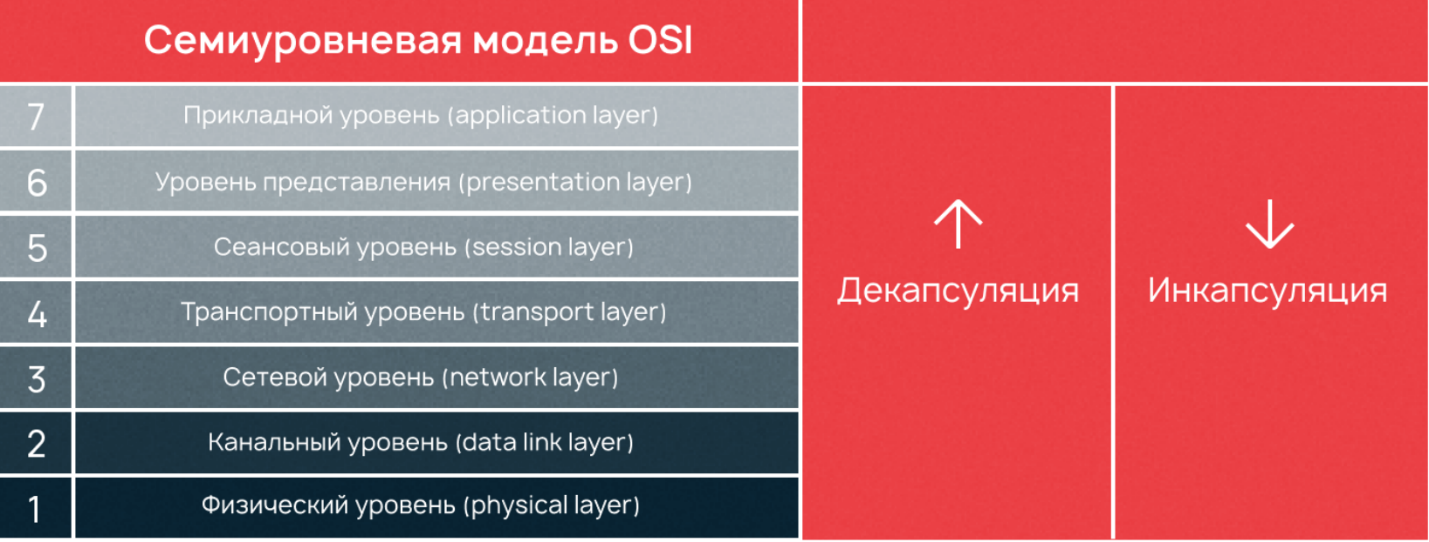
**Сетевая модель OSI**

Все, что происходит при отправке и приеме данных, детально описывает семиуровневая модель OSI.

Сетевая модель OSI имеет семь уровней, иерархически расположенных от большего к меньшему. Cамым верхним является седьмой (прикладной), а самым нижним — первый (физический). Модель OSI разрабатывалась еще в 1970-х годах, чтобы описать архитектуру и принципы работы сетей передачи данных.

На **седьмом** уровне информация представляется в виде данных, на **первом** — в виде бит

Процесс, когда информация отправляется и переходит из данных в биты, называется **инкапсуляцией**. Обратный процесс, когда информация, полученная в битах на первом уровне, переходит в данные на седьмом, называется декапсуляцией.



Каждый уровень имеет свои PDU (Protocol Data Unit), представляемые в той форме, которая будет понятна на данном уровне и, возможно, на следующем до преобразования. Работа с чистыми данными происходит только на уровнях с пятого по седьмой.

*Уровни группы Media Layers (L1, L2, L3) занимаются передачей информации (по кабелю или беспроводной сети), используются сетевыми устройствами, такими как коммутаторы, маршрутизаторы и т.п. Уровни группы Host Layers (L4, L5, L6, L7) используются непосредственно на устройствах, будь то стационарные компьютеры или мобильные устройства.*

*Первые четыре уровня — специализация сетевых инженеров. С последними тремя они не так часто сталкиваются, потому что пятым, шестым и седьмым занимаются разработчики.*

1. **Физический:** Он отвечает за обмен физическими сигналами между физическими устройствами, «железом». Устройства физического уровня оперируют битами. Они передаются по кабелям (например, через оптоволокно) или без — например, через Bluetooth или IRDA, Wi-Fi, GSM, 4G и так далее.
2. **Второй уровень, канальный**: Второй уровень решает проблему адресации при передаче информации. Канальный уровень получает биты и превращает их в кадры (frame, также «фреймы»). Задача здесь — сформировать кадры с адресом отправителя и получателя, после чего отправить их по сети.

На втором уровне OSI работают коммутаторы, их задача — передать сформированные кадры от одного устройства к другому, используя в качестве адресов только физические MAC-адреса.

