**HTTP** – прикладной протокол передачи данный, используемый для получения информации с веб-сайтов.  *HyperText Transfer Protocol*

## Структура протокола[[править](https://ru.wikipedia.org/w/index.php?title=HTTP&veaction=edit&vesection=6) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=HTTP&action=edit&section=6)]

Каждое HTTP-сообщение состоит из трёх частей, которые передаются в указанном порядке:

1. Стартовая строка ([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) *Starting line*) — определяет тип сообщения;
2. Заголовки ([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) *Headers*) — характеризуют тело сообщения, параметры передачи и прочие сведения;
3. Тело сообщения ([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) *Message Body*) — непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

Заголовки и тело сообщения могут отсутствовать, но стартовая строка является обязательным элементом, так как указывает на тип запроса/ответа. Исключением является версия 0.9 протокола, у которой сообщение запроса содержит только стартовую строку, а сообщения ответа только тело сообщения.

Для версии протокола 1.1 сообщение запроса обязательно должно содержать заголовок *Host*.

**HTTPS**- расширение протокола HTTP, поддерживающее шифрование по протоколам SSL и TLS.

Еще одно техническое отличие – в портах, используемых для доступа по протоколу HTTP и HTTPS. Первый обычно взаимодействует с портом 80, второй – с портом 443.

1. HTTP – непосредственно протокол передачи данных, HTTPS – расширение этого протокола.
2. HTTPS используется для защищенного посредством шифрования  обмена данными.
3. HTTPS применяется в том числе и для авторизации на серверах, требующих повышенного внимания к безопасности данных.
4. HTTP работает **с портом 80,** HTTPS **– с портом 443.**
5. 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). Он обеспечивает защиту от атак, основанных на прослушивании сетевого соединения — от [снифферских](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) атак и атак типа [man-in-the-middle](https://ru.wikipedia.org/wiki/%D0%A7%D0%B5%D0%BB%D0%BE%D0%B2%D0%B5%D0%BA_%D0%BF%D0%BE%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5), при условии, что будут использоваться шифрующие средства и *сертификат сервера проверен и ему доверяют*.

**Прокси-сервер** (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) *proxy* — «представитель, уполномоченный») — сервер (комплекс программ) в [компьютерных сетях](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), позволяющий [клиентам](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)) выполнять косвенные запросы к другим сетевым службам. Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс (например, [e-mail](https://ru.wikipedia.org/wiki/E-mail)), расположенный на другом [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)). Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного [кэша](https://ru.wikipedia.org/wiki/%D0%9A%D1%8D%D1%88) (в случаях, если прокси имеет свой кэш). В некоторых случаях запрос клиента или ответ сервера может быть изменён прокси-сервером в определённых целях. Прокси-сервер позволяет защищать компьютер клиента от некоторых сетевых атак и помогает сохранять анонимность клиента.

## Использование[[править](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%BA%D1%81%D0%B8-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80&veaction=edit&vesection=1) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%BA%D1%81%D0%B8-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80&action=edit&section=1)]

Чаще всего прокси-серверы применяются для следующих целей:

* Обеспечение доступа компьютеров локальной сети к сети Интернет.
* [Кэширование](https://ru.wikipedia.org/wiki/%D0%9A%D1%8D%D1%88) данных: если часто происходят обращения к одним и тем же внешним ресурсам, то можно держать их копию на прокси-сервере и выдавать по запросу, снижая тем самым нагрузку на канал во внешнюю сеть и ускоряя получение клиентом запрошенной информации. С развитием динамического контента кэширование утратило актуальность.
* Сжатие данных: прокси-сервер загружает информацию из Интернета и передаёт информацию конечному пользователю в сжатом виде. Такие прокси-серверы используются в основном с целью экономии внешнего [сетевого трафика](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA) клиента или внутреннего — компании, в которой установлен прокси-сервер.
* Защита [локальной сети](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B2%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) от внешнего доступа: например, можно настроить прокси-сервер так, что локальные компьютеры будут обращаться к внешним ресурсам только через него, а внешние компьютеры не смогут обращаться к локальным вообще (они «видят» только прокси-сервер). См. также [*NAT*](https://ru.wikipedia.org/wiki/NAT).
* Ограничение доступа из локальной сети к внешней: например, можно запретить доступ к определённым [веб-сайтам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82), ограничить использование интернета каким-то локальным пользователям, устанавливать квоты на трафик или полосу пропускания, фильтровать рекламу и [вирусы](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%B5_%D0%B2%D0%B8%D1%80%D1%83%D1%81%D1%8B).
* [Анонимизация](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%B0%D0%B9%D0%B7%D0%B5%D1%80) доступа к различным ресурсам. Прокси-сервер может скрывать сведения об источнике запроса или пользователе. В таком случае целевой сервер видит лишь информацию о прокси-сервере, например, [IP-адрес](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81), но не имеет возможности определить истинный источник запроса. Существуют также *искажающие прокси-серверы*, которые передают целевому серверу ложную информацию об истинном пользователе.
* Обход ограничений доступа. Прокси-серверы популярны среди пользователей стран, где доступ к некоторым ресурсам ограничен законодательно и фильтруется.

