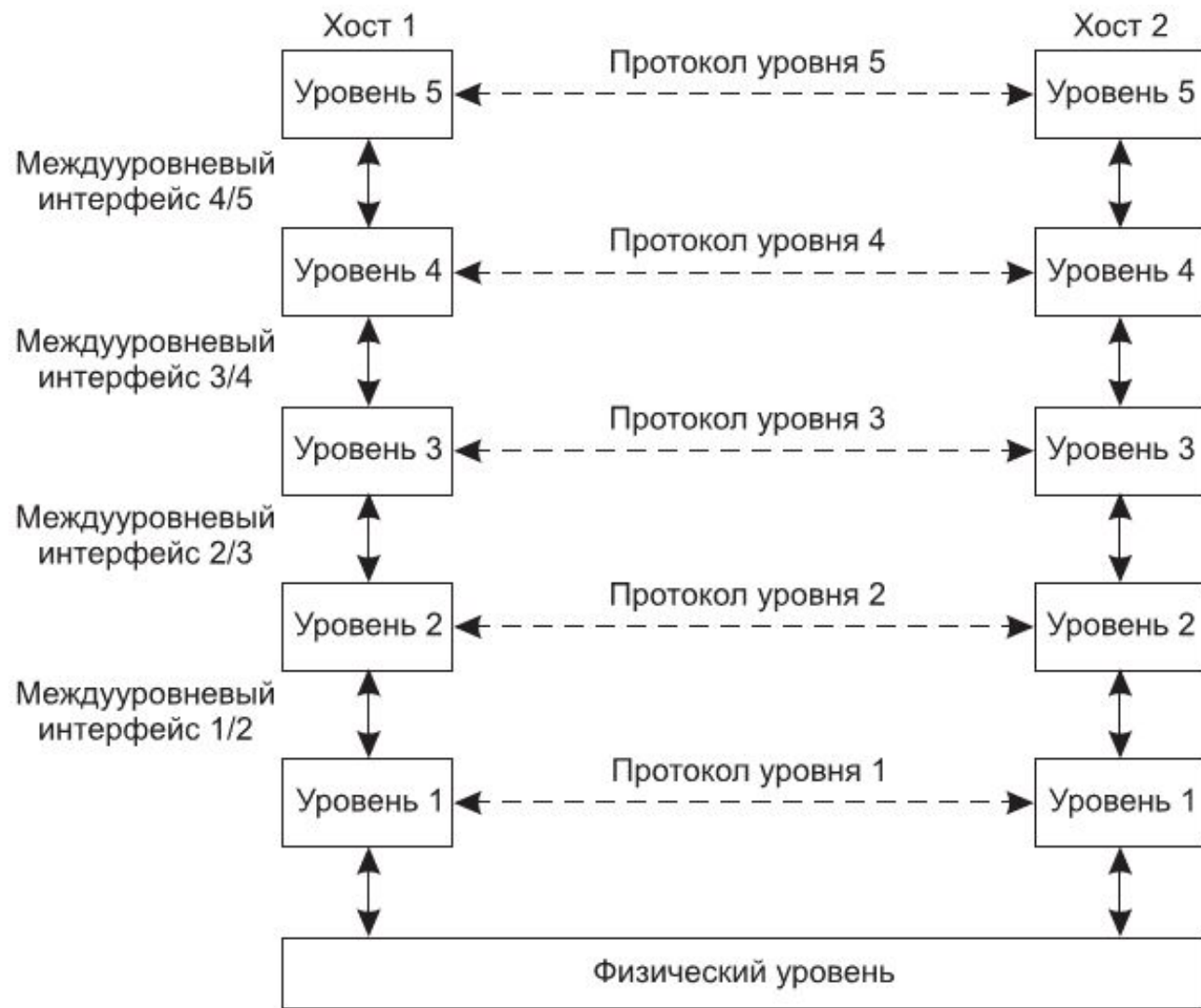


# Компьютерные Сети

# Компьютерные сети. уровни, протоколы интерфейсы.

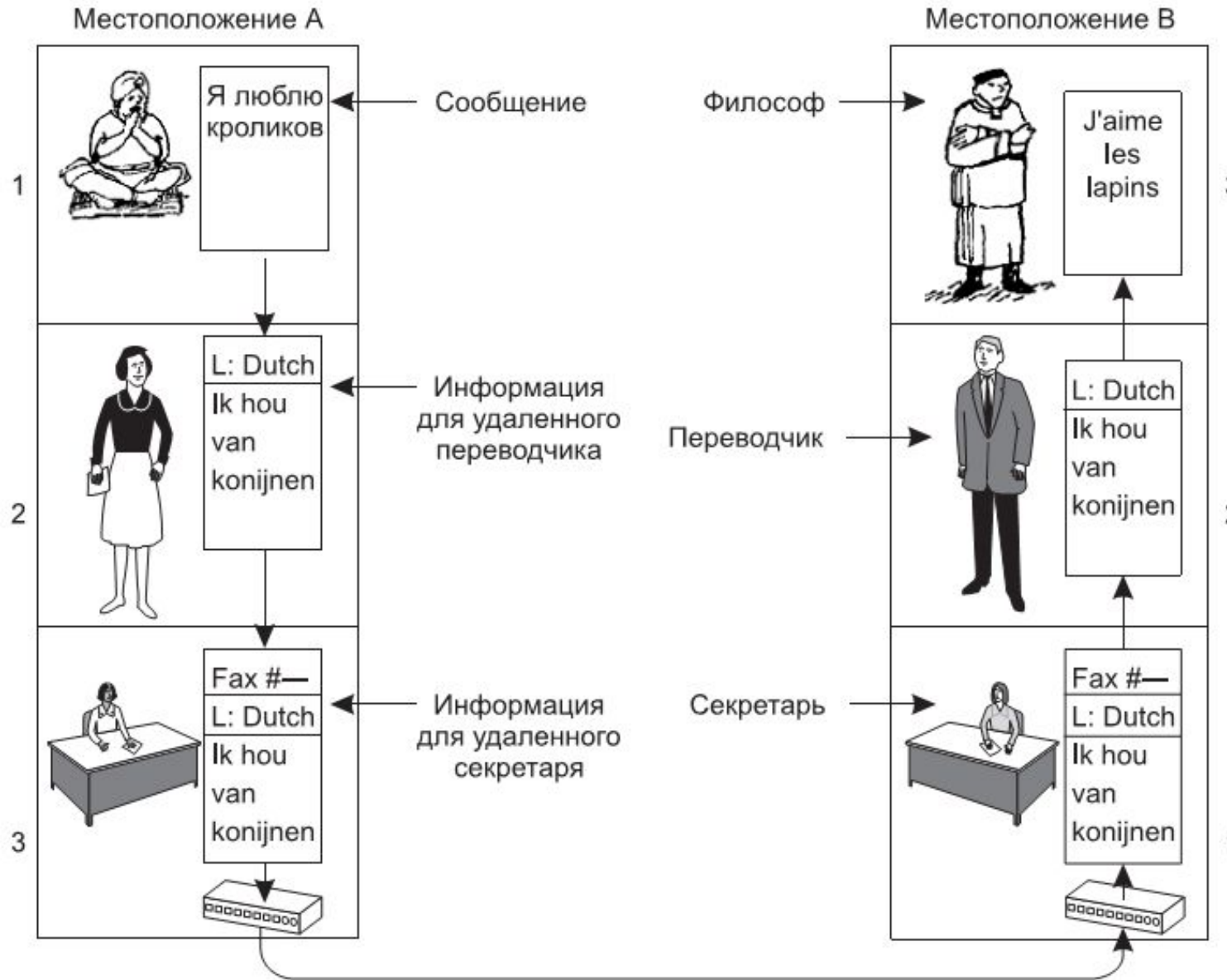
- Для упрощения структуры большинство сетей организуются в наборы **уровней**, каждый последующий возводится над предыдущим.
- Целью каждого уровня является предоставление неких сервисов для вышестоящих уровней. При этом от вышестоящих уровней скрываются детали реализации предоставляемого сервиса.
- Уровень  $n$  одной машины поддерживает связь с уровнем  $n$  другой машины. Правила и соглашения, используемые в данном общении, называются **протоколом уровня  $n$** . По сути, протокол является договоренностью общающихся сторон о том, как должно происходить общение.
- В действительности, данные не пересылаются с уровня  $n$  одной машины на уровень  $n$  другой машины. Вместо этого каждый уровень передает данные и управление уровню, лежащему ниже, пока не достигается самый нижний уровень. Ниже первого уровня располагается физическая среда, по которой и производится обмен информацией.
- Между каждой парой смежных уровней находится **интерфейс**, определяющий набор примитивных операций, предоставляемых нижним уровнем верхнему.