На канальном уровне активно используется протокол **ARP** (Address Resolution Protocol — протокол определения адреса). С помощью него 64-битные MAC-адреса сопоставляются с 32-битными IP-адресами и наоборот, тем самым обеспечивается инкапсуляция и декапсуляция данных.

## Третий уровень, сетевой: На третьем уровне появляется новое понятие — маршрутизация. Для этой задачи были созданы устройства третьего уровня — маршрутизаторы (их еще называют роутерами). Маршрутизаторы получают MAC-адрес от коммутаторов с предыдущего уровня и занимаются построением маршрута от одного устройства к другому с учетом всех потенциальных неполадок в сети.

## Четвертый уровень, транспортный: Его главной задачей является транспортировка пакетов.

## При передаче данных, наиболее чувствительных к потерям на транспортном уровне, используется протокол TCP, контролирующий целостность доставленной информации.

## Для мультимедийных данных, наиболее чувствительных к задержкам, используется протокол UDP, позволяющий организовать связь без установки соединения.

1. **Пятый уровень, сеансовый:** Пятый уровень оперирует чистыми данными. Помимо пятого, чистые данные используются также на шестом и седьмом уровне. Сеансовый уровень отвечает за поддержку сеанса или сессии связи. Пятый уровень оказывает услугу следующему: управляет взаимодействием между приложениями, открывает возможности синхронизации задач, завершения сеанса, обмена информации.

Службы сеансового уровня зачастую применяются в средах приложений, требующих удаленного вызова процедур, т.е. чтобы запрашивать выполнение действий на удаленных компьютерах или независимых системах на одном устройстве (при наличии нескольких ОС).

Примером работы пятого уровня может служить видеозвонок по сети. Во время видеосвязи необходимо, чтобы два потока данных (аудио и видео) шли синхронно. Когда к разговору двоих человек прибавится третий — получится уже конференция. Задача пятого уровня — сделать так, чтобы собеседники могли понять, кто сейчас говорит.

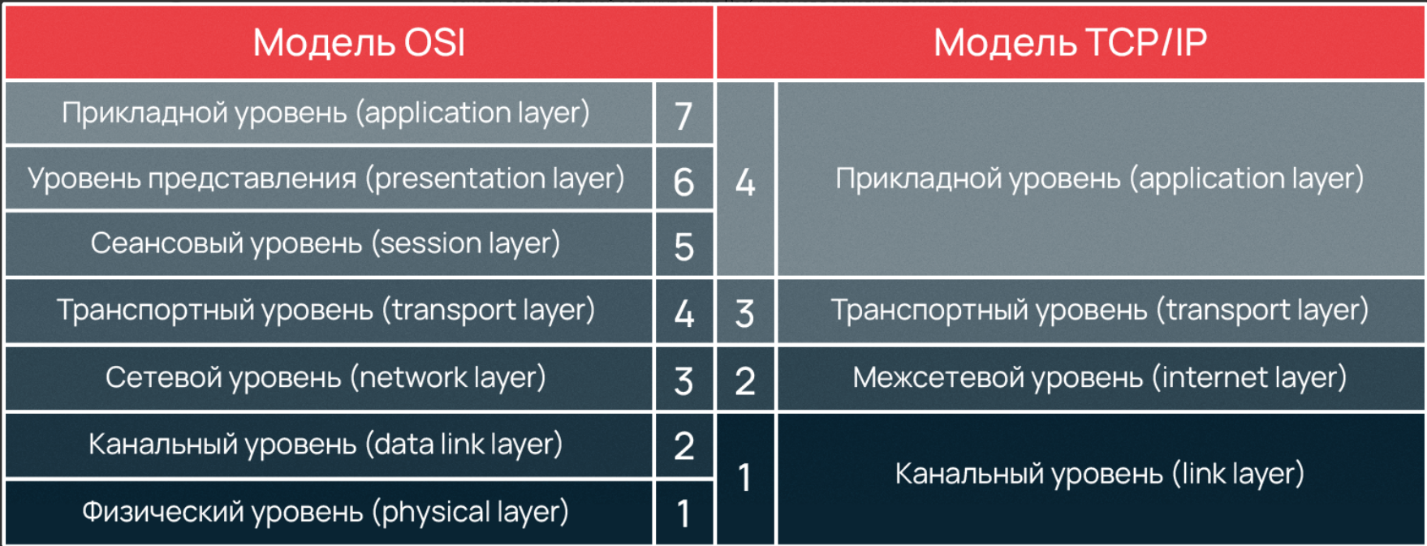
## Шестой уровень, представления данных: Шестой уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Шестой уровень также занимается представлением картинок (в JPEG, GIF и т.д.), а также видео-аудио (в MPEG, QuickTime). А помимо этого → шифрованием данных, когда при передаче их необходимо защитить.

