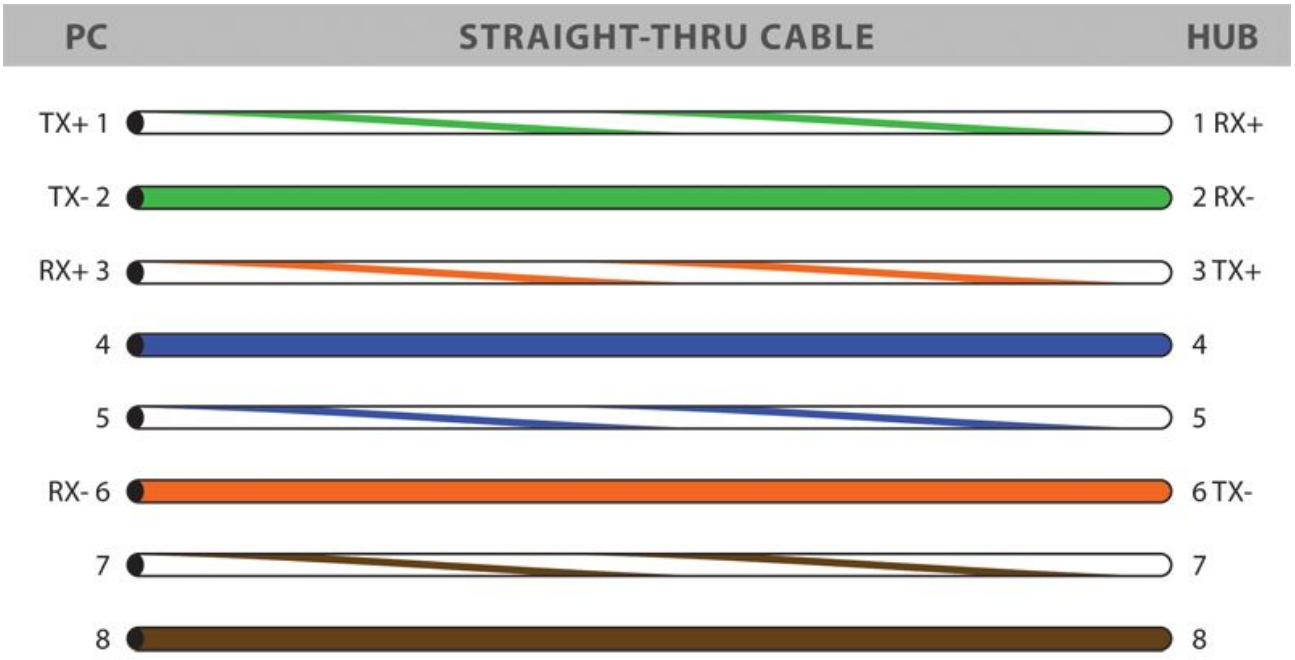
Если два компьютера подключаются непосредственно витой парой, то необходимо, чтобы прием одного коннектора приходился на передачу другого, и наоборот. Также кросскабель используется для соединения двух хабов (концентраторов) между собой.



Для подключения к концентратору или коммутатору используется прямой кабель, так как концентратор или коммутатор сам переворачивает сигналы



Прямой кабель EIA/TIA-568B.

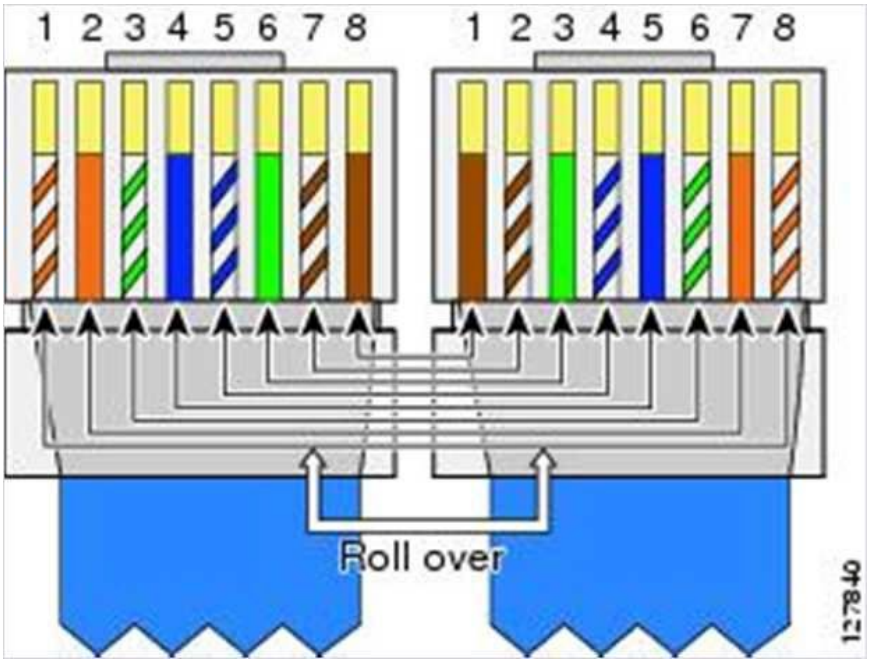


Прямой кабель EIA/TIA-568A.

Почти все современные сетевые устройства поддерживают автоматическое определение MDI и позволяют использовать прямой кабель. Очень редко может потребоваться кросскабель, как правило, при подключении напрямую двух старых компьютеров.

Также для работы с сетевым оборудованием используется консольный кабель, где позиции 1-8 полностью переворачиваются. На самом деле это RS-232 кабель с разъемами 8P8C

Консольный кабель Cisco.



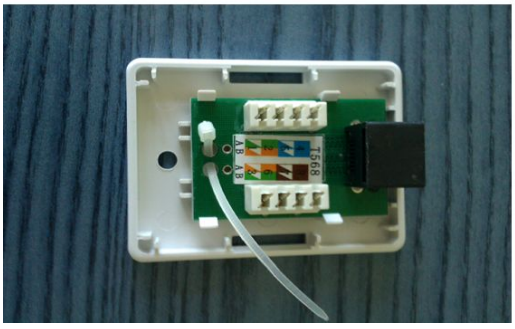
Распиновка консольного кабеля

Сетевая розетка

Сетевая розетка - это разъём для быстро разнимаемого подключения сетевого оборудования (IP-телефон, персональный компьютер и т.п.).

Сетевая розетка является пассивным сетевым оборудованием и служит для проведения информационных сигналов. Относится к физическому уровню модели OSI.

Аналогично сетевому патч-корду имеет 2 схемы обжима EIA/TIA-568A/B. Маркировка на самой розетке позволяет определить необходимую последовательность разделки витой пары. Существуют сетевые розетки на один или два порта. Многопортовые сетевые розетки называют патч-панелями и устанавливают в коммутационных узлах. Розетки с экранированием (они имеют специальные разъемы для крепления заземления, которые необходимо подключать к заземлению). Это необходимо, потому что если этого не сделать, экран в сетевом кабеле не будет выполнять свои функции и будет работать как антенна, собирающая дополнительные внешние наводки.



Патч-панель

Патч-панель - аналог сетевой розетки. Устанавливается внутрь коммутационной стойки или шкафа. Служит для расшивки сетевых кабелей подключенных к сетевым розеткам на рабочих местах. Обеспечивает системному администратору быстрый доступ к сетевым портам. Позволяет оптимизировать инфраструктуру, быстро находить неисправности и проводить диагностику структурированной кабельной системы.