# Компьютерные сети. уровни, протоколы и интерфейсы.



- Набор уровней и протоколов называется **архитектурой сети**.
- Детали реализации и описания интерфейсов не являются частями архитектуры, они спрятаны внутри и не видны «снаружи».
- **Не требуется, чтобы интерфейсы на всех машинах сети были одинаковыми, лишь бы каждая машина правильно применяла все протоколы.**
- Список протоколов, используемых системой, по одному протоколу на уровень называется **стаканом**.

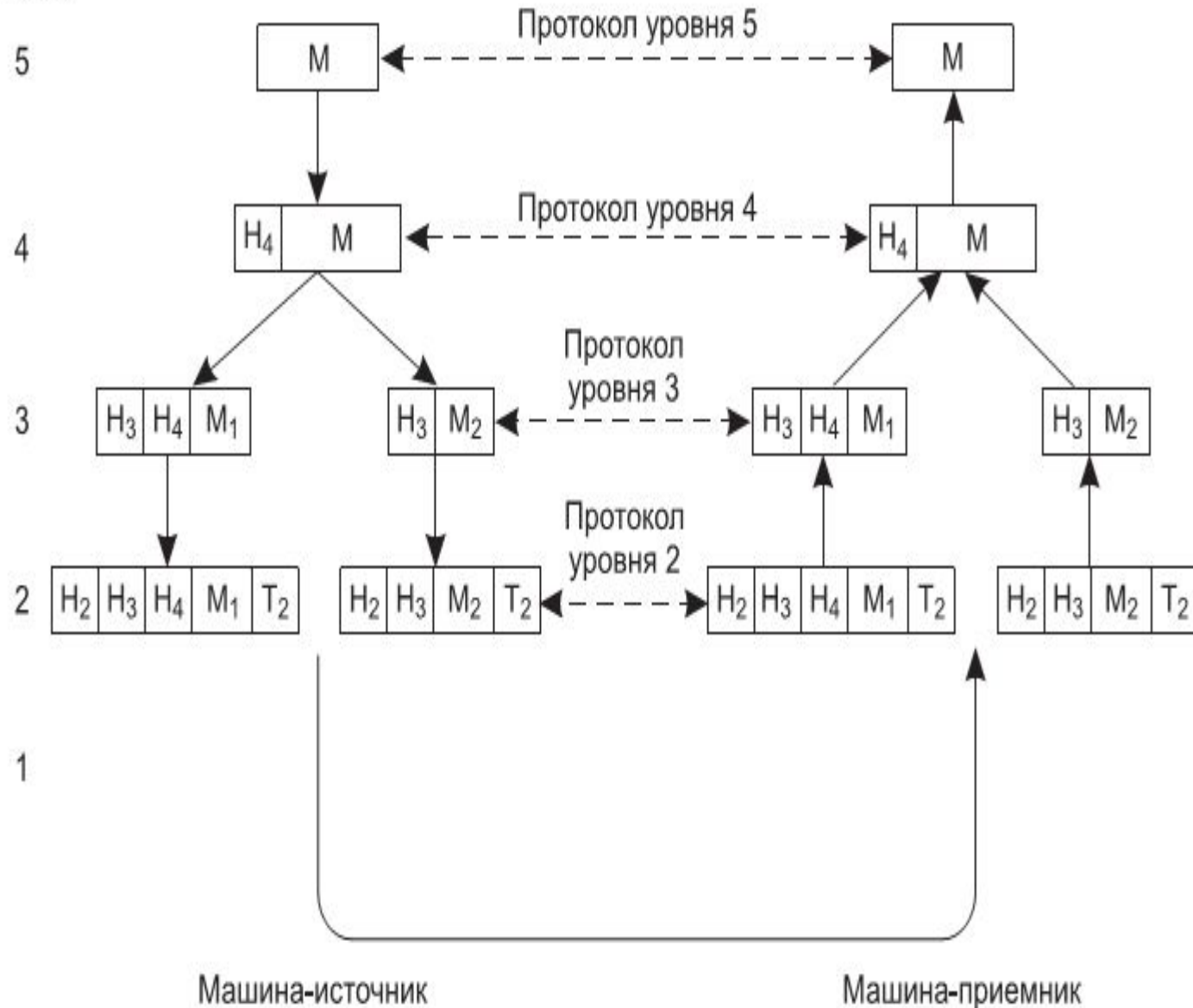
# Простой пример. Уровни, протоколы и интерфейсы.



- Рассмотрим архитектуру (уровни+протоколы) философ-переводчик-секретарь.
- Каждый протокол независим от других. Переводчики могут переключиться с голландского на финский, при условии, что оба будут согласны (**единый протокол уровня**), при этом в интерфейсах первого и третьего уровней со вторым ничего не изменится. Аналогично секретари могут использовать не факс, а электронную почту, не затрагивая другие уровни.
- **Изменение протокола каждого уровня может выполняться независимо и не повлияет на протоколы других уровней.**
- При этом вышестоящим уровням «кажется», что они