Прокси-сервер, к которому может получить доступ любой пользователь сети интернет, называется [открытым](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D0%BA%D1%81%D0%B8).

**Прозрачный прокси** — схема связи, при которой трафик, или его часть, перенаправляется на прокси-сервер неявно (средствами [маршрутизатора](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)). При этом клиент может использовать все преимущества прокси-сервера без дополнительных настроек браузера (или другого приложения для работы с интернетом). *Пример: route -p add 10.32.5.5 mask 255.255.255.255 10.32.1.14*

[**Обратный прокси**](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D0%BA%D1%81%D0%B8) — прокси-сервер, который в отличие от прямого, ретранслирует запросы клиентов из внешней сети на один или несколько серверов, логически расположенных во внутренней сети. Часто используется для балансировки сетевой нагрузки между несколькими [веб-серверами](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и повышения их безопасности, играя при этом роль [межсетевого экрана](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%8D%D0%BA%D1%80%D0%B0%D0%BD) на [прикладном уровне](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI).

Классификация прокси-серверов для целей анонимизации представлена в статье [Веб-прокси](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%BE%D0%BA%D1%81%D0%B8).

**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](https://ru.wikipedia.org/wiki/%D0%A3%D0%B4%D0%B0%D0%BB%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) [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) и [rlogin](https://ru.wikipedia.org/wiki/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](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA), 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](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) 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. **port 22**

**FTP** ([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)) — стандартный [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), предназначенный для передачи файлов по TCP-сетям (например, Интернет). **Использует 21-й порт**. FTP часто используется для загрузки сетевых страниц и других документов с частного устройства разработки на открытые [сервера](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)) [хостинга](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3).

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.

**SMTP** ([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 [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/IP.

SMTP впервые был описан в [RFC 821](https://tools.ietf.org/html/rfc821) (1982 год); последнее обновление в [RFC 5321](https://tools.ietf.org/html/rfc5321) (2008) включает масштабируемое расширение — **ESMTP** ([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)  *Extended SMTP*). В настоящее время под «протоколом SMTP», как правило, подразумевают и его расширения. Протокол SMTP предназначен для передачи исходящей почты с использованием [**порта**](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_(TCP/IP))[**TCP**](https://ru.wikipedia.org/wiki/TCP)**25.**

**DNS** ([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) *Domain Name System* — система доменных имён) — компьютерная [распределённая система](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) для получения информации о [доменах](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BC%D1%8F). Чаще всего используется для получения IP-адреса по имени [хоста](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82) (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене ([SRV-запись](https://ru.wikipedia.org/wiki/SRV-%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C)).

[Распределённая база данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) DNS поддерживается с помощью иерархии [DNS-серверов](https://ru.wikipedia.org/wiki/DNS-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), взаимодействующих по определённому [протоколу](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).

Основой DNS является представление об иерархической структуре [доменного имени](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BC%D1%8F) и *зонах*. Каждый сервер, отвечающий за имя, может *делегировать* ответственность за дальнейшую часть домена другому серверу (с административной точки зрения — другой организации или человеку), что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени. **use TCP port 53 if UDP port 53**

**Доме́нное и́мя** — символьное имя, служащее для [идентификации](https://ru.wikipedia.org/wiki/%D0%98%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B)) областей — единиц административной [автономии](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D1%8F) в сети [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) — в составе вышестоящей по[иерархии](https://ru.wikipedia.org/wiki/%D0%98%D0%B5%D1%80%D0%B0%D1%80%D1%85%D0%B8%D1%8F) такой области. Каждая из таких областей называется ***доме́ном***. Общее [пространство имён](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%B8%D0%BC%D1%91%D0%BD) Интернета функционирует благодаря [DNS](https://ru.wikipedia.org/wiki/DNS) — системе доменных имён. Доменные имена дают возможность адресации интернет-[узлов](https://ru.wikipedia.org/wiki/%D0%A3%D0%B7%D0%B5%D0%BB_%D1%81%D0%B5%D1%82%D0%B8) и расположенных на них сетевых ресурсов ([веб-сайтов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82), [серверов электронной почты](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), других служб) в удобной для человека форме.

*Доме́нная зона* — совокупность доменных имён определённого уровня, входящих в конкретный домен. Например, зона wikipedia.org включает все доменные имена третьего уровня в этом домене. Термин «доменная зона» в основном применяется в технической сфере, при настройке [DNS](https://ru.wikipedia.org/wiki/DNS)-серверов (поддержание зоны, делегирование зоны, трансфер зоны).