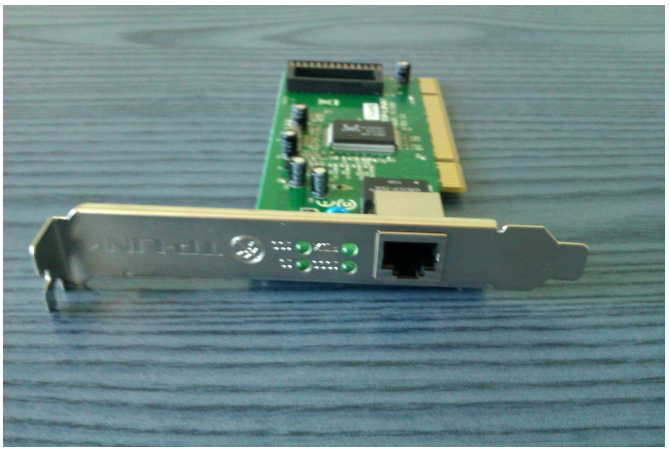
Патч-панель является пассивным сетевым оборудованием и служит для проведения информационных сигналов. Относится к физическому уровню модели OSI.

Сетевые адаптеры

Сетевой адаптер или сетевая карта – это аппаратная плата, устанавливаемая внутрь компьютера или подключаемая через USB-интерфейс. Обеспечивает подключение компьютера к сети.

Сетевая карта является активным сетевым оборудованием и служит для генерации и приема информационных сигналов в сети. Относится к физическому и канальному уровню модели OSI. Компьютер осуществляет взаимодействие с сетевой картой через драйвер, который управляет работой сетевого контроллера.

Сетевая карта обеспечивает подключение к одной из сетевых технологий или стеку, например, приведенная на рисунки ниже карта обеспечивает работу Ethernet технологии на скоростях 10/100/1000 Мбит/с. Аналогично беспроводные Wi-Fi-карты могут работать со стеком технологий например 802.11 g/n/ac и т.п.



Повторитель

Повторитель (repeater) — аппаратное устройство, дублирующее входной сигнал. Повторитель является активным сетевым оборудованием и служит для увеличения расстояния передачи информационных сигналов, путем их повторной передачи. Относится к физическому уровню модели OSI.

Повторители широко используются в оптических линиях связи для увеличения длины линии связи. Аналогично повторитель используются в беспроводных сетях, где выполняется приём исходного сигнала, его обработка, а затем повторная передача с исходным уровнем. Необходимость

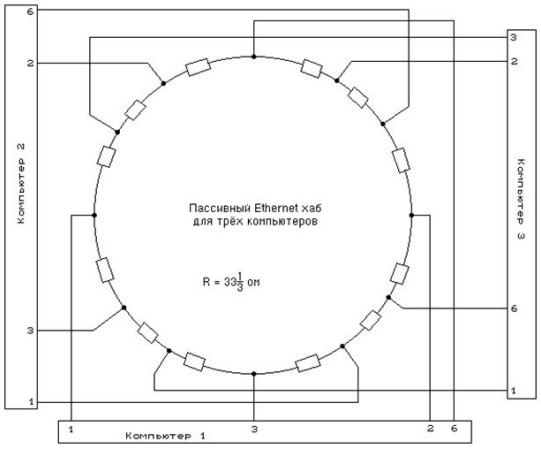
применения повторителя связана с особенностями распространения волн. Информационный сигнал, передаваемый волной, постепенно затухает, повторитель выполняет восстановление исходной формы сигнала и его повторную передачу. Беспроводные точки доступа, настроенные на усиление сигнала определенной сети, также могут называться повторителями.

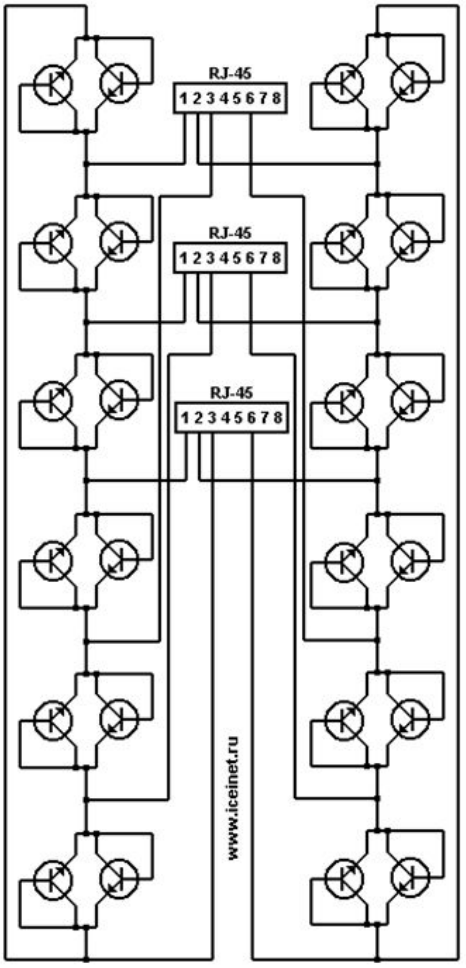
Повторитель обычно имеет 2 одинаковых интерфейса\порта.

Концентратор

Сетевой концентратор или хаб (hub/центр) – это активное сетевое устройство, выполняющее функцию объединения компьютеров в сети.

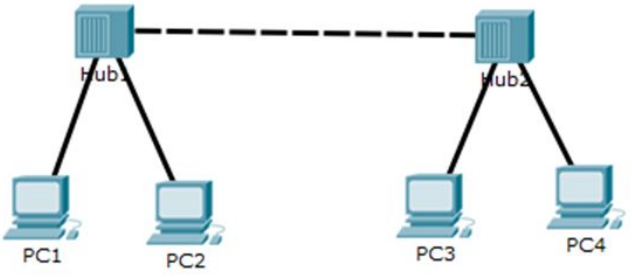
Концентратор является активным сетевым оборудованием и служит для увеличения расстояния передачи информационных сигналов, путем их повторной передачи, а также физического объединения устройств по топологии звезда или дерево. Относится к физическому уровню модели OSI.





На рисунке выше изображены концентратор и два варианта электрической схемы концентратора. Концентратор – это очень простое устройство, которое тиражирует пришедший в порт сигнал во все остальные порты. В нем нет программируемой логики, концентратор не анализирует заголовки канального уровня.

Наиболее часто встречающиеся устройство этого типа работают с технологией USB. Концентраторы в сетях Ethernet вытеснены в силу ряда причин, мешающих эффективной работе сети. В настоящее время вытеснены сетевыми коммутаторами.



Сетевые концентраторы Ethernet работают на скорости 10 Мбит/с, обеспечивая коммутацию по топологии общая шина. Таким образом устройства, объединенные сетевым концентратором, объединены в домен коллизии.