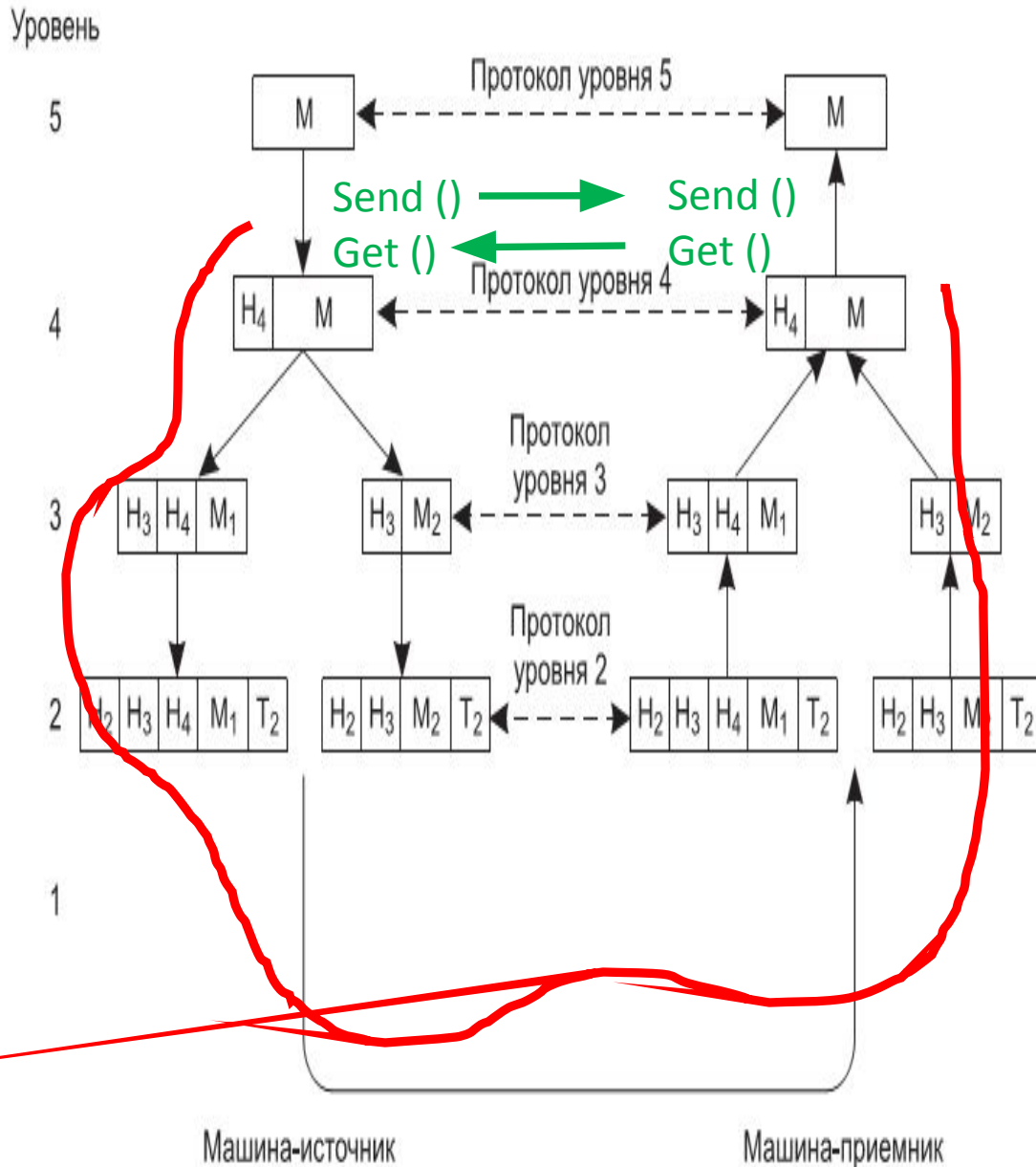
# Поток информации в пятиуровневой сети

Уровень



- Задача – передать сообщение **M (message)**.
- Уровень 5 передает **M** на уровень 4. Уровень 4 добавляет к сообщению **заголовок H4 (header)** и передает результат уровню 3.
- Заголовок может включать управляющую информацию: адреса, порядковые номера, размер, время и т.д.
- Уровень 3 разбивает сообщение на пакеты (**M1** и **M2**), добавляя к каждому пакету заголовок **H3**.
- Уровень 3 передает сообщение уровню 2, который добавляет не только заголовки (**H2**), но и контрольные суммы (**T2**), после чего передает сообщение по физическому каналу (уровень 1).

# Поток информации в пятиуровневой сети



- На получающей машине сообщение двигается по уровням вверх, при этом заголовки убираются на каждом уровне по мере продвижения сообщения. Заголовки нижних уровней более высоким уровням не передаются.

- Необходимо понять соотношение между **виртуальным** и **реальным** общением и разницу между **протоколом** и **интерфейсом**:

- Процессы уровня 4 считают свое общение горизонтальным, использующим **протокол 4-го уровня**. У каждого из них имеется процедура с названием **Send (Отправить)** и **Get (Получить)**, даже если на самом деле эти процедуры общаются не друг с другом, а с нижними уровнями при помощи интерфейсов 3/4.



## Общие выводы. Уровни, протоколы и интерфейсы.

- Абстракция одноуровневых процессов является ключевой для проектирования сетей.
- С ее помощью невыполнимая задача разработки целой сети может быть разбита на несколько меньших по размеру и вполне разрешимых проблем разработки, а именно, разработки индивидуальных уровней.
- Следует отметить, что нижние уровни в иерархии протоколов часто реализуются аппаратно или программно-аппаратно. Тем не менее при этом используются сложные алгоритмы протоколов, хотя они и внедряются в аппаратуру частично или целиком. В этом смысле все уровни иерархии можно определить как «сетевое программное обеспечение».

# Эталонные сетевые модели

- OSI (*open systems interconnection basic reference model*, базовая эталонная модель взаимодействия открытых систем) – сетевая модель стека сетевых протоколов OSI/ISO. Протоколы, связанные с эталонной моделью OSI, **сейчас не используются**, тем не менее, **сама модель до сих пор актуальна**, а свойства ее уровней очень важны.
  - Модель OSI не описывает протоколы, используемые на каждом уровне. Она просто **определяет, что должен делать каждый уровень**.
  - Тем не менее, ISO (создатель OSI) разработала протоколы для каждого уровня, те самые, которые в итоге **сейчас не используются**.
- TCP/IP - название происходит из двух важнейших протоколов семейства — Transmission Control Protocol (TCP) и Internet Protocol (IP). Наоборот, **сама модель уровней TCP/IP сейчас почти не используется, а ее протоколы являются едва ли не самыми распространенными**.