1. **Седьмой уровень, прикладной:** Седьмой уровень иногда еще называют уровень приложений, но чтобы не запутаться можно использовать оригинальное название — application layer. Прикладной уровень — это то, с чем взаимодействуют пользователи, своего рода графический интерфейс всей модели OSI, с другими он взаимодействует по минимуму.

Все услуги, получаемые седьмым уровнем от других, используются для доставки данных до пользователя.

# **Cтек протоколов TCP/IP**

Стек протоколов TCP/IP (Transmission Control Protocol/Internet Protocol, протокол управления передачей/протокол интернета) — сетевая модель, описывающая процесс передачи цифровых данных.  Он используется в качестве основы для глобальной сети интернет.



### **Канальный уровень:**на канальном уровне самым распространенным протоколом является Ethernet

* В сети, по стандарту Ethernet, устройства отправителя и адресата имеют определенный MAC-адрес — идентификатор «железа». MAC-адрес инкапсулируется в Ethernet вместе с типом передаваемых данных и самими данными

1. **Межсетевой уровень:** межсетевое взаимодействие — это основной принцип построения интернета. Локальные сети по всему миру объединены в глобальную, а передачу данных между этими сетями осуществляют магистральные и пограничные маршрутизаторы.

Именно на межсетевом уровне функционирует **протокол IP**, позволивший объединить разные сети в глобальную.

### **Маска подсети и IP-адреса**

Маска подсети помогает маршрутизатору понять, как и куда передавать пакет.



### **Протокол IP (Internet Protocol)** используется маршрутизатором, чтобы определить, к какой подсети принадлежит получатель. Свой уникальный IP-адрес есть у каждого сетевого устройства, при этом в глобальной сети не может существовать два устройства с одинаковым IP.

### IP предназначен для определения адресата и доставки ему информации.

Протокол **IPv4** предусматривает назначение каждому устройству 32-битного IP-адреса, что ограничивало максимально возможное число уникальных адресов 4 миллиардами (2^32). В более привычном для человека десятичном виде IPv4 выглядит как четыре блока (октета) чисел от 0 до 255, разделенных тремя точками. Первый октет IP-адреса означает класс сети, классов всего 5: A, B, C, D, E. При этом адреса сети D являются мультикастовыми, а сети E вообще не используются.

Рассмотрим, например, IPv4 адрес класса С 223.135.100.7. Первые три октета определяют класс и номер сети, а последний означает номер конечного устройства

Протокол **IPv6** (IP version 6, версии 6), который использует 128-битные адреса и позволяет назначить уникальные адреса для 2^128 устройств.

IPv6 имеет вид восьми блоков по четыре шестнадцатеричных значения, а каждый блок разделяется двоеточием. IPv6 выглядит следующим образом:

**2dab:ffff:0000:0000:01aa:00ff:dd72:2c4a.**

Так как IPv6 адреса длинные, их разрешается сокращать по определенным правилам, которые также описываются [RFC](https://www.rfc-editor.org/rfc/rfc5952).

* 1. Информация о том, какой протокол инкапсулируется, отражается в заголовке IP-пакета. Так, ICMP будет обозначен числом 1, а IGMP будет обозначен числом 2.

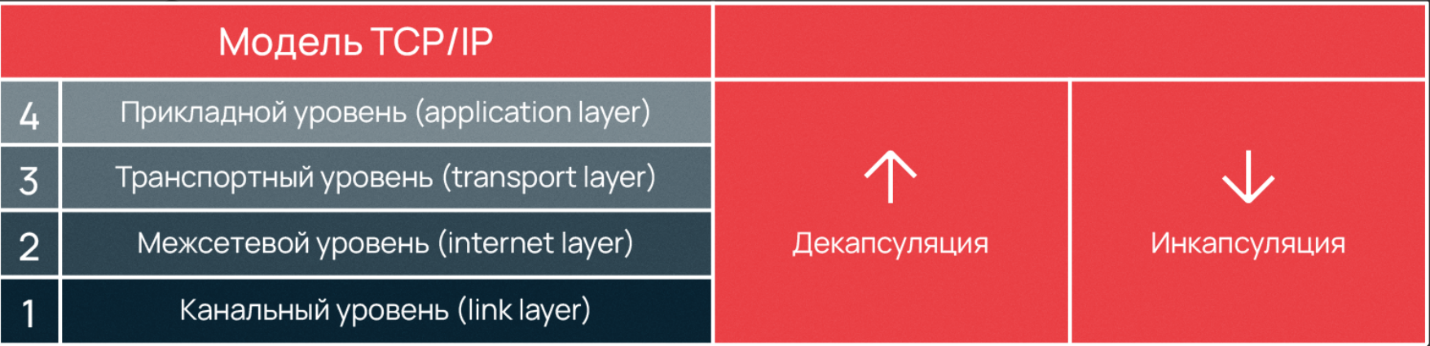
### **Транспортный уровень:** протоколы TCP и UDP, они занимаются доставкой информации.