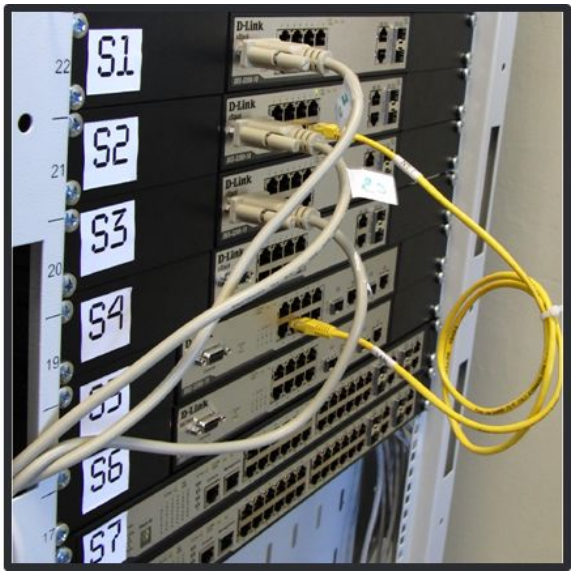
Коллизия - наложение двух и более кадров (сигналов) от компьютеров/абонентов, передающих одновременно из-за наличия задержки распространения сигнала по сети или в связи с неисправностью аппаратуры. Сетевая технология предусматривает защиту от возникновения коллизий путем использования произвольных интервалов на передачу. Кроме того, устройство не может осуществлять передачу, пока сеть занята. Таким образом, через концентратор одновременно возможно передача только от одного абонента в один момент времени.

В связи с особенностями технологии Ethernet, существует целый ряд ограничений на количество используемых последовательно концентраторов, но поскольку данные устройства сейчас практически не находят применения, мы не будем их рассматривать.

Коммутаторы (2,2+,3,3+)

Сетевой коммутатор — это устройство, используемое для коммутации сетевых узлов компьютерной сети в пределах одного или нескольких сетевых сегментов. Коммутатор является активным сетевым оборудованием, широко используется в современных сетях. Коммутатор работает на физическом и канальном уровне модели OSI. Отдельное модели коммутаторов могут работать на сетевом уровне.

Коммутатор анализирует заголовки канального уровня, и сопоставляет MAC-адреса устройств и собственные номера портов, через которые те соединены. Кроме того, коммутатор имеет буфер, в котором накапливаются кадры, если они поступают в коммутатор одновременно для поочередной обработки. Такая технология получила название Storing Forward.

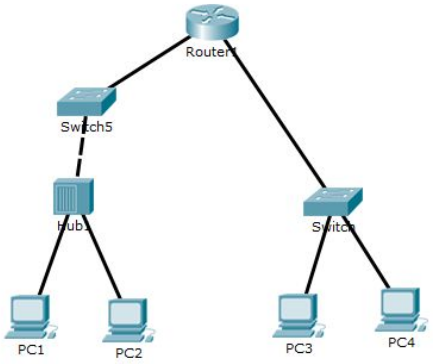


Коммутаторы могут быть неуправляемые, настраиваемые и управляемые. Управляемые коммутаторы имеют дополнительный консольный порт, подключение к которому осуществляется через COM(RS-232) интерфейс.

Маршрутизаторы

Маршрутизатор (также роутер или рутер в зависимости от англ амер./брит.) – это специализированное устройство (может быть компьютером), имеющее два или больше сетевых интерфейсов (плат) и маршрутизирующее пакеты с данными между сетевыми сегментами. Маршрутизатор позволяет объединять сети с различной архитектурой, выполняя функции шлюза. Для определения пути передачи пакета маршрутизатор использует информацию о топологии, состояния каналов и правила, которые могут быть заданы сетевым администратором. Процесс маршрутизации нами будет рассмотрен отдельно в следующих темах.

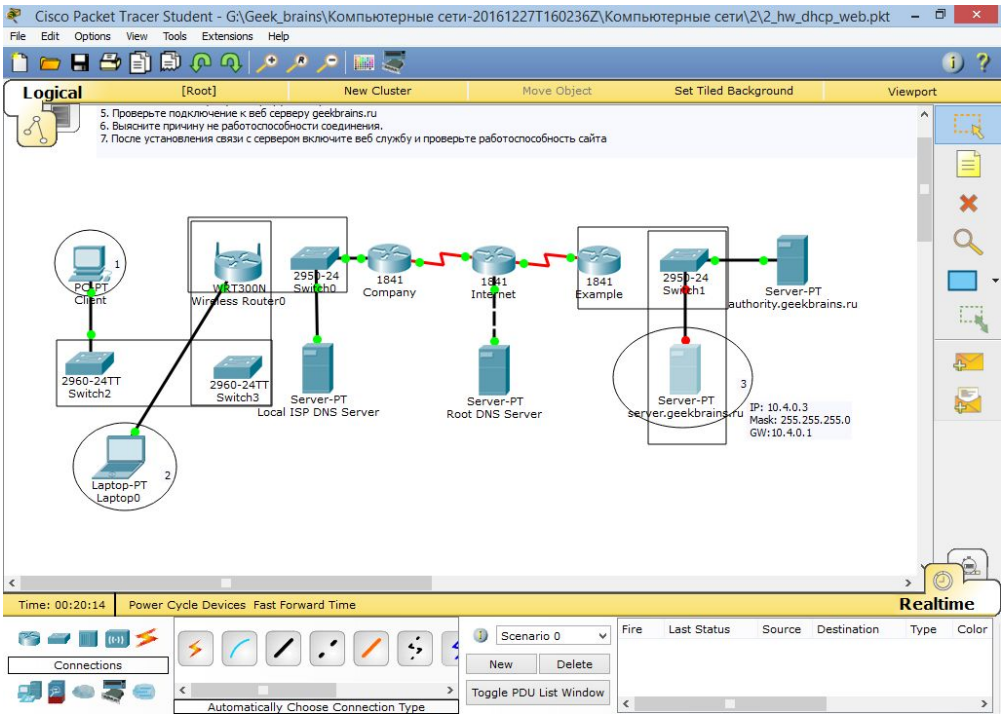
Маршрутизаторы принимают кадры, но обрабатывают их уже на сетевом уровне и работают с пакетом, помещенным в кадр, в отличие от коммутаторов или хабов, которые оперируют кадрами и работают на втором и первом уровне модели OSI.



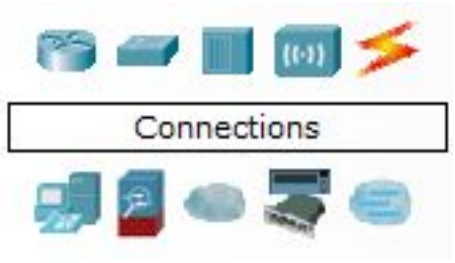
Домашнее устройство, называемое ошибочно роутером, является комбайном из точки доступа, коммутатора и маршрутизатора.

Packet Tracer

Программа позволяет симулировать простые сети. Основное окно программы показано на рисунке ниже.



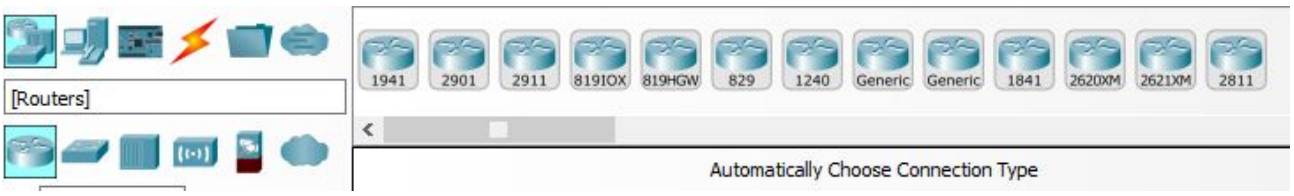
В основном окне производится моделирование сети. Для добавления элементов можно воспользоваться блоком с типами устройств.