# Сетевая модель OSI

Модель OSI



Модель OSI

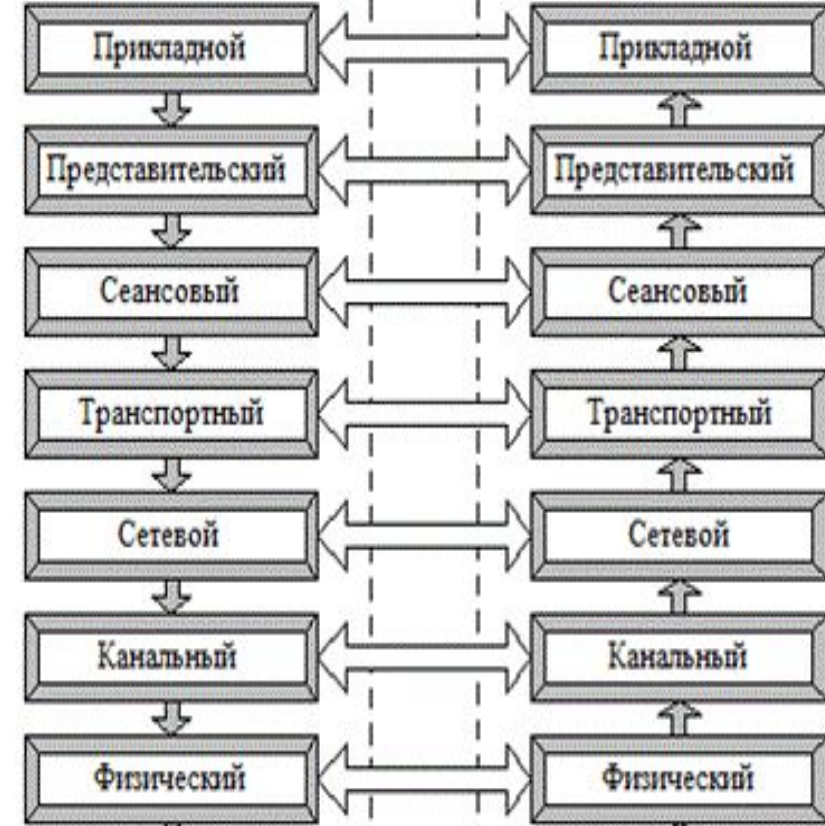


Компьютер-отправитель



Виртуальная  
связь

Компьютер-



Физическая среда

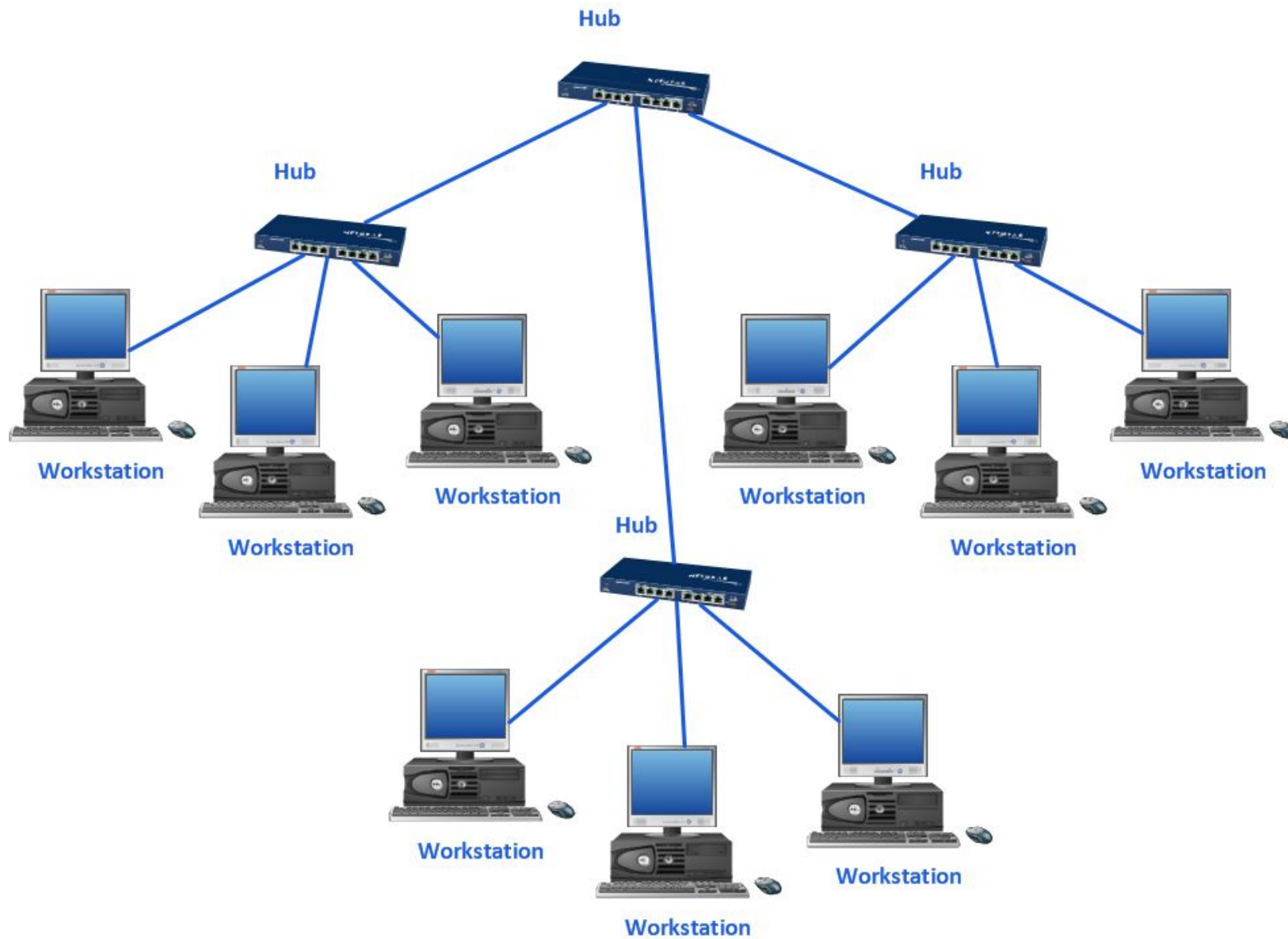
www.share

# OSI. Физический уровень.

- **Реальная передача необработанных битов по каналу связи.**
- При разработке сети необходимо убедиться, что когда одна сторона передает единицу, то принимающая сторона получает также единицу, а не ноль.
- **Принципиальными вопросами здесь являются следующие:** какое напряжение должно использоваться для отображения единицы, а какое для нуля; сколько микросекунд длится бит; может ли передача производиться одновременно в двух направлениях (симплекс, дуплекс, полудуплекс); как устанавливается начальная связь и как она прекращается, когда обе стороны закончили свои задачи; из какого количества проводов должен состоять кабель и какова функция каждого провода.
- **Вопросы разработки в основном связаны** с механическими, электрическими и процедурными интерфейсами, а также с физическим носителем (электричество (ВЧ-передача), свет (оптоволокно), радиоэфир).
- **На физическом уровне работают концентраторы (hub).** Концентраторы ретранспируют входящий сигнал с одного из портов на все остальные



# OSI. Физический уровень. Концентратор (Hub).



# OSI. Канальный уровень (уровень передачи данных)

- Канальный уровень предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками, которые могут возникнуть. Полученные с физического уровня данные, представленные в битах, он упаковывает в **кадры**, проверяет их на целостность и, если нужно, исправляет ошибки (формирует повторный запрос поврежденного кадра) и отправляет на **сетевой (верхний)** уровень.
- Еще одна проблема, возникающая на канальном уровне, — как не допустить ситуации, когда быстрый передатчик заваливает приемник данными. Может быть предусмотрен некий механизм регуляции, который информировал бы передатчик о наличии свободного места в буфере приемника на текущий момент.
- **На этом уровне работают, например, коммутаторы (switch - переключатель).** Switch предназначен для соединения нескольких узлов сети, он работает на канальном (втором) уровне модели OSI. В отличие от концентратора (hub, первый уровень OSI), который распространяет трафик от одного устройства ко всем остальным, **коммутатор передает данные только непосредственно получателю** (исключение составляет широковещательный трафик). Это повышает производительность и безопасность сети, избавляя остальные сегменты сети от необходимости обрабатывать данные, которые им не предназначались. **Но как коммутатор это делает???**

# OSI. Канальный уровень (КУ). Подуровень

## MAC

- Так каким же образом коммутатор (switch) локализует трафик?
- Одним из подуровней КУ является **подуровень управления доступом к среде** (media access control, MAC).
- **MAC обеспечивает адресацию** и механизмы управления доступом к каналам, что позволяет нескольким терминалам или точкам доступа общаться между собой в многоточечной сети (например, в локальной или городской вычислительной сети).
- Механизм адресации подуровня MAC называется физической адресацией или MAC-адресами. **MAC-адрес – уникальный серийный номер, который присваивается каждому сетевому устройству (сетевая карта или коммутатор) во время изготовления, и позволяет однозначно определить его среди других сетевых устройств в мире.**
- Коммутатор хранит в памяти таблицу коммутации, в которой **указывается соответствие MAC-адреса узла порту коммутатора**. При включении коммутатора эта таблица пуста и он работает в режиме обучения. В этом режиме поступающие на какой-либо порт данные передаются на все остальные порты коммутатора. При этом коммутатор анализирует фреймы (кадры) и, определив MAC-адрес хоста-отправителя, заносит его в таблицу на некоторое время. Впоследствии, если на один из портов коммутатора поступит кадр, предназначенный для хоста, MAC-адрес которого уже есть в таблице, то этот кадр будет передан только через порт, указанный в таблице. Если MAC-адрес хоста-получателя не ассоциирован с каким-либо портом коммутатора, то кадр будет отправлен на все порты, за исключением того порта, с которого он был получен. **Со временем коммутатор строит таблицу для всех активных MAC-адресов, в результате трафик локализуется.**