**TCP (протокол управления передачей)** — **надежный,** он обеспечивает передачу информации, проверяя дошла ли она, насколько полным является объем полученной информации и т.д. TCP дает возможность двум конечным устройствам производить обмен пакетами через предварительно установленное соединение. Пакеты в TSP называются сегментами.

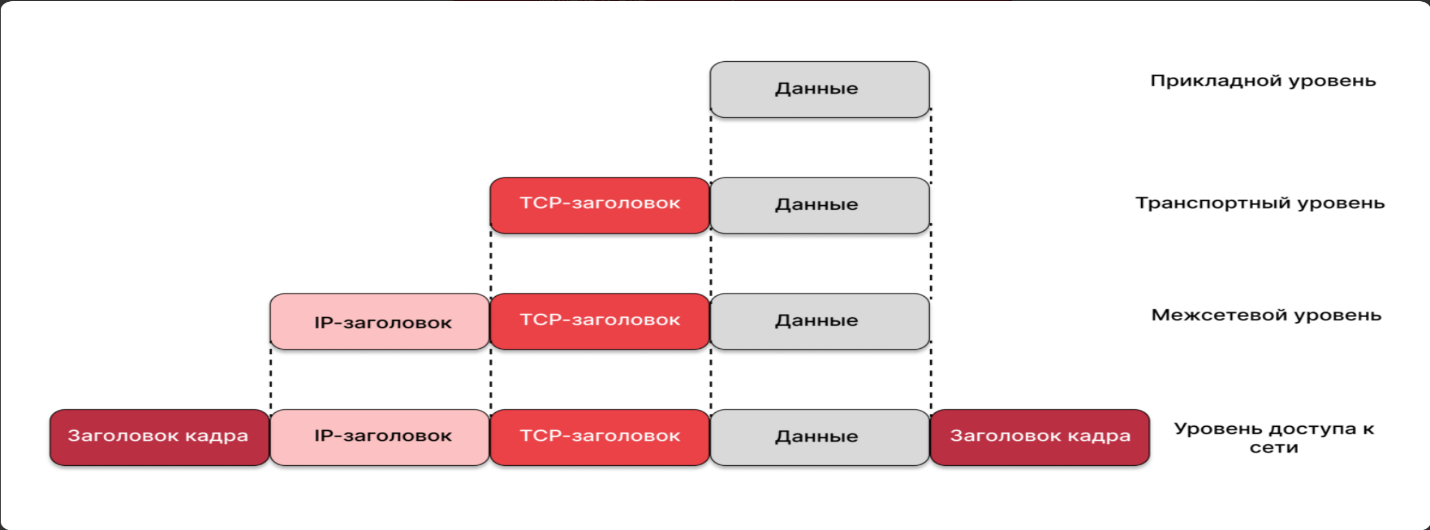
**UDP (протокол пользовательских датаграмм)** — ненадежный, он занимается передачей автономных датаграмм. UDP не гарантирует, что всех датаграммы дойдут до получателя. Датаграммы уже содержат всю необходимую информацию, чтобы дойти до получателя, но они все равно могут быть потеряны или доставлены в порядке отличном от порядка при отправлении. (в видеоиграх или потоковой передаче видео.)

### **Прикладной уровень (application layer)**

Протоколы прикладного уровня действуют для большинства приложений, они предоставляют услуги пользователю или обмениваются данными с «коллегами» с нижних уровней по уже установленным соединениям. Здесь для большинства приложений созданы свои протоколы. Например, браузеры используют HTTP для передачи гипертекста по сети, почтовые клиенты — SMTP для передачи почты, FTP-клиенты — протокол FTP для передачи файлов, службы DHCP — протокол назначения IP-адресов DHCP и так далее.



Для обеспечения корректной работы протоколов различных уровней в сетевых моделях используется специальный метод — **инкапсуляция**. Суть этого метода заключается в добавлении заголовка протокола текущего уровня к данным, полученным от протокола вышестоящего уровня.



**FTP**  [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F)) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *File Transfer Protocol*) — [протокол передачи](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) по [сети](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C). Протокол построен на архитектуре «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)» и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль [открытым текстом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82), или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол [SSH](https://ru.wikipedia.org/wiki/SSH) для безопасной передачи, шифрующей (скрывающей) логин и пароль, а также содержимое файла.

**SFTP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Secure File Transfer Protocol*) — [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) передачи файлов, работающий поверх безопасного канала[[1]](https://ru.wikipedia.org/wiki/SFTP#cite_note-1). Предназначен для копирования и выполнения других операций с файлами поверх надёжного и безопасного соединения. Протокол разработан группой [IETF](https://ru.wikipedia.org/wiki/IETF) как расширение к [SSH-2](https://ru.wikipedia.org/wiki/SSH), однако SFTP допускает реализацию и с использованием иных [протоколов сеансового уровня](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%B0%D0%BD%D1%81%D0%BE%D0%B2%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C).