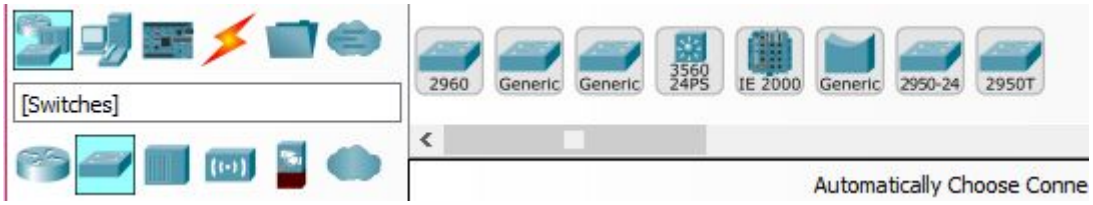
Справа от данной области расположены сами объекты сети.

Доступные типы объектов:

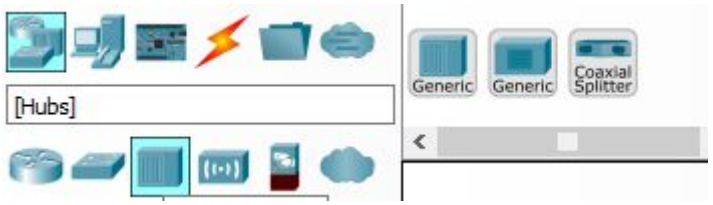
Routers — маршрутизаторы, доступны маршрутизаторы Cisco различных серий, а также обобщенный (Generic) маршрутизатор, выполняющий общие для всех маршрутизаторов функции.



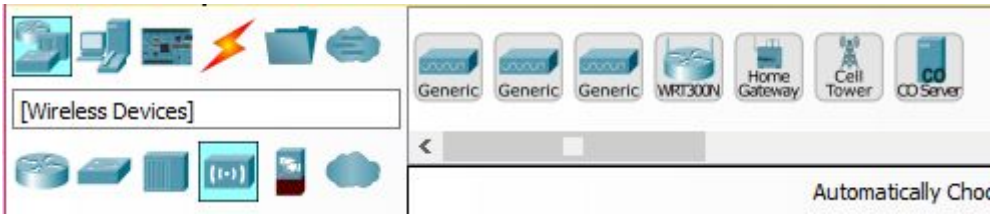
Switches — коммутаторы, доступны управляемые коммутаторы второго уровня разных серий.



Hubs — концентраторы.



Wireless Devices — беспроводные устройства. Generic — обобщенная точка доступа, Linksys — беспроводной маршрутизатор с интегрированными службами Linksys.



Connection — различные виды соединений между устройствами:

Auto — автоматическое определение типа соединения (автоматически определяется наилучший способ соединения устройств), но не всегда выбирает правильный порт и тип кабеля

Console — соединение при помощи консольного кабеля (COM порт на ПК и вход Console на устройствах Cisco),

Copper Straight-Through — соединение при помощи кабеля типа витая пара прямое.

 Copper Cross-Over — соединение при помощи кабеля типа витая пара перекрестное,

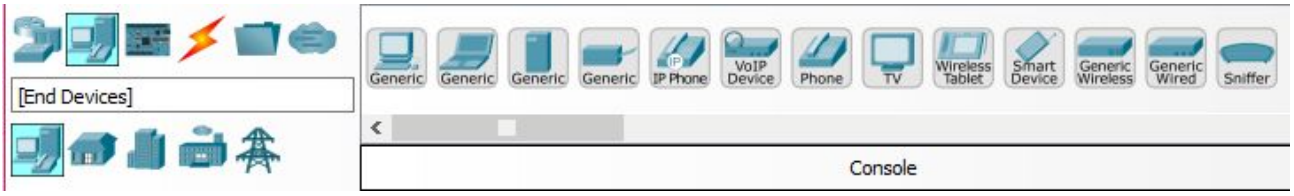
 Fiber — соединение при помощи волоконно-оптической линии связи (ВОЛС),

 Phone — соединение при помощи телефонной линии,

 Coaxial — соединение при помощи коаксиального кабеля,

 Serial DCE и Serial DTE — последовательные (RS-232) каналы связи.

End Devices — конечные устройства:



 PC-PT — персональный компьютер.

 Laptop-PT — мобильный компьютер (ноутбук).

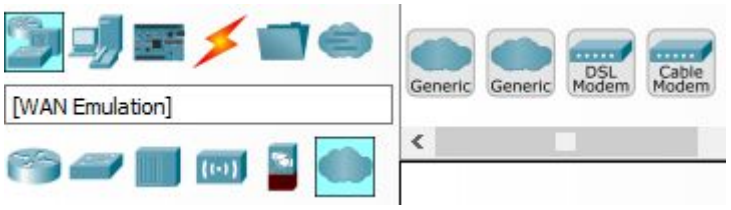
 Sniffer — устройство для перехвата трафика.

 Server-PT — серверная станция.

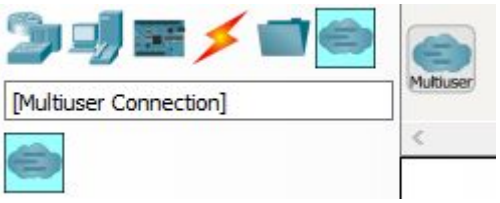
Security — брандмауэры, осуществляющие защиту сети от проникновения.

Wan Emulation — эмуляция глобальной сети, эмуляция сети в общем (Cloud) или модемной связи (DSL Modem, Cable Modem).