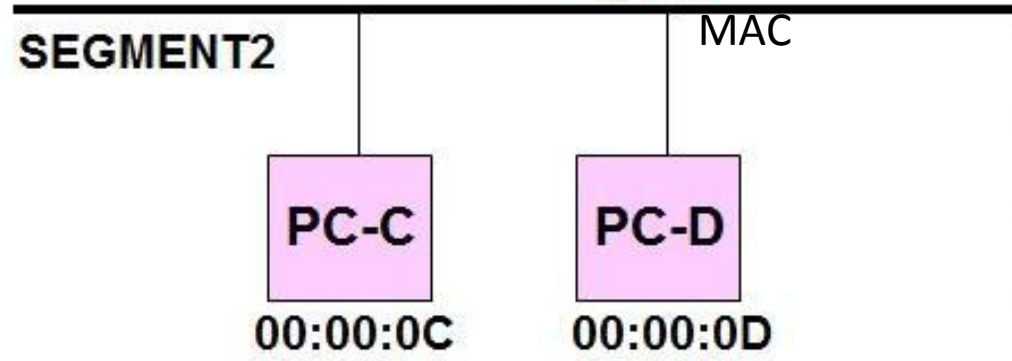
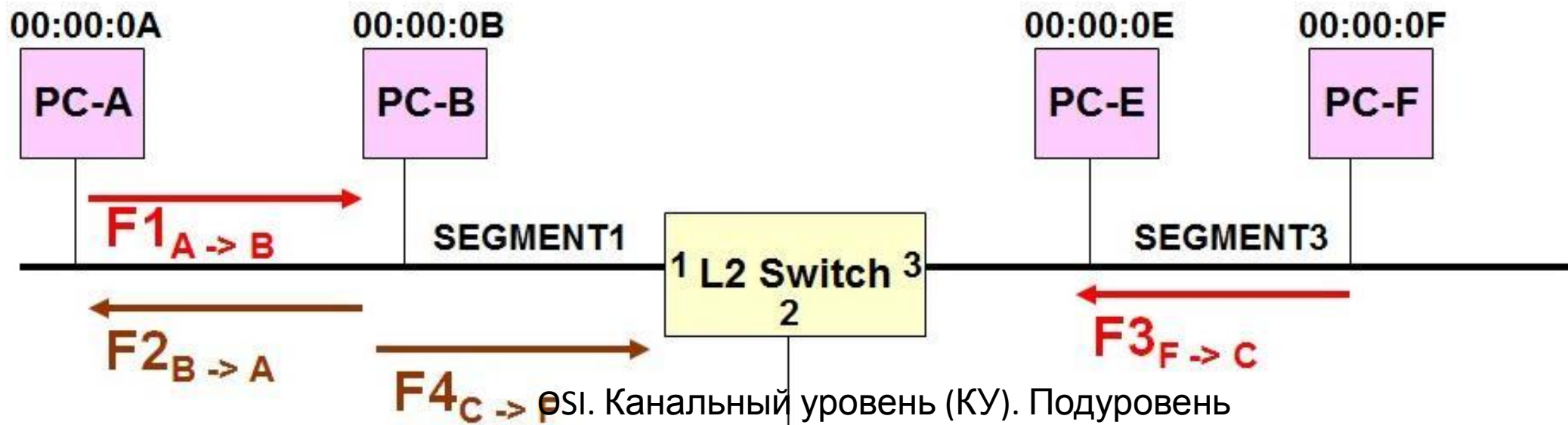
# OSI. Канальный уровень (КУ). Кадр Ethernet



**MAC  
адреса**



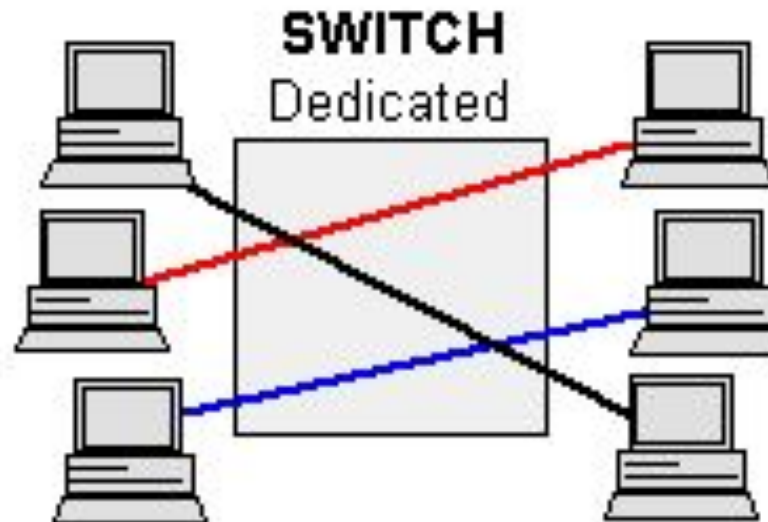
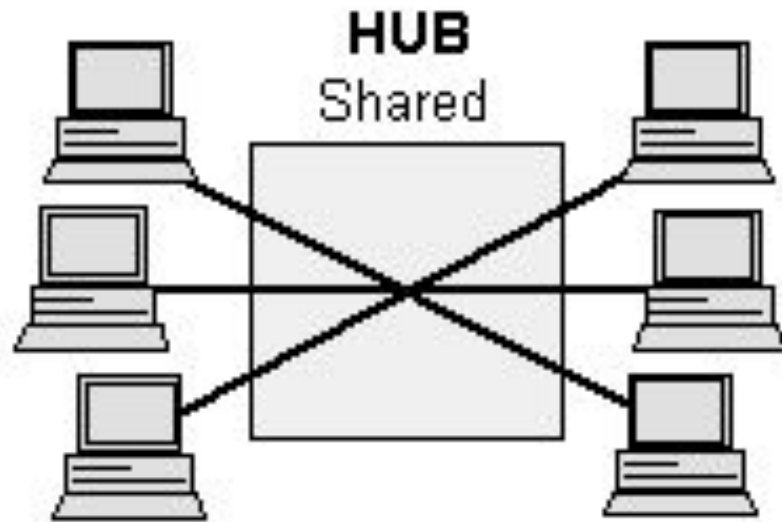
# OSI. Канальный уровень (КУ). Подуровень MAC. Обучение коммутатора.



MAC-ADDRESS	PORT
00:00:0A	1
00:00:0B	1
00:00:0F	3
00:00:0C	2

**Example of L2 Switch Learning based on Source MAC address field in incoming frames**

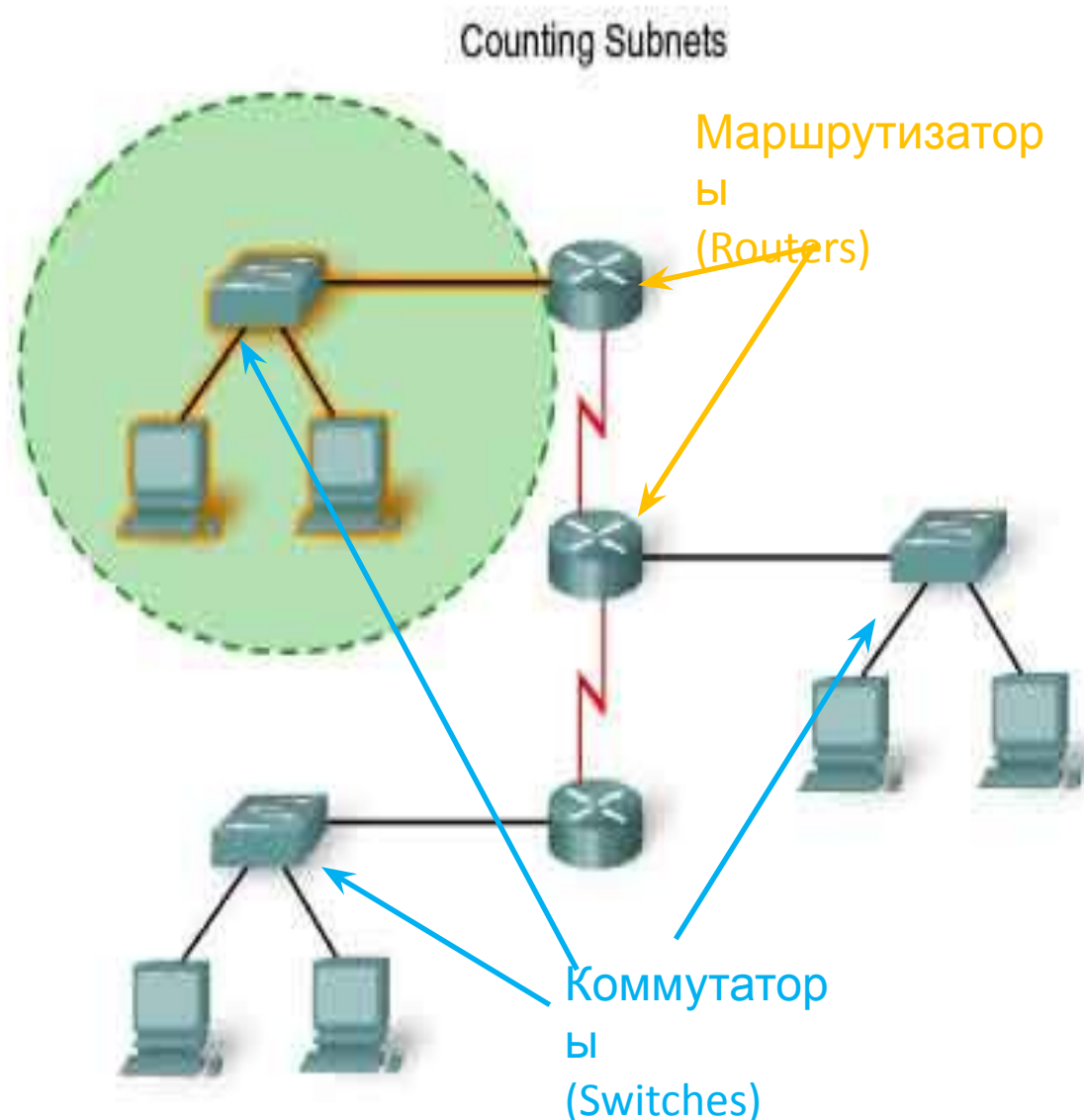
# OSI. Канальный уровень (КУ). Switch VS Hub



