**Интернет** – это инфраструктура, объединяющая компьютеры в сети для передачи данных и обмена информацией.

**Схема организации сетевого оборудования:**

Компьютер – switch (может не быть) – цепочка маршрутизаторов (плюс DHCP –сервер и DNS-resolver) – switch (может не быть) - компьютер

*Компьютер* – любое устройство в сети, способное подключаться к ней и обмениваться данными с другими устройствами этой сети (сервер, компьютер, смартфон и тд)

*Switch* (коммутатор) – сетевое устройство для соединения компьютеров по локальной сети, не имеет окна в глобальную сеть (работает до канального уровня модели OSI). Управляет трафиком, направляя его к нужному устройству, используя Mac-адрес. Благодаря switch реализуется сетевая топология звезда.

*Маршрутизатор* – сетевое устройство, обеспечивающее маршрутизацию трафика. Основная функция – найти кратчайший путь, по которому нужно отравить пакет, исходя из конечной адресации пакета.

Маршрутизатор работает в купе с DHCP-сервером, который отвечает за автоматическое назначение сетевым устройствам ip, его функционал может быть встроен в роутер или маршрутизатор. Также работает в купе с DNS-сервером, который хранит информацию о соответствии ip адресов и их доменных имен.

**Сетевая модель** – абстрактная модель взаимодействия сетевых протоколов. Описывает уровни архитектуры компьютерной сети.

**OSI** (Open System Interconnection) – Сетевая модель взаимодействия открытых систем. Является эталонной теоретической моделью, описывающая процесс передачи данных от одного устройства к другому. Не декларирует использование определенных протоколов для уровней.

Модель подразумевает прохождение данных по всем уровням дважды (инкапсуляция, декапсуляция) – отправитель преобразует данные от пользователя в электрический сигнал, получатель преобразует электрический сигнал в данные для пользователя.

Идея в прохождении сообщения через каждый уровень, который оборачивает сообщение заголовками, характерными для этого уровня (и наоборот).

7. Прикладной уровень (application layer) – уровень, который напрямую взаимодействует с данными от пользователей, то есть обеспечивает взаимодействие приложения с сетью (протоколы http, ftp, smtp, ssh и др.)

6. Уровень представления (presentation layer) – преобразование данных предыдущего уровня во взаимно согласованные форматы (кодировка, шифрование, компрессия) и обеспечение обратных процессов (декодирование, дешифрование, декомпрессия)

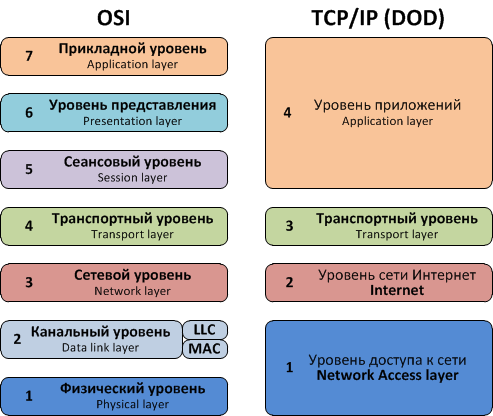
5. Сеансовый уровень (session layer) – организует сеанс связи между компьютерами (сессия –время между открытием и закрытием соединения между устройствами).

4. Транспортный уровень (transport layer) – управляет передачей данных, разбивает поток данных на сегменты или датаграммы (зависит от протокола) и отмечает у каждого в заголовке порт получатель и отправитель (*TCP и UDP)*.