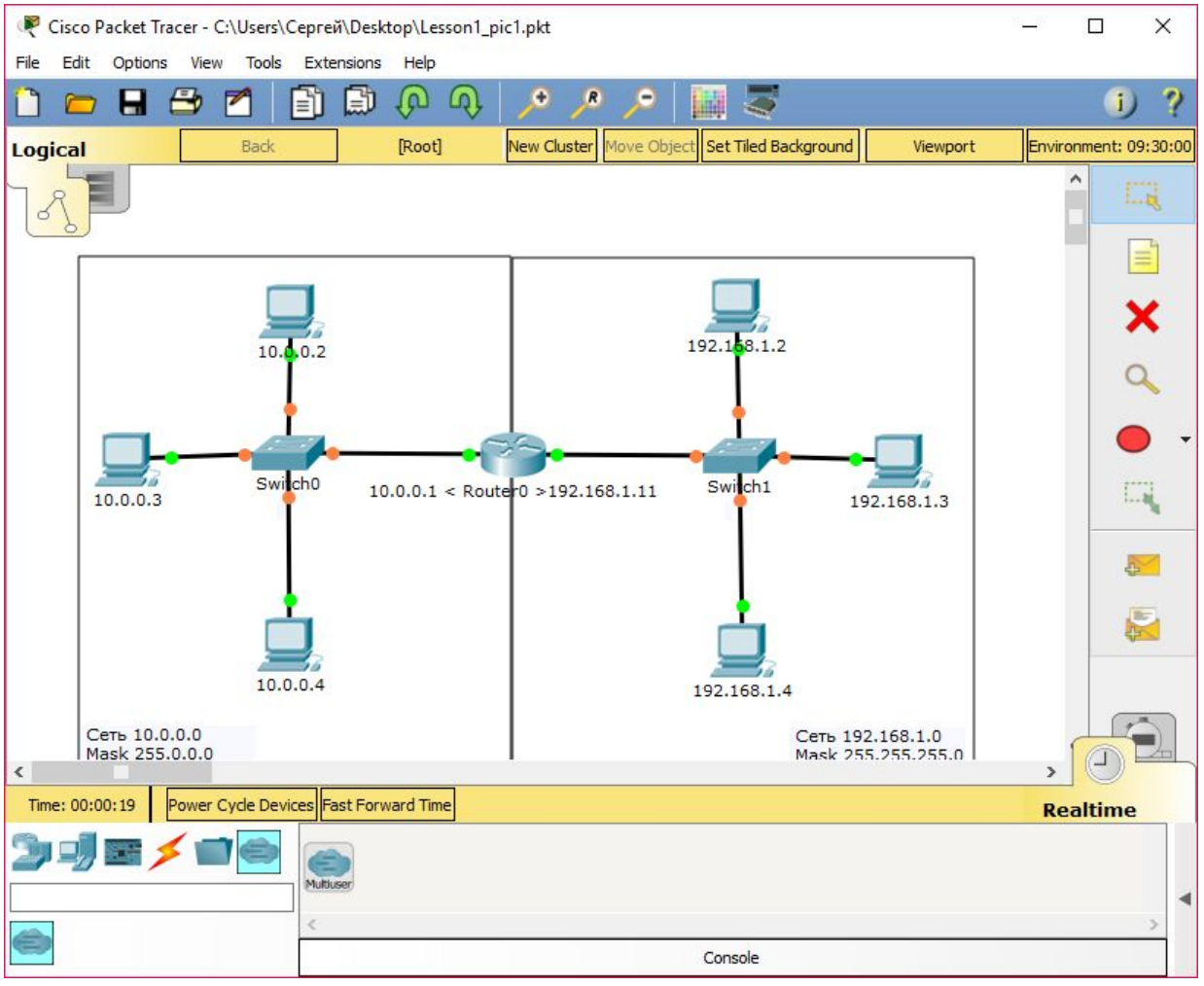
Custom Made Devices — позволяет производить конфигурацию устройств на физическом уровне (добавлять новые сетевые платы, платы расширения и пр.).



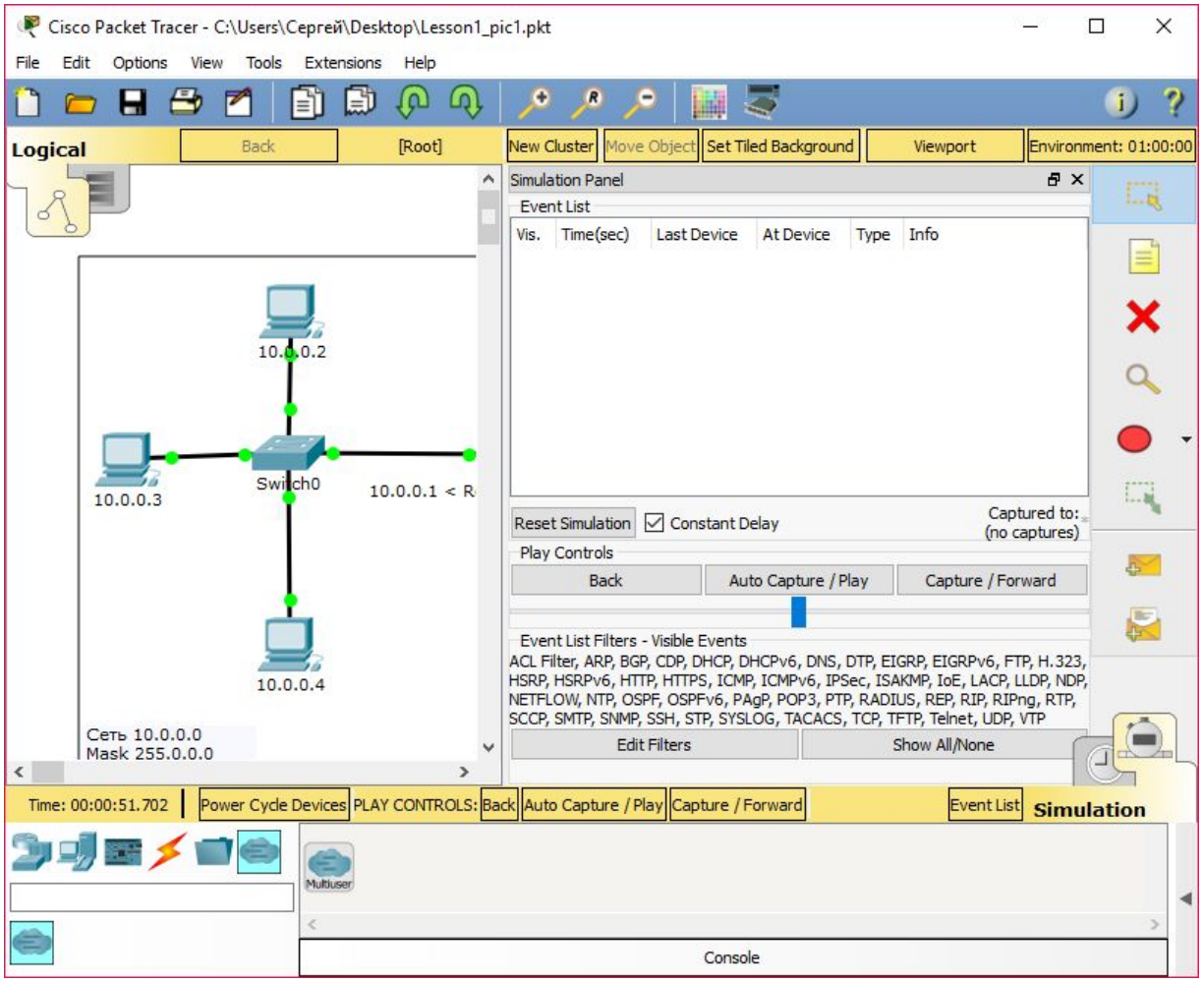
Multiuser Connection — эмуляция соединений через реальную сеть.



В Cisco Packet Tracer имеется режим симуляции и режим реального времени.