SFTP не следует путать с [FTPS](https://ru.wikipedia.org/wiki/FTPS) и [FTP через SSH](https://ru.wikipedia.org/wiki/FTP#FTP_%D1%87%D0%B5%D1%80%D0%B5%D0%B7_SSH_(%D0%BD%D0%B5_SFTP)), это скорее новый протокол, разработанный с нуля.

Этот протокол предполагает, что он работает по защищенному каналу, например SSH, что сервер уже аутентифицировал клиента и что протоколу доступна информация пользователя.

**HTTP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *HyperText Transfer Protocol* — «протокол передачи [гипертекста](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82)») сетевой протокол[прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F)

Для получения информации с [веб-сайтов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82).

Основой HTTP является [технология «клиент-сервер»](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), то есть предполагается существование:

* Потребителей ([клиентов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9))), которые инициируют соединение и посылают запрос;
* Поставщиков ([серверов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5))), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

HTTP используется также в качестве **«транспорта»** для других протоколов прикладного уровня, таких как [SOAP](https://ru.wikipedia.org/wiki/SOAP), [XML-RPC](https://ru.wikipedia.org/wiki/XML-RPC), [WebDAV](https://ru.wikipedia.org/wiki/WebDAV).

**Основные преимущества HTTP/2**

**HTTP/2** предлагает ряд значительных улучшений по сравнению с его предшественником, HTTP/1.1. Вот некоторые из ключевых преимуществ:

**Мультиплексирование**: Это позволяет отправлять множество запросов и ответов одновременно через одно TCP-соединение. Это устраняет проблему "очереди головы" (head-of-line blocking), когда последующие запросы не могут быть отправлены, пока не завершится обработка предыдущего.

**Сжатие заголовков**: HTTP/2 сокращает ненужный и повторяющийся размер заголовков, что уменьшает общий объем передаваемых данных.

**Server Push**: Эта функция позволяет серверам "предсказывать" ресурсы, которые клиенту понадобятся далее, и отправлять их заранее, уменьшая время ожидания.

**Приоритизация запросов**: Клиенты могут указывать приоритеты запросов, что позволяет серверу оптимизировать ответы, сначала отправляя более важные ресурсы.

**Безопасность**: Хотя HTTP/2 сам по себе не требует использования HTTPS, большинство клиентских реализаций (например, веб-браузеров) делают это обязательным, что повышает безопасность передачи данных.

**Как работает HTTP/2**

* **HTTP/2** использует бинарный протокол вместо текстового, что делает его более эффективным и устойчивым к ошибкам разбора. Бинарный протокол легче анализировать, меньше подвержен ошибкам и позволяет компактнее кодировать данные.
* Мультиплексирование в HTTP/2 позволяет обрабатывать множество запросов и ответов параллельно, что значительно уменьшает задержки, связанные с построением соединений и ожиданием ответов. Это особенно полезно для веб-страниц с большим количеством элементов, таких как изображения, стили и скрипты.
* Сжатие заголовков с помощью алгоритма HPACK уменьшает размер заголовков, что ускоряет их передачу. Server Push позволяет серверу активно отправлять ресурсы клиенту, даже если клиент еще не запросил их, что может ускорить загрузку страниц.
* Внедрение HTTP/2 может потребовать обновления серверного программного обеспечения и, в некоторых случаях, перехода на HTTPS, если это еще не было сделано.
* Однако разработка не стоит на месте, и уже существует **HTTP/3** (также известный как QUIC), который обещает еще большую скорость и надежность за счет использования протокола UDP вместо TCP.

# **Коды ответов http и ошибки сервера**

**1xx –**[**Информационные**](https://mixtelecom.ru/kody-otvetov-i-oshibki-servera#1xx) **2xx –**[**Успешные коды ответа**](https://mixtelecom.ru/kody-otvetov-i-oshibki-servera#2xx) **3xx –**[**Коды редиректов**](https://mixtelecom.ru/kody-otvetov-i-oshibki-servera#3xx) **4xx –**[**Ошибки со стороны клиента**](https://mixtelecom.ru/kody-otvetov-i-oshibki-servera#4xx) **5xx –**[**Ошибки со стороны сервера**](https://mixtelecom.ru/kody-otvetov-i-oshibki-servera#5xx)

**Информационные коды:**

**Данная группа отвечает за передачу данных. Коды этого типа свидетельствуют о том, что запрос принят сервером и обрабатывается.**

**100 — Continue — Временный код ответа, означающий начало принятия запроса к его последующей обработке.**

**101 — Switching Protocols — Сообщает о переключении сервера на протокол, которые был указан в заголовке Upgrade запроса клиента.**

**102 — Processing — Информация о том, что запрос принят сервером и находится в обработке, но этот процесс еще не завершен.**

**103 — Early Hints — Используется для предварительной загрузки данных пока сервер формирует полный ответ.**

## Успешная обработка запроса

Коды группы сообщают, что запрос не только принят сервером, но и успешно обработан.