3. Сетевой уровень (network layer) – Осуществляется маршрутизация трафика – определение оптимального маршрута от отправителя к получателю с помощью логической адресации IP. **Сегменты в Пакеты. Маршрутизаторы.**

2. Канальный уровень (data link layer) – разбивка пакета на кадры (фреймы) и добавление заголовка с адресом отправителя и получателя. Доставка данных до получателя внутри локальной сети осуществляется с помощью MAC-адреса. **Пакеты в фреймы. Коммутаторы.**

1. Физический уровень (physical layer) – определяет способы передачи бит данных по физическим каналам (**Wi-Fi, Bluetooth** и **Ethernet**). **Биты в сигнал. Провода**



**TCP/IP** (Transmission Control Protocol) – стандартное семейство сетевых протоколов, которые обеспечивают коммуникацию между устройствами в сети. Более практична и широко применяется на практике, в то время как модель OSI является более теоретической. Модель отличается от модели OSI кол-вом уровней и определением протоколов для уровней.

**Понятия**

**MAC-адрес** (MAC - Media Access Control) - физический адрес сетевого устройства, используется только на канальном уровне OSI (локальные сети) ввиду его непрактичности в глобальной сети (нужны огромные таблицы коммутации, чтобы знать, куда отправлять пакеты, обновление оборудования затруднит идентификацию устройства)

**Протокол** – стандарт, определяющий по каким правилам, будут взаимодействовать различные программы в сети.

**Порт** – некое число в диапазоне от 0 до 65535, которое используется для идентификации сетевого приложения на устройстве. Т.е. используются для направления сетевого трафика нужному приложению.

**IP** (Internet Protocol) – используется для маршрутизации доставки данных между устройствами сети на основе их IP адресов. С помощью IP адреса обеспечивается логическая адресация устройства, т.е. можно однозначно определить, куда отправлять пакет (за счет уникальности IP-адреса).

Приватные адреса – зарезервированные адреса для использования в локальных сетях и не могут напрямую использоваться в интернете. Если устройству с приватным IP адресом требуется выход в глобальную сеть, то роутер использует маску подсети, для определения находится ли запрос в пределах подсети или требуется выход в глобальную сеть - в случае необходимости применяется протокол NAT (каждому приватному IP адресу соответствует порт роутера).

Хост – устройство с публичным ip адресом. Подсеть – все устройства, которые обслуживает конкретное сетевое оборудование (т.е. по отношению к этому оборудованию все устройства являются подсетью, а он их хостом).

Маска подсети –используется для определения того, что запрос находится за пределами локальной сети (для получения адреса подсети путем применения логическое &)

Публичный адрес – уникальный адрес, используемый в глобальной сети интернет (уникальность контролируется организацией IPS).

127.0.0.1 – адрес localhost

192.168.xxx.xxx – адреса, зарезервированные для приватных ip

Версии протокола IP:

IPv4 – использует 32 битную адресацию (4 миллиарда уникальных адресов), которое разделено на 4 октета, отделенные друг от друга точкой, каждый октет может принимать значения от 0 до 255.

245.14.23.61 – пример.

IPv6 – 128 битная адресация, каждая секция имеет 32 бита (4 шестнадцатеричных символа) разделенные двоеточием, что решает проблему исчерпания адресов в будущем.

2001:0db8:85a3:0000:0000:8a2e:0370:7334 – пример.

Сокет – комбинация IP адреса и порта.

DNS – ‘псевдоним ip адреса’ система сопоставления публичных ip адресов символьным обозначениям (система доменных имен) предоставляет возможность использования понятных человеку имен и решает проблему изменения ip адреса при смене локации (для компаний), в случае с доменным именем закрепленным за компанией, к этому доменному имени привязывается новый ip.

Получение Ip адреса по домену: сначала на компе был файл соответствия ip адресов и доменных имен – файл hosts. С ростом числа доменов стали использоваться DNS сервера, которых хранят эти соответствия.