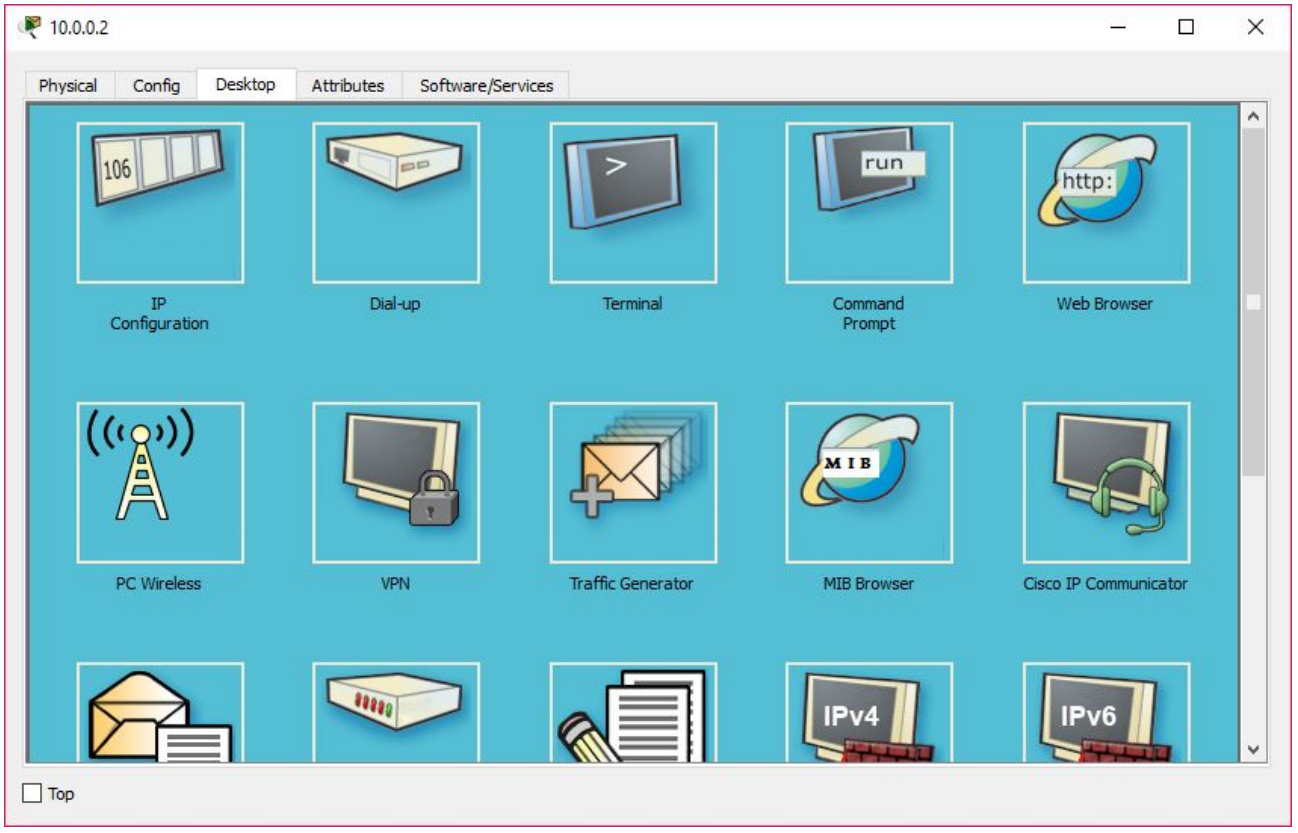


Режим реального времени.

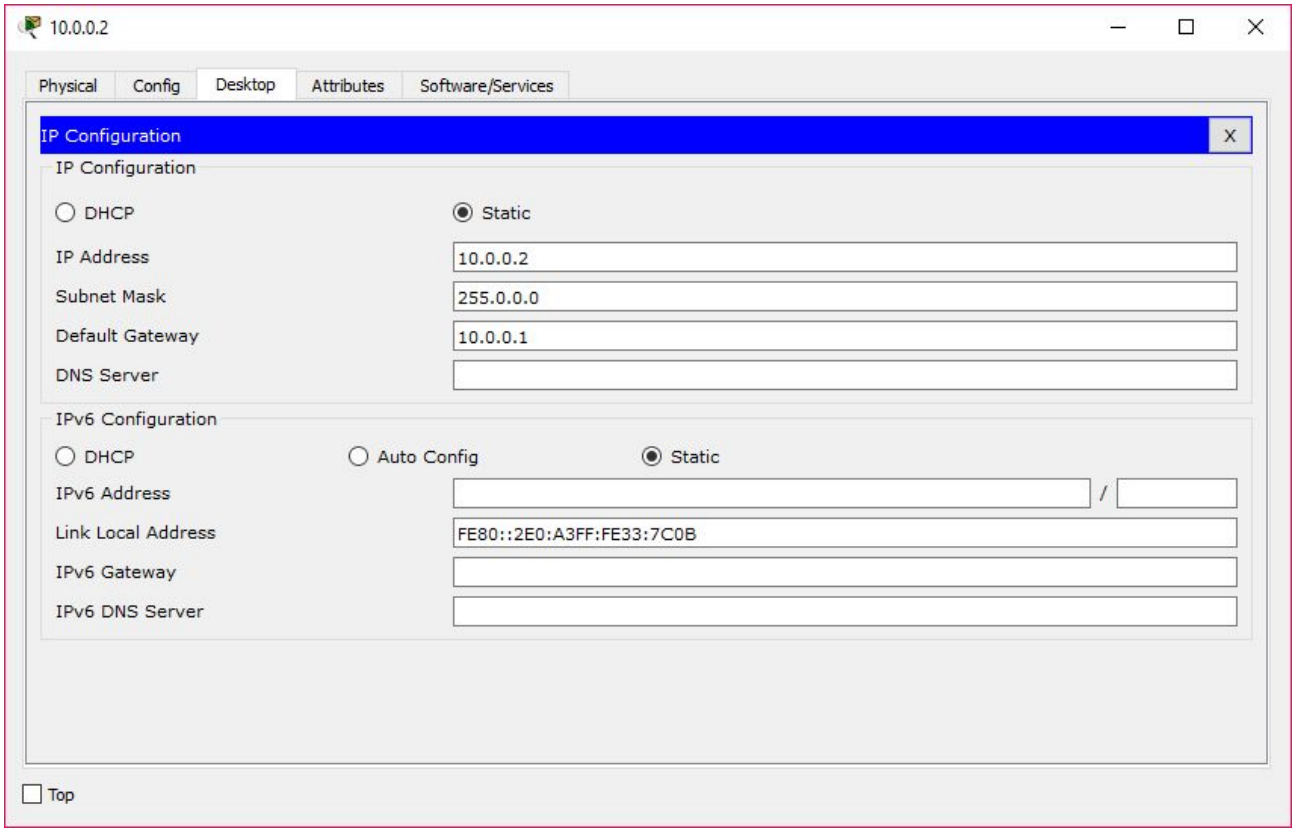


Режим симуляции.