# Локальные и глобальные сети

- ЛВС (Local Area Network, LAN) – компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (существуют исключения). ЛВС обладает относительно регулярной топологией!
- Отдельная ЛВС может иметь связь с другими ЛВС через **маршрутизаторы (router)**, а также **быть частью глобальной вычислительной сети (ГВС)** (например, Интернет) или иметь подключение к ней.
- Чаще всего ЛВС построены на технологиях Ethernet или Wi-Fi. Для построения простой локальной сети используются **маршрутизаторы (router, сетевой уровень (СУ))**, **коммутаторы (switch, КУ)**.
- Технологии ЛВС реализуют, как правило, функции только двух нижних уровней модели OSI - физического и канального. Функциональности этих уровней достаточно для доставки **кадров (используя MAC-адрес)** в пределах стандартных (регулярных) топологий, которые поддерживают ЛВС.
- Однако из этого не следует, что компьютеры, связанные в ЛВС, не поддерживают протоколы уровней, расположенных выше канального. Эти протоколы также устанавливаются и работают на узлах локальной сети, но выполняемые ими функции не относятся к технологии ЛВС

# ЛВС и глобальные сети (ГС)



- Для обеспечения связи локальных (регулярных) сетей с глобальными (нерегулярными) применяются **маршрутизаторы**.
- Для глобальной сети, которая включает большое количество ЛВС недопустимо использовать принципы адресации, которые используются в ЛВС. **В ГС на первый план выходят задачи определения пути передачи данных, определения кратчайших маршрутов, отслеживания неполадок и «заторов» в сети.**
- Данные задачи являются прерогативой **сетевого уровня**.

# OSI. Сетевой уровень (СУ).

- Сетевой уровень занимается управлением операциями подсети. Важнейшим моментом здесь является **определение маршрутов пересылки пакетов** от источника к пункту назначения. **Маршруты могут быть жестко заданы** в виде таблиц и не меняться **либо, что бывает чаще, автоматически изменяться, чтобы избежать отказавших компонентов**. Они также **могут быть в высокой степени динамическими**, то есть вычисляемыми заново для каждого пакета с учетом текущей загруженности сети. Если в подсети одновременно присутствует слишком большое количество пакетов, то они могут закрыть дорогу друг другу, образуя заторы в узких местах. **Недопущение подобной закупорки также является задачей сетевого уровня в соединении с более высокими уровнями, которые адаптируют загрузку.**
- При путешествии пакета из одной сети в другую также может возникнуть ряд проблем. Так, **способ адресации, применяемый в одной сети, может отличаться от принятого в другой**. Сеть может вообще отказаться принимать пакеты из-за того, что они слишком большого размера. Также могут различаться протоколы и т. д.
- **Именно сетевой уровень должен разрешать все эти проблемы, позволяя объединять разнородные сети.** В широкополосных сетях, а также в ЛВС проблема маршрутизации очень проста, поэтому в них сетевой уровень очень примитивный или вообще отсутствует.



# Пояснение. IP и MAC адреса

- IP – протокол сетевого уровня. На сетевом уровне для идентификации отправителя и получателя используются IP адреса - уникальный сетевой адрес узла в сети, построенной по протоколу IP (**IPv4 (4 байта)**, IPv6).
- Проведем некоторые аналогии. **IP адрес** – это своего рода почтовый адрес, по которому с вами можно связаться из любой точки земного шара.
- **Маршрутизатор (router)**. Если Вы работаете в очень крупной организации, то в вашу канцелярию может прийти письмо с адресом «Большая организация, Большой отдел, Служба №1», в этом случае только **канцелярия (маршрутизатор)** знает, как доставить почту в «Службу №1», которая находится внутри **«Большой организации» (локальная сеть)**. Если Вы перейдете работать в «Службу №2», то ваш адрес (**IP-адрес**) изменится.
- **MAC-адрес** – например, это уникальный цвет, размер и т.п. вашего почтового ящика, который вы таскаете с собой внутри «Большой организации». Характеристики Вашего почтового ящика знает **канцелярия (маршрутизатор)** и, возможно, некоторые из ваших **коллег (коммутаторы)**, но вряд ли их (**ваш MAC-адрес**) знают те, кто



# Транспортный уровень

- **На пути от отправителя к получателю пакеты могут быть искажены или утеряны.** Многие приложения предпочитают иметь дело с надежным соединением. Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням стека (прикладной, сеансовый, представления) надежную передачу данных.
- Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем:
  - срочностью,
  - возможностью восстановления прерванной связи,
  - наличие средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол,
  - способностью к обнаружению и исправлению ошибок передачи.

# Транспортный уровень. Выбор протокола.

- Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, **в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких уровней**, а с другой стороны, этот выбор зависит от того, **насколько надежной является вся система транспортировки данных в сети**.
- Если качество каналов передачи связи очень высокое, и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. **Например, протокол UDP.**
- Если же транспортные средства изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок - с помощью предварительного установления логического соединения, контроля доставки сообщений с помощью контрольных сумм и циклической нумерации пакетов, установления тайм-аутов доставки и т.п. **Например, протокол TCP.**

# Уровни сеансовый и представления

- *Сеансовый уровень* обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, вместо того, чтобы начинать все с начала. **На практике немногие приложения используют сеансовый уровень, и он редко реализуется.**
- *Уровень представления* обеспечивает гарантию того, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе. При необходимости уровень представления выполняет преобразование форматов данных в некоторый общий формат представления, а на приеме, соответственно, выполняет обратное преобразование. **Для нас не особо интересен.**

# Прикладной уровень

- **Прикладной уровень** - это просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется *сообщением (message)*.
- Протоколы прикладного уровня:
  - FTP
  - SMTP, IMAP
  - HTTP
  - DNS
  - BitTorrent
  - Telnet