|  |  |  |
| --- | --- | --- |
|  | TCP | UDP |
| Надежность передачи данных | Обеспечивает надежную передачу данных с гарантией доставки без потерь и в правильном порядке. Использует механизмы подтверждения доставки и повторной передачи. | Передает данные без гарантии доставки, без подтверждения доставки или повторной передачи. |
| Управление потоком | Регулирует поток данных между отправителем и получателем, предотвращает перегрузку сети или переполнение буфера получателя. | Не управляет потоком данных. Отправляет данные на скорости, с которой они генерируются приложением. |
| Установка соединения | Установка соединения между клиентом и сервером (трехстороннее рукопожатие) | Соединение с сервером не устанавливается |
| Получение данных от сервера | Можно получать ответы от сервера в виде входящих пакетов данных | Не подразумевает получение пакетов от сервера |
| Затраты на накладные расходы | Обеспечение надежности и управление потоком приводят к большему объему заголовков и дополнительным затратам на накладные расходы. | Более легковесный и эффективный протокол без дополнительных механизмов надежности, что уменьшает накладные расходы. |
| Задержка и скорость | Из-за механизмов надежности и управления потоком может приводить к более высокой задержке и меньшей скорости передачи данных по сравнению с UDP. | Обычно имеет меньшую задержку и более высокую скорость передачи данных, так как отсутствуют механизмы надежности и управления потоком. |
| Применение | Чаще используется в приложениях, где надежность и последовательность доставки данных критически важны, таких как веб-серверы, электронная почта, передача файлов. | Применяется в приложениях, где более высокая скорость и меньшая задержка имеют большее значение, такие как приложения с реально временным аудио и видео взаимодействием (стриминг, приложения видеозвонков) |

**Про браузер**

www. (world wide web) – распределенная система, представляющая собой систему взаимосвязанных посредством гиперссылок веб-страниц (гипертекстовых документов в интернете), для загрузки и просмотра которых необходима специальная программа — браузер. На прикладном уровне использует протоколы http или https.

То есть интернет != веб, это только одна из разновидностей сущностей, с которыми можно взаимодействовать с помощью интернета, помимо веб-сервисов так же присутствуют сервисы телефонии (технология VoIP), электронной почты (SMTP), обмена файлами (FTP) и др.

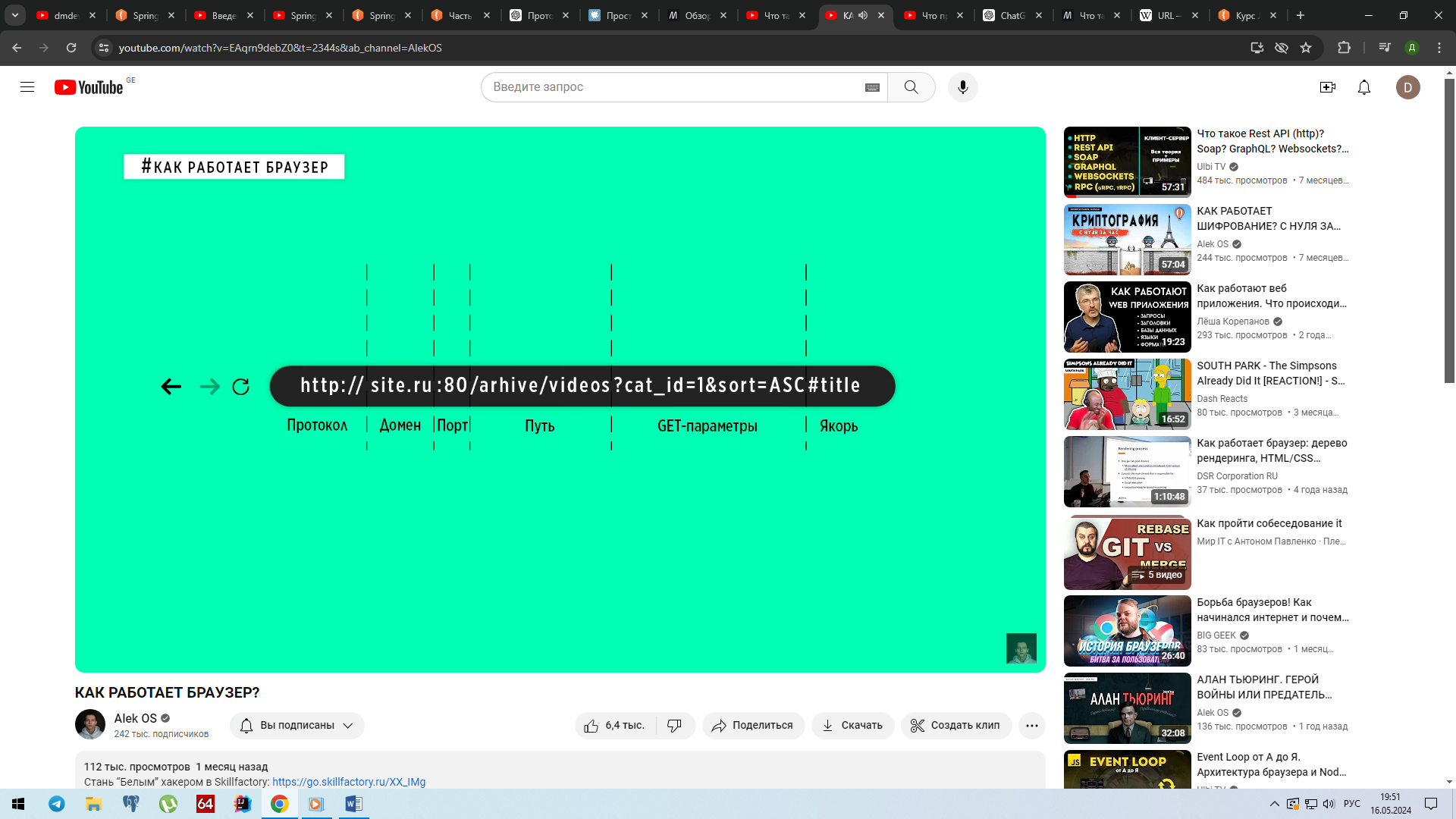
URI – уникальный идентификатор ресурса в интернете (URL + параметры)