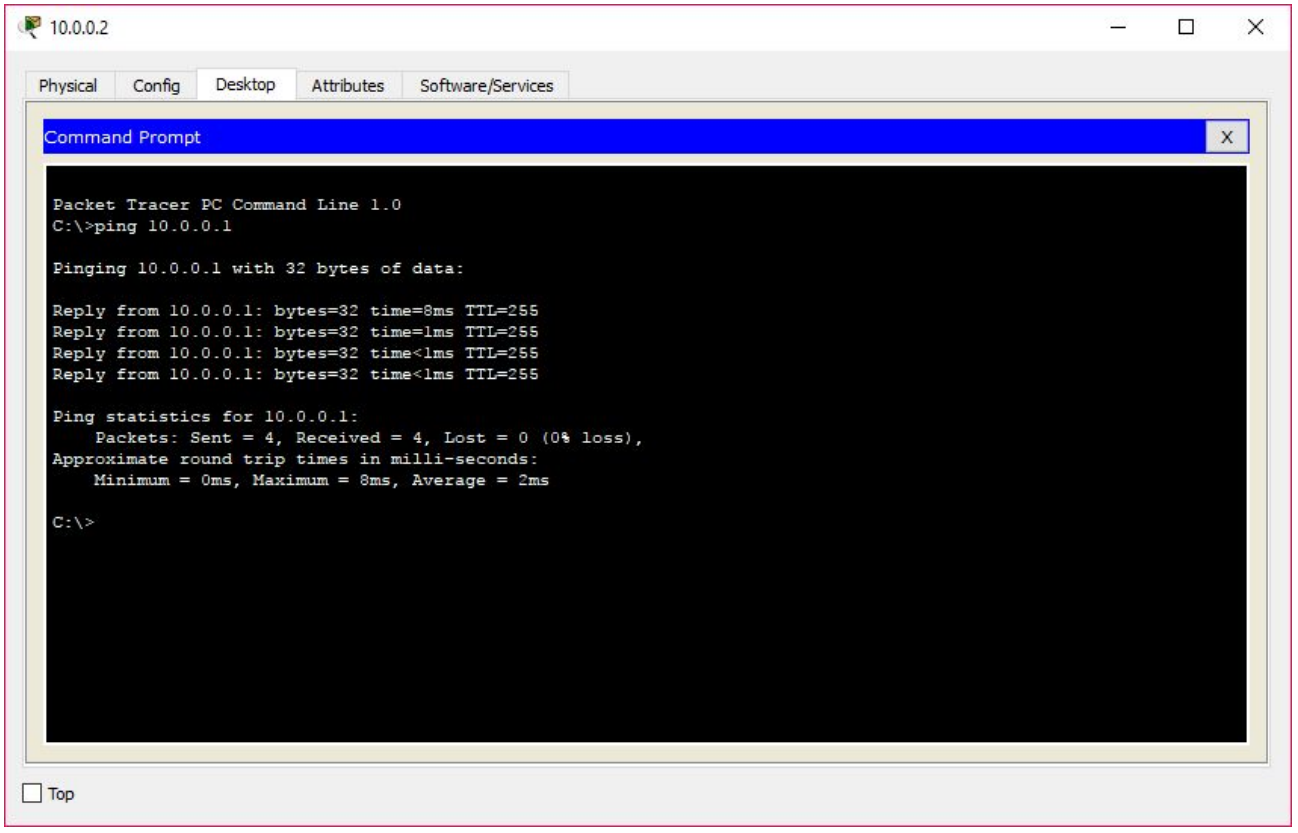
Если кликнуть на компьютер и выбрать Desktop, получим рабочий стол с набором сетевых утилит.



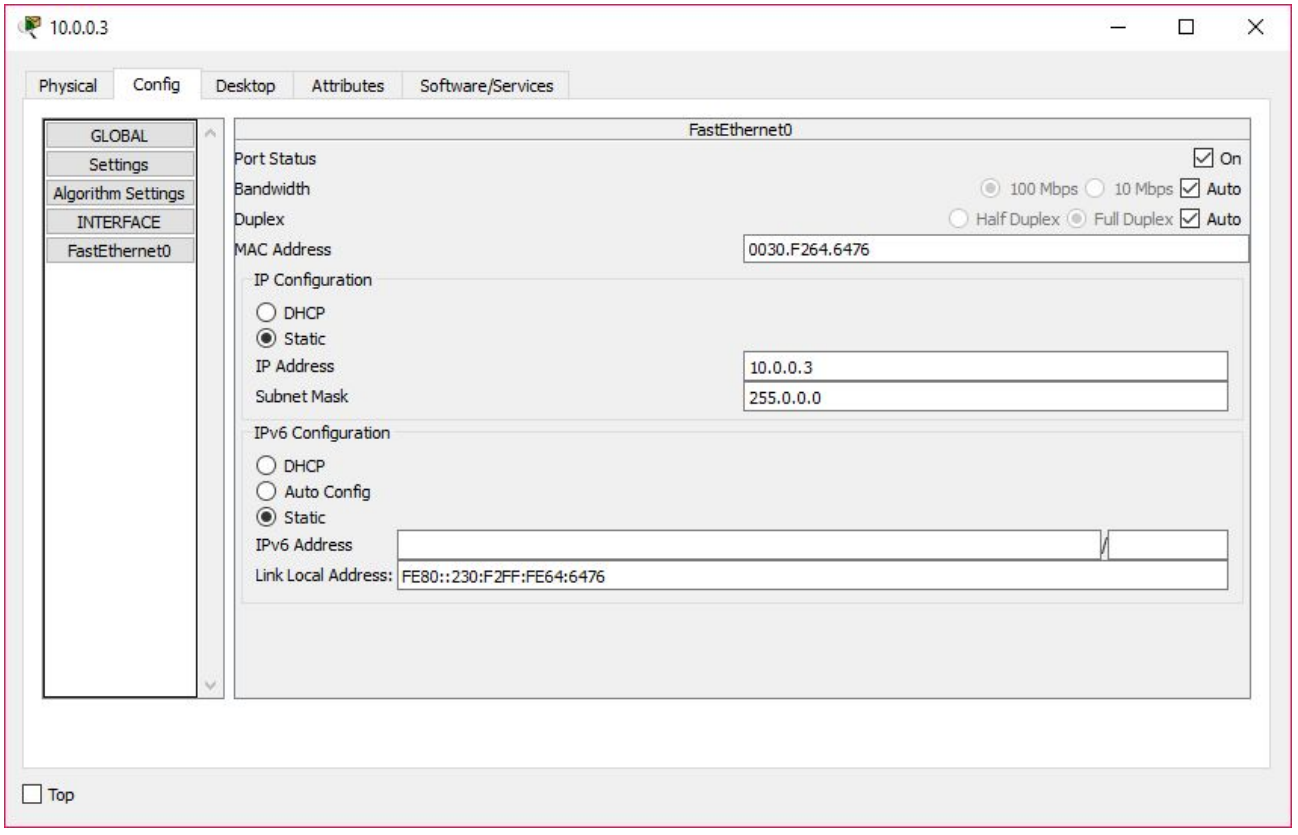
IP Configuration позволяет настроить параметры TCP/IP.



Командная строка позволяет использовать сетевые утилиты.