# Сетевая модель TCP/IP

## Модель OSI

Прикладной уровень

Уровень представления

Сеансовый уровень

Транспортный уровень

Сетевой уровень

Канальный уровень

Физический уровень

## Модель TCP/IP

Уровень  
приложения

Транспортный уровень

Уровень интернета

Уровень сетевого  
интерфейса

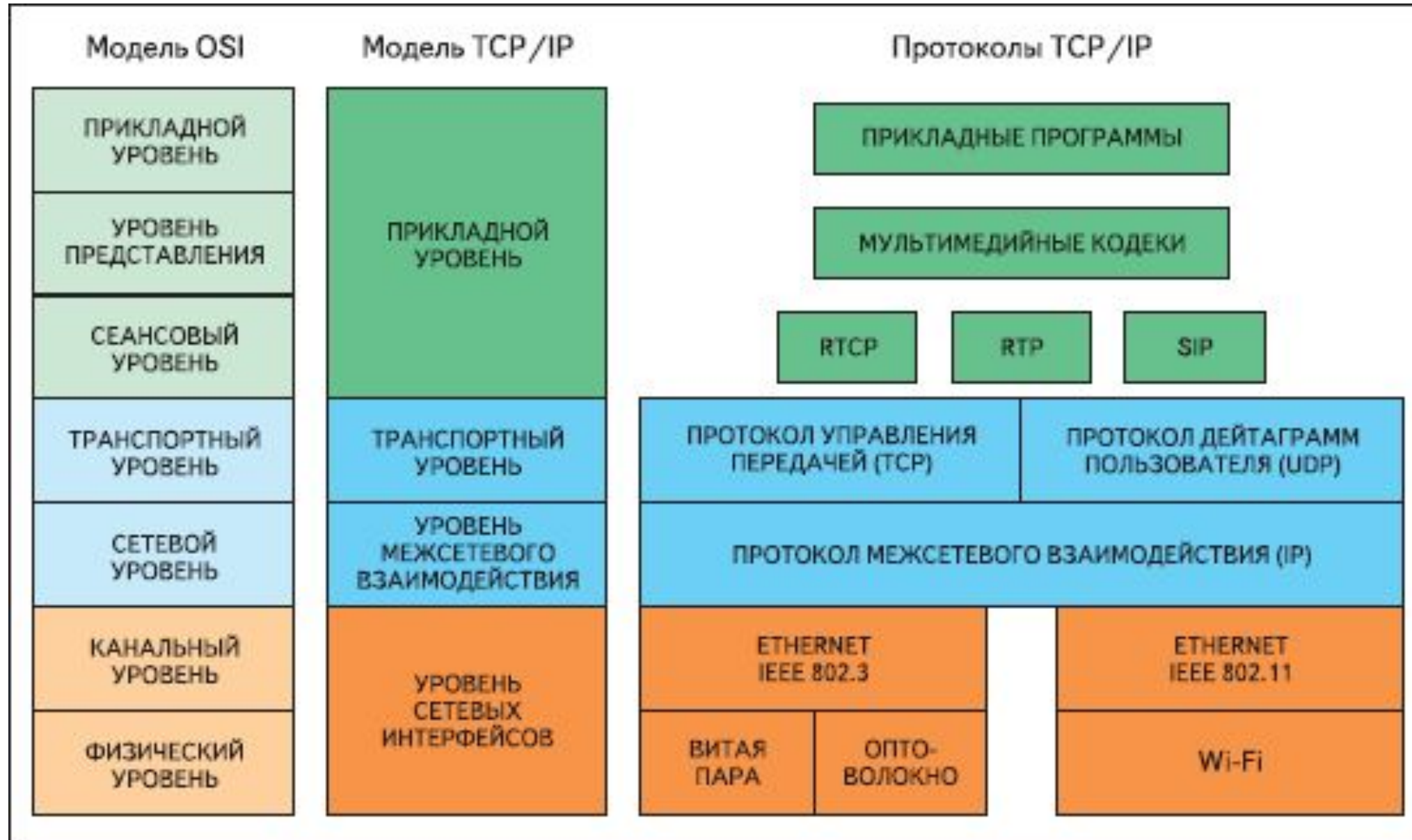
Telnet, FTP,  
SMTP, DNS,  
RIP, SNMP,  
HTTP

TCP, UDP

IP, ARP, ICMP, IGMP

Ethernet,  
Token Ring,  
ATM, Frame Reale

# Сетевая модель TCP/IP

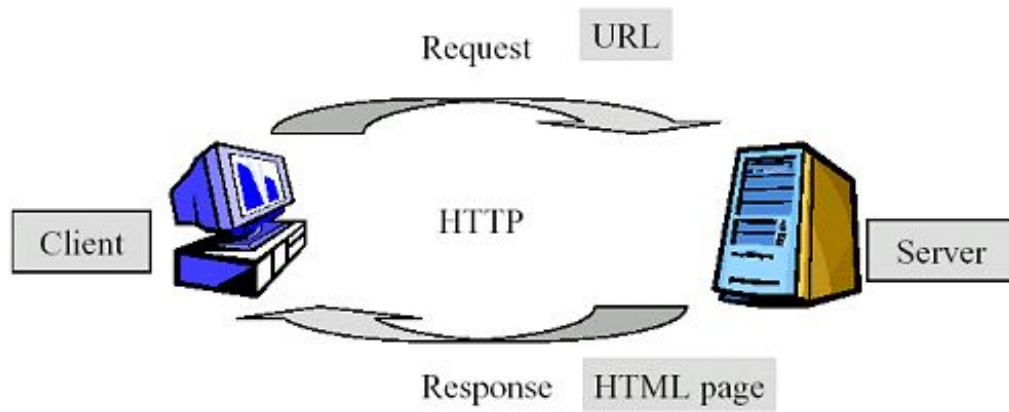
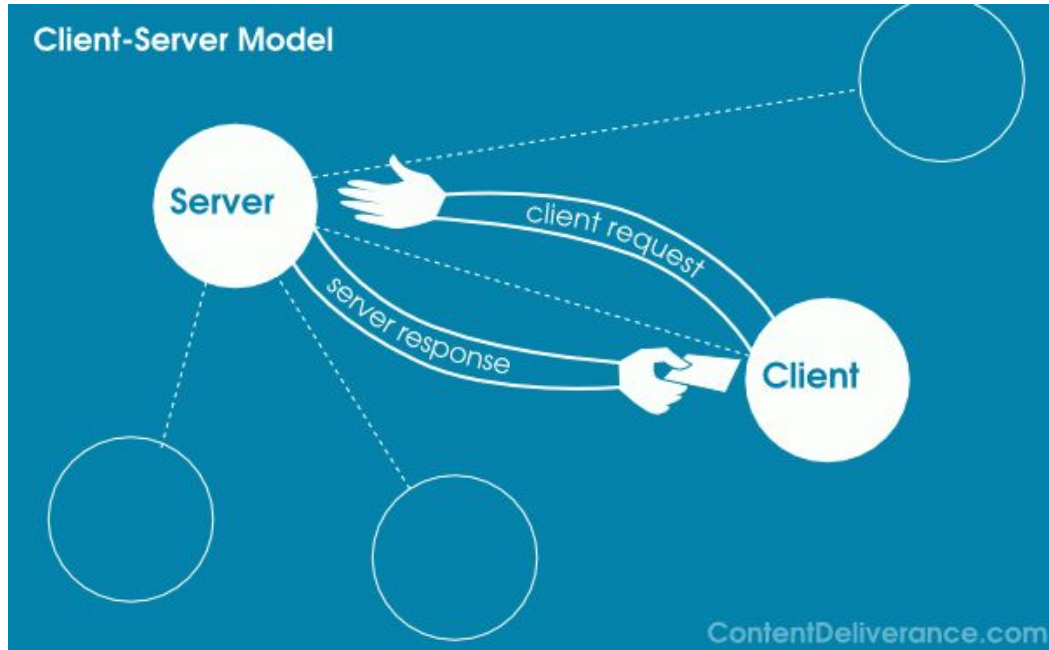




# Архитектура клиент-сервер

- Клиент-сервер - вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.
- **Физически клиент и сервер — это программное обеспечение.** Обычно они взаимодействуют через компьютерную сеть посредством сетевых протоколов (имеется в виду весь стек протоколов, а не только протоколы сетевого уровня) и находятся на разных вычислительных машинах, но могут выполняться также и на одной машине.
- **Программы — сервера**, ожидают от **клиентских программ** запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов, потоковое мультимедиа, работа с базами данных и т.п.) или сервисных функций (работа с электронной почтой, общение посредством обмена