**200** — OK — Один из самых популярных ответов. Он свидетельствует о том, что обмен данными между клиентом и сервером прошли успешно. Именно код ответа 200 ждут от ресурса, чтобы проверить, что все работает как надо: сайт загружен, файл открывается и т.д.

**201** — Created — Информирует об успешном создании нового ресурса в результате выполнения запроса. Например, была создана новая страница. Если сервер по каким-то причинам не смог обработать запрос и ресурс не был создан, то код ответа будет 202.

**202** — Accepted — Сообщает, что сервер принял запрос, но не завершил его обработку.

**203** — Non-Authoritative Information — Отвечает об успешной обработке запроса с оговоркой на то, что передаваемая информация была предоставлена не из исходного сервера, а другого источника (например, резервной копии) и может быть неактуальной.

**204** — No Content — Сообщает об успешном принятии и обработке запроса, а также о том, что у сервера нет содержимого для отправки пользователю.

**205** — Reset Content — Сервер передает пользователю ответ в виде требований к сбросу введенных данных. Например, о необходимости очистить форму с заполненными до этого данными.

**206** — Partial Content — Свидетельствует о частичном выполнении GET-запроса сервером, возвращая только запрошенную пользователем часть контента. Этот код встречается при использовании кэширования.

## Коды редиректов

Данная группа кодов состояния сообщает о перенаправлении пользователя с его согласием или без него.

**300** — Multiple Choices — Ответ срабатывает при условии, что по указанному запросу существует несколько вариантов URL. При таком варианте пользователь или User-agent должен выбрать альтернативный адрес.

**301** — Moved Permanently — Свидетельствует о перемещении ранее проиндексированного URL на новый адрес. Это команда роботу индексировать новую страницу вместо старой.

**302 — Found, 302 Moved Temporarily** — Сообщает, что ранее проиндексированный URL был временно перемещен по другому адресу. При этом страница остается в индексе, а в ответе указывается новый адрес запрашиваемого URL.

**303 — See Other** — Указывает пользователю, что запрошенная страница находится по другому адресу с запросом GET.

**304 — Not Modified** — Показывает, что запрашиваемая страница или объект не были изменены с момента последнего обновления кэша данного документа.

**305 — Use Proxy** — Сообщает пользователю, что запрашиваемый ресурс доступен только через прокси. Данные по прокси указаны в ответе сервера.

**307 — Temporary Redirect** — Код схож с 302, сообщая о временном перемещении ресурса на другой адрес. Разница заключается в способе обращения к ресурсу, который должен быть получен тем же методом, что и предыдущий запрос.

## Ошибки клиента

Коды состояний данной группы сообщают об ошибках клиента, при которых сервер не может вызвать запрашиваемый результат.

**400** — **Bad Request** — Ошибка свидетельствует от том, что сервер не понял запрос пользователя из-за синтаксической ошибки.

**401** — **Unauthorized** — Сообщает о необходимости быть авторизованным для получения запрашиваемого доступа. Возникает при неправильном вводе данных пользователем при авторизации.

**403** — **Forbidden** — Запрет доступа к запрашиваемой странице. Доступ может быть ограничен настройками индексации или запрещен для определенных IP.

**[404 Not Found](https://ru.wikipedia.org/wiki/%D0%9E%D1%88%D0%B8%D0%B1%D0%BA%D0%B0_404)** — Пожалуй, самая распространенная ошибка. Сообщает о том, что запрашиваемая страница не найдена. Самая частая причина — ошибка в написании адреса.

## Ошибки сервера

В эту группу входят коды ошибок со стороны сервера, когда по тем или иным причинам он не способен обработать запрос или выполнить требуемую операцию.

**500**— **Internal Server Error** — Код оповещает о возникшей внутренней ошибке сервера или его аварийном отказе.

**501** — **Not Implemented** — Сервер столкнулся с запросом, который не смог распознать. Либо запрос не поддерживается и не может быть обработан.

**502** — **Bad Gateway** — Сообщает о неправильном получении ответа вышестоящего сервера. Частая причина — несогласованные протоколов между шлюзом и сервером (ошибки DNS, прокси, хостинга).

**503 — Service Unavailable** — Указывает на временную недоступность сервера. Причиной может быть его перезагрузка, техническое обслуживание, обращение слишком большого количества пользователей при наличии подобных ограничений. Как правило, сообщение об ошибке содержит параметр Retry-After, информирующий о времени восстановления штатной работы ресурса.

**504 — Gateway Time-out** — Истек срок ожидания ответа от вышестоящего сервера. Возможные причины: недостаток ресурсов, неполадки с сетевым соединением, ошибки HTTP протокола, настроен слишком короткий срок ожидания.

**505 — HTTP Version Not Supported** — Используемая в запросе версия протокола HTTP не поддерживается сервером. Встречается при использовании устаревшего формата HTTP-протокола.