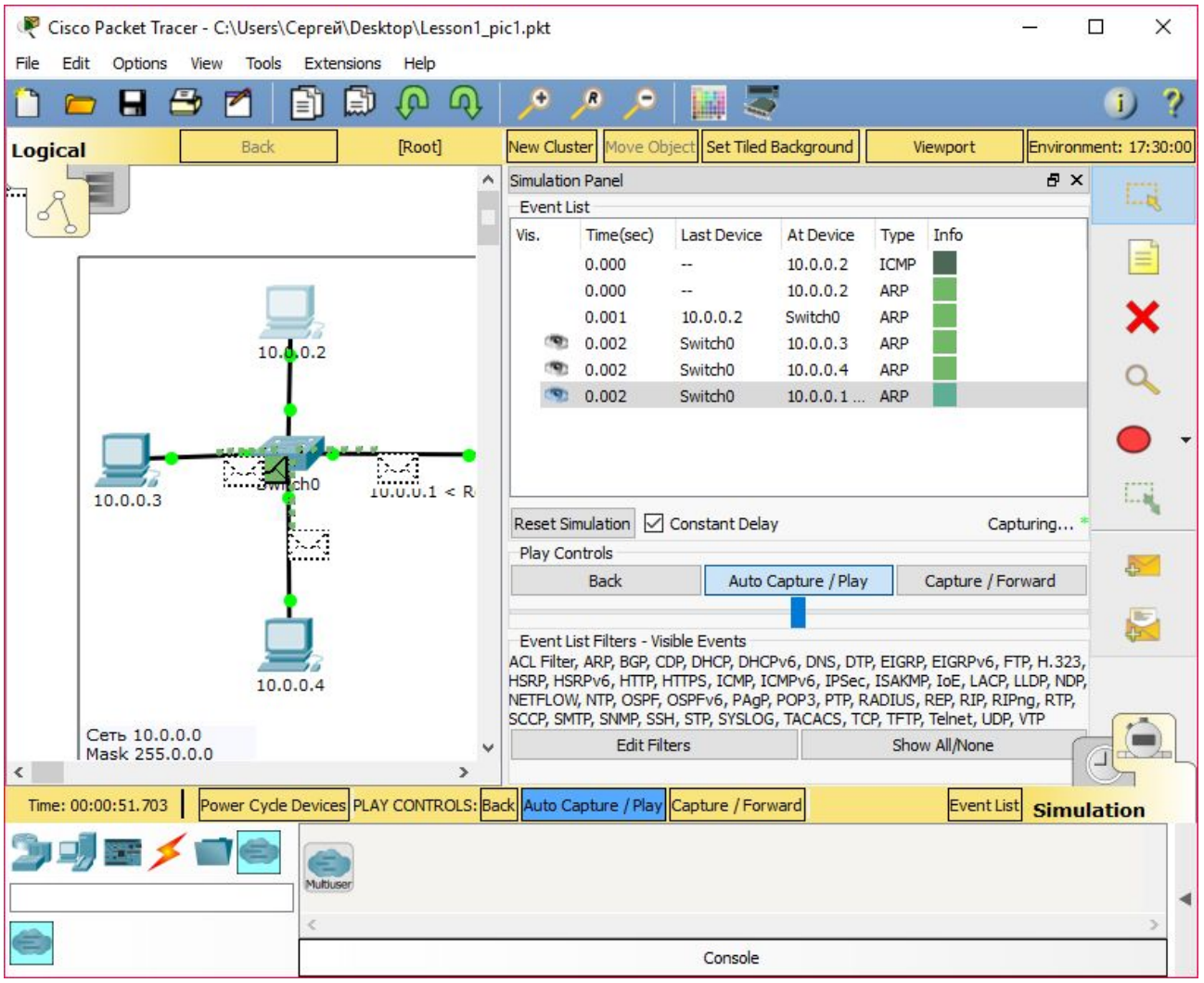


В разделе Config в разделе Interfaces можно настроить параметры сетевых интерфейсов.



В частности, выбрать режим определения скоростей: автосогласование или задать вручную.

Ethernet совместимы со старыми стандартами, поэтому применяется протокол автосогласования (работает на физическом уровне), который позволит работать совместно двум сетевым интерфейсам, работающим с разными версиями Ethernet. Если же выбираются настройки вручную (дуплекс/полудуплекс, скорость), они должны быть одинаковыми на обоих сетевых интерфейсах.



Режим симуляции позволяет наблюдать, что происходит, если вы введете команду в командной строке (например, ping 10.0.0.1). Чтобы начать симуляцию, необходимо нажать Auto Capture/Play

Сетевые утилиты

В состав TCP/IP входят диагностические утилиты, предназначенные для проверки конфигурации тестирования сетевого соединения.

Служба	Описание
arp*	Выводит для просмотра и изменения таблицу трансляции адресов, используемую протоколом разрешения адресов ARP (Address Resolution Protocol - определяет локальный адрес по IP-адресу)
hostname	Выводит имя локального хоста. Используется без параметров.
ipconfig* ifconfig**	Выводит значения для текущей конфигурации стека TCP/IP: IP-адрес, маску подсети, адрес шлюза по умолчанию, адреса WINS (Windows Internet Naming Service) и DNS (Domain Name System)
nbtstat	Выводит статистику и текущую информацию по NetBIOS, установленному поверх TCP/IP. Используется для проверки состояния текущих соединений NetBIOS.
netstat*	Выводит статистику и текущую информацию по соединению TCP/IP.
nslookup*	Осуществляет проверку записей и доменных псевдонимов хостов, доменных сервисов хостов, а также информации операционной системы, путем запросов к серверам DNS.
ping*	Осуществляет проверку правильности конфигурирования TCP/IP и проверку связи с удаленным хостом.
route	Модифицирует таблицы маршрутизации IP. Отображает содержимое таблицы, добавляет и удаляет маршруты IP.
tracert* traceroute**	Осуществляет проверку маршрута к удаленному компьютеру путем отправки эхо-пакетов протокола ICMP (Internet Control Message Protocol). Выводит маршрут прохождения пакетов на удаленный компьютер.

* - команды, доступные для исполнения в cisco PT.

** — команды, используемые в UNIX-подобных ОС.

Домашнее задание

1. Скачать и установить cisco packet tracer 7.0.

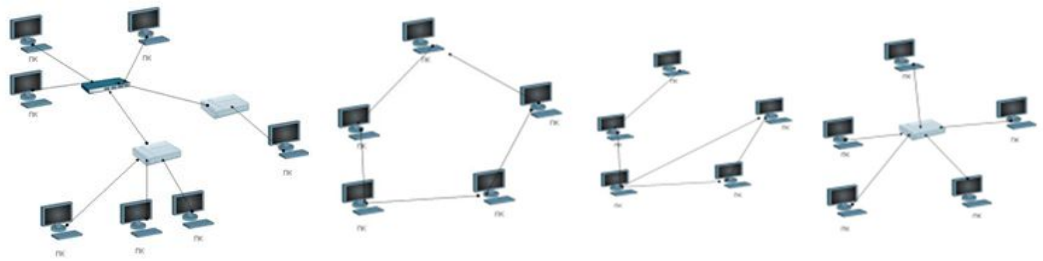
Скачать его после регистрации по ссылке:

https://www.netacad.com/ru/about-networking-academy/packet-tracer/

После регистрации на сайте вы можете найти ссылку на скачивание в разделе.

Exploratory \ Pages \ Download \ Packet Tracer

2. Диагностика физического уровня. Скачать файл packet tracer, в котором собрана сеть с несколькими хостами (в центре хаб, а также пара компьютер - компьютер), в каждом из которых проблема с линком по той или иной причине, задача поднять все линки и проверить связь командой ping
3. Скачать и установить putty. http://www.putty.org/ (понадобится в дальшнейшем)
4. Скачать и установить wireshark https://www.wireshark.org/download.html (при установке будет предложено установить драйвер pcap, это необходимо сделать, иначе wireshark не получит доступ к канальному уровню ОС).
5. Попробовать команды tracert/ping/ipconfig на домашнем компьютере.
6. Попробовать команды (по желанию) hostname / arp разобраться с выводом.
7. Посмотреть ролик про история Интернета (по желанию). https://www.youtube.com/watch?v=MbMAPoga8tE
8. Определить и записать физическую топологию сетей.



.

Дополнительные материалы

1. Таненбаум Э., Уэзеролл Д. Т18 Компьютерные сети. 5-е изд. — СПб.: Питер, 2012. — 960 с. **(Глава 1)**
2. Ролик про история Интернета. https://www.youtube.com/watch?v=MbMAPoga8tE

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. https://ru.wikipedia.org/wiki/ARPANET
2. http://xgu.ru/wiki/IP_Multicast
3. ГОСТ Р ИСО/МЭК 7498-1-99 Информационная технология (ИТ). Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель http://docs.cntd.ru/document/1200028699
4. http://minsvyaz.ru/ru/documents/3464/
5. https://ru.wikipedia.org/wiki/Маршрутизация
6. https://ru.wikipedia.org/wiki/Ячеистая_топология