# Архитектура клиент-сервер

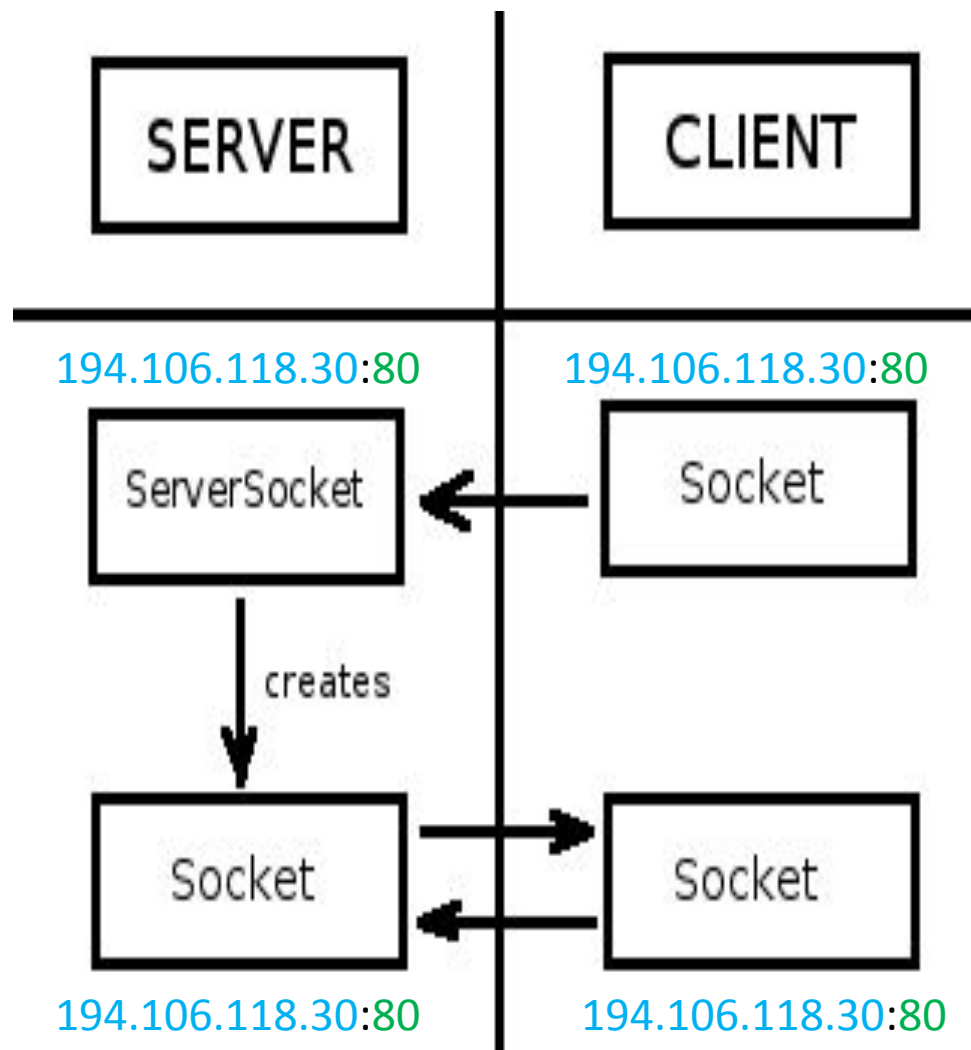


- Достоинства архитектуры клиент-сервер:
  - Отсутствие дублирования кода программы-сервера программами-клиентами.
  - Так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются.
  - Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов.
- Недостатки архитектуры клиент-сервер:
  - Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть.
  - Поддержка работы данной системы требует отдельного специалиста —

# Сокеты (программные интерфейсы)

- **Сокет** (*socket* — разъём) — название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. **Сокет** — **абстрактный объект**, представляющий конечную точку соединения.
- Сокет представляет собой пару: **IP адрес** + **порт**. Например: **194.106.118.30:80**. В данном случае **порт** — это абстрактное понятие, число в диапазоне от 0 до 65535.
- **На хосте** (серверная машина) для работы сервера обычно выделяется **порт (port)**. Хост имеет **IP адрес** + выделенный **порт** = **серверный сокет**, к которому будет подключаться клиент.
- Клиент для связи с портом хоста, который соединен в свою очередь с нужным сервером (программой), создает сокет с **IP адресом хоста** + выделенным **портом** хоста.

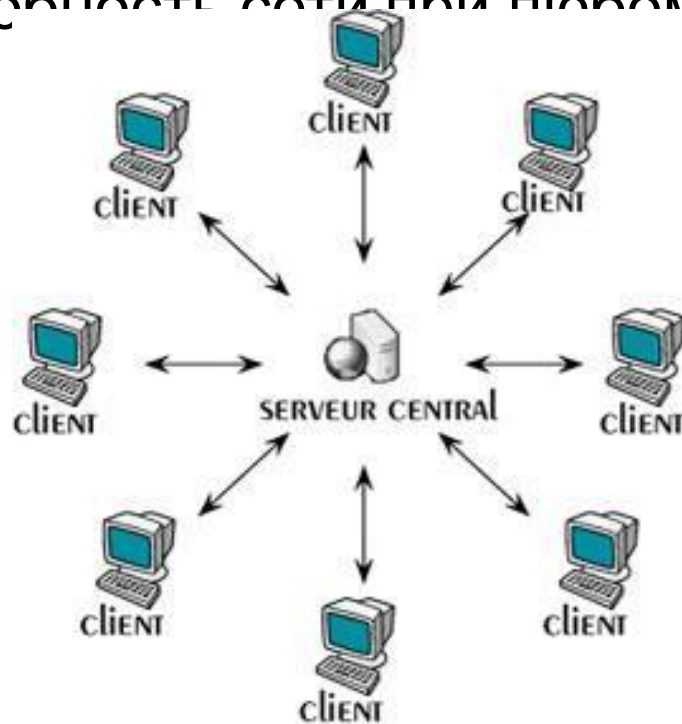
# Алгоритм работы клиент-сервер



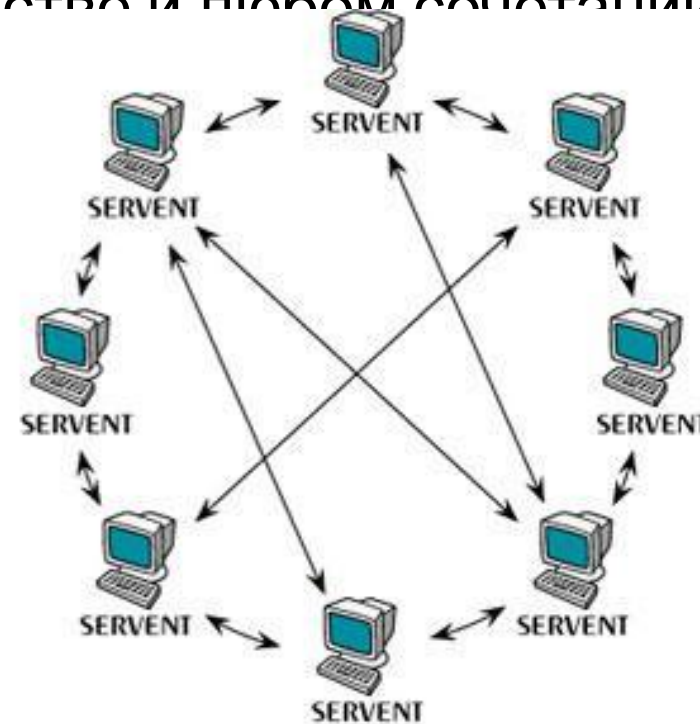
- Сервер подключается к порту на хосте и ждет соединения с клиентом;
- Клиент создает сокет и пытается соединить его с портом на хосте;
- Если создание сокета прошло успешно, то сервер переходит в режим ожидания команд от клиента;
- Клиент формирует команду и передает ее серверу, переходит в режим ожидания ответа;
- Сервер принимает команду, выполняет ее и пересылает ответ клиенту.
- и т.д.

# Одноранговые (пиринговые) сети

- **Одноранговая, децентрализованная или пиринговая** (*peer-to-peer, P2P* — равный к равному) **сеть** — это компьютерная сеть, основанная на равноправии участников.
- Часто в такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и выполняет функции сервера. В отличие от архитектуры клиент-сервер, такая организация позволяет сохранять работоспособность сети при любом количестве и любом соотношении доступных узлов.



**Интернет сегодня**



**Вероятное будущее**