**HTTPS -** (аббр. от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *HyperText Transfer Protocol Secure*) — расширение [протокола](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [HTTP](https://ru.wikipedia.org/wiki/HTTP) для поддержки [шифрования](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) в целях повышения безопасности. Данные в протоколе HTTPS передаются поверх криптографических протоколов [TLS](https://ru.wikipedia.org/wiki/TLS) или устаревшего в 2015 году [SSL](https://ru.wikipedia.org/wiki/SSL)[[1]](https://ru.wikipedia.org/wiki/HTTPS#cite_note-1). В отличие от HTTP с [TCP](https://ru.wikipedia.org/wiki/TCP)-[портом](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_(TCP/IP)) 80, для HTTPS по умолчанию используется TCP-порт 443[[2]](https://ru.wikipedia.org/wiki/HTTPS#cite_note-%D0%B0%D0%B2%D1%82%D0%BE%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B01-2).

HTTPS не является отдельным протоколом. Это обычный HTTP, работающий через шифрованные транспортные механизмы [SSL](https://ru.wikipedia.org/wiki/SSL) и [TLS](https://ru.wikipedia.org/wiki/TLS)[[4]](https://ru.wikipedia.org/wiki/HTTPS#cite_note-4). Он обеспечивает защиту от атак, основанных на прослушивании сетевого соединения.

По умолчанию HTTPS URL использует 443 [TCP](https://ru.wikipedia.org/wiki/TCP)-[порт](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_(TCP/IP)) (для незащищённого HTTP — 80)[[2]](https://ru.wikipedia.org/wiki/HTTPS#cite_note-%D0%B0%D0%B2%D1%82%D0%BE%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B01-2). Чтобы подготовить веб-сервер для обработки https-соединений, администратор должен получить и установить в систему сертификат открытого и закрытого ключа для этого веб-сервера.

**SMTP -**  (прикладной) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Simple Mail Transfer Protocol* — простой протокол передачи почты) — это широко используемый [сетевой протокол](https://ru.wikipedia.org/wiki/IP), предназначенный **для передачи** исходящей [электронной почты](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0) в сетях [TCP](https://ru.wikipedia.org/wiki/TCP/IP)/IP, используя *25-й TCP порт.*

**IMAP -** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Internet Message Access Protocol*) — [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) для доступа к [электронной почте](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0).

Базируется на [транспортном протоколе](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [TCP](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol) и использует [порт](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8)) 143, а IMAPS (IMAP поверх [SSL](https://ru.wikipedia.org/wiki/SSL)) — порт 993. IMAP работает только с [сообщениями](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B5) и не требует каких-либо [пакетов](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BA%D0%B5%D1%82_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D1%8B%D0%B5_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8)) со специальными заголовками[[1]](https://ru.wikipedia.org/wiki/IMAP#cite_note-_a7b1ccd14fe6e0f4-1).

IMAP предоставляет пользователю широкие возможности для работы с [почтовыми ящиками](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0), находящимися на почтовом [сервере](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)). [Почтовая программа](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), использующая этот [протокол](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D1%81%D0%B5%D0%B0%D0%BD%D1%81%D0%B0), получает доступ к хранилищу [корреспонденции](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%80%D0%B5%D1%81%D0%BF%D0%BE%D0%BD%D0%B4%D0%B5%D0%BD%D1%86%D0%B8%D1%8F) на [сервере](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) так, как будто эта корреспонденция расположена на [компьютере](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) получателя. [Электронными письмами](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D0%B8%D1%81%D1%8C%D0%BC%D0%BE) можно манипулировать с [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) пользователя ([клиента](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0))) без постоянной пересылки с [сервера](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) и обратно полного содержания писем.

**PPP - Point-to-Point** (прикладной) (от точки к точке, двухточечный) протоколе, также известном как PPP. PPP уникален по своим функциям, он применяется для коммуникации между двумя маршрутизаторами без участия хоста или какой-либо сетевой структуры в промежутке. При необходимости PPP обеспечивает аутентификацию, шифрование, а также сжатие данных. Он широко используется при построении физических сетей, например, кабельных телефонных, сотовых телефонных, сетей по кабелю последовательной передачи и транк-линий (когда один маршрутизатор подключают к другому для увеличения размера сети).

У PPP есть два подвида — PPPoE (PPP по Ethernet) и PPPoA (PPP через асинхронный способ передачи данных — ATM), интернет-провайдеры часто их используют для DSL соединений.

PPP и его старший аналог SLIP (протокол последовательной межсетевой связи) формально относятся к межсетевому уровню TCP/IP, но в силу особого принципа работы, иногда выделяются в отдельную категорию. Преимущество PPP в том, что для установки соединения не требуется сетевая инфраструктура, а необходимость маршрутизаторов отпадает. Эти факторы обуславливают специфику использования PPP протоколов.