URL (Uniform Resource Locator) – формат записи строки, которая предоставляет информацию о местонахождении ресурса в сети и способе взаимодействия с ним. Может включать только символы в кодировке ASCI, причем некоторые из них должны быть экранированы с помощью %, остальные кодируются с помощью последовательности байт в 16 формате, перед каждым байтом ставится %

<https://example.com/путь/с/символами/ы>

https://example.com/%D0%BF%D1%83%D1%82%D1%8C/%D1%81/%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%B0%D0%BC%D0%B8/%D1%8B

Состоит из:



Протокол – задает протокол взаимодействия между клиентом и сервером (http, https, ftp и др.)  
Домен, порт – адрес, по которому расположен ресурс  
Путь – указатель на конкретный ресурс на сервере  
Get-параметры – дополнительные параметры, передаваемые на сервер  
Якорь – метка на HTML-странице, которая указывает браузеру, что нужно проскролить страницу до этого якоря

**Что происходит при обращении браузера к веб-сайту:**

1) Определение поисковой запрос или url.

2) Если это введенная строка соответствует формату URL, то парсит его, если поисковой запрос, то обращение к веб-сервису поисковой системы по тем же шагам, введенный текст передается в качестве параметра этого запроса.

3) Парсинг: Определяется протокол взаимодействия (если протокол не указан, то браузер обратится к HSTS – preload list – механизм принудительно активирующие протокол https, который содержит домены, к которым должен быть применен https, если в списке его нет, то он попробует подсоединится с помощью https, если безуспешно, то http). Определение IP адреса с помощью DNS-адреса (ищется в кэшах, файл hosts, затем иерархически в DNS-resolver), если порт не был указан, то применяется порт по умолчанию http – 80, https -443.

4) Если использовался протокол https, то применяется TLS протокол

5) Установление соединения с сервером с помощью TCP-протокола

6) Формирование и отправка HTTP-запроса

7) Получение соответствующего ответа от сервера