**TELNET -**  [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) (сокр. от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***Tel****etype****Net****work*) — [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) для реализации текстового [терминального интерфейса](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB) по [сети](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) при помощи транспорта [TCP](https://ru.wikipedia.org/wiki/TCP).

Протокол telnet использовался для [удалённого администрирования](https://ru.wikipedia.org/wiki/%D0%A3%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) различными [сетевыми устройствами](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D1%8B%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0) и [программными серверами](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)), но уступил [SSH](https://ru.wikipedia.org/wiki/SSH) (удалённое управление операционной системой) из-за безопасности. Тем не менее может являться единственной возможностью взаимодействовать через [CLI](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9_%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8) (интерфейс командной строки) со [встраиваемыми системами](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D1%82%D1%80%D0%B0%D0%B8%D0%B2%D0%B0%D0%B5%D0%BC%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), например, с [маршрутизаторами](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80), так как на них отсутствует ssh.

**ICMP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Internet Control Message Protocol* —[протокол прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F), межсетевых управляющих сообщений[[1]](https://ru.wikipedia.org/wiki/ICMP#cite_note-1)) — [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), входящий в [стек протоколов TCP/IP](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BA_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%BE%D0%B2_TCP/IP). В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуаций, возникших при передаче данных, например, запрашиваемая услуга недоступна или хост или [маршрутизатор](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80) не отвечают. Также на ICMP возлагаются некоторые сервисные функции (services).

ICMP основан на протоколе IP. Каждое ICMP-сообщение инкапсулируется непосредственно в пределах одного [IP](https://ru.wikipedia.org/wiki/IP)-пакета, и, таким образом, как и [UDP](https://ru.wikipedia.org/wiki/UDP) и в отличие от [TCP](https://ru.wikipedia.org/wiki/TCP), ICMP является т. н. «ненадежным» (не контролирующим доставку и её правильность)

**SSH** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***S****ecure****Sh****ell* — «безопасная оболочка»[[1]](https://ru.wikipedia.org/wiki/SSH#cite_note-1)) — [сетевой протокол прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F), позволяющий производить удалённое управление [операционной системой](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) и [туннелирование](https://ru.wikipedia.org/wiki/%D0%A2%D1%83%D0%BD%D0%BD%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8)) [TCP](https://ru.wikipedia.org/wiki/TCP)-соединений (например, для передачи файлов). Схож по функциональности с протоколами [Telnet](https://ru.wikipedia.org/wiki/Telnet" \o "Telnet) и [rlogin](https://ru.wikipedia.org/wiki/Rlogin" \o "Rlogin), но, в отличие от них, [шифрует](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) весь трафик, включая и передаваемые [пароли](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%BE%D0%BB%D1%8C). SSH допускает выбор различных алгоритмов шифрования. SSH-[клиенты](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) и SSH-[серверы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищённой среде практически любой другой [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB). Таким образом, можно не только удалённо работать на компьютере через [командную оболочку](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BE%D0%BB%D0%BE%D1%87%D0%BA%D0%B0), но и передавать по шифрованному каналу звуковой поток или видео (например, с [веб-камеры](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BA%D0%B0%D0%BC%D0%B5%D1%80%D1%8B))[[2]](https://ru.wikipedia.org/wiki/SSH#cite_note-2). Также SSH может использовать [сжатие](https://ru.wikipedia.org/wiki/%D0%A1%D0%B6%D0%B0%D1%82%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов [X Window System](https://ru.wikipedia.org/wiki/X_Window_System).

SSH-сервер обычно прослушивает соединения на TCP-порту 22

**DNS –** прикладной уровень **(англ. )** — распределённая система (распределённая база данных), способная по запросу, содержащему доменное имя хоста ([компьютера](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) или другого сетевого устройства), сообщить [IP](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:IP) адрес или (в зависимости от запроса) другую информацию. *(преобразует доменное имя в IP адрес и наоборот)*

DNS работает в сетях [TCP/IP](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:TCP/IP). Как частный случай, DNS может хранить и обрабатывать и обратные запросы, *определения имени хоста по его IP адресу* — IP адрес по таблице соответствия преобразуется в доменное имя, и посылается запрос на информацию типа «PTR».

**DHCP -** прикладной уровень([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Dynamic Host Configuration Protocol* — протокол динамической настройки [узла](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82)) — [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), позволяющий [сетевым устройствам](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE) автоматически получать [IP-адрес](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) и другие параметры, необходимые для работы в сети [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP). Данный протокол работает по модели «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)».  Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому *серверу DHCP* и получает от него нужные параметры. [Сетевой администратор](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80) может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

**POP3** (Post Office Protocol v. 3) — протокол прикладного уровня, используемый для получения сообщений с сервера электронной почты. IMAP (Internet Message Access Protocol) — протокол, который используют для доступа к сообщениям, хранящимся на сервере электронной почты.