**Идентифика́ция** в информационных системах — процедура, в результате выполнения которой для субъекта идентификации выявляется его идентификатор, однозначно идентифицирующий этого субъекта в информационной системе. Для выполнения процедуры идентификации в информационной системе субъекту предварительно должен быть назначен соответствующий идентификатор (т.е. проведена регистрация субъекта в информационной системе).

Процедура идентификации напрямую связана с [аутентификацией](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F): субъект проходит процедуру аутентификации, и если аутентификация успешна, то информационная система на основе [факторов аутентификации](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) определяет идентификатор субъекта. При этом достоверность идентификации полностью определяется уровнем достоверности выполненной процедуры аутентификации.

**Аутентифика́ция** ([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) *authentication* от греческого: αὐθεντικός authentikos, «реальный, подлинный» от αὐθέντης authentes, «автор») — процедура проверки подлинности, например:

* проверка подлинности пользователя путём сравнения введённого им пароля с паролем, сохранённым в [базе данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) пользователей;
* подтверждение подлинности [электронного письма](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) путём проверки [цифровой подписи](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D1%8C) письма по [открытому ключу отправителя](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%BC_%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%BC);
* проверка [контрольной суммы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D1%83%D0%BC%D0%BC%D0%B0) [файла](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) на соответствие сумме, заявленной автором этого файла.

**POP3** ([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) *Post Office Protocol Version 3* — [протокол](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) почтового отделения, версия 3) — стандартный [интернет-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), используемый [клиентами электронной почты](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) для получения почты с удаленного [сервера](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) по [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP)-соединению.port 110

**TELNET** (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) ***te****rmina****l 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 интерфейса](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) 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) (в современной форме — при помощи транспорта [TCP](https://ru.wikipedia.org/wiki/TCP)). Название «telnet» имеют также некоторые утилиты, реализующие клиентскую часть протокола. Современный стандарт протокола описан в [RFC 854](https://tools.ietf.org/html/rfc854).port number 23

**RDP** ([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) *Remote Desktop Protocol* — протокол удалённого рабочего стола) — проприетарный протокол прикладного уровня, позаимствованный [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) из купленной у PictureTel (ныне известной как Polycom) телекоммуникационной программы Liveshare Plus (названной впоследствии NetMeeting), использующийся для обеспечения удалённой работы пользователя с сервером, на котором запущен сервис [терминальных подключений](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF). Клиенты существуют практически для всех версий Windows (включая [Windows CE](https://ru.wikipedia.org/wiki/Windows_CE), [Phone](https://ru.wikipedia.org/wiki/Windows_Phone) и [Mobile](https://ru.wikipedia.org/wiki/Windows_Mobile)), [Linux](https://ru.wikipedia.org/wiki/Linux), [FreeBSD](https://ru.wikipedia.org/wiki/FreeBSD), [Mac OS X](https://ru.wikipedia.org/wiki/Mac_OS_X), [iOS](https://ru.wikipedia.org/wiki/IOS), [Android](https://ru.wikipedia.org/wiki/Android), [Symbian](https://ru.wikipedia.org/wiki/Symbian). **По умолчанию используется порт TCP 3389**. Официальное название Майкрософт для клиентского ПО — Remote Desktop Connection или Terminal Services Client (TSC), в частности, клиент в Windows 2k/XP/2003/Vista/2008/7/8/10 называется mstsc.exe.

**TCP** ([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) ***t****ransmission****c****ontrol****p****rotocol* — протокол управления передачей) — один из основных [протоколов передачи данных](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) интернета, предназначенный для управления [передачей данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Сети и подсети, в которых совместно используются протоколы TCP и [IP](https://ru.wikipedia.org/wiki/IP) называются сетями [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP).

В [стеке протоколов IP](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%BE%D0%B2_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) TCP выполняет функции протокола [транспортного уровня](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_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C) [модели OSI](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI).

Механизм TCP предоставляет [поток данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) с предварительной установкой соединения, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым, в отличие от [UDP](https://ru.wikipedia.org/wiki/UDP), целостность передаваемых данных и уведомление отправителя о результатах передачи.

Реализации TCP обычно встроены в [ядра](https://ru.wikipedia.org/wiki/%D0%AF%D0%B4%D1%80%D0%BE_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) [ОС](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). Существуют реализации TCP, работающие в [пространстве пользователя](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F).

Когда осуществляется передача от компьютера к компьютеру через Интернет, TCP работает на верхнем уровне между двумя конечными системами, например, [браузером](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) и веб-сервером. TCP осуществляет надежную передачу потока байтов от одной программы на некотором компьютере к другой программе на другом компьютере (например, программы для электронной почты, для обмена файлами). TCP контролирует длину сообщения, скорость обмена сообщениями, сетевой трафик.

## Well-known ports

**The port numbers in the range from 0 to 1023 are the *well-known ports* or *system ports*.**[**[2]**](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#cite_note-2)**They are used by system processes that provide widely used types of network services. On**[**Unix-like**](https://en.wikipedia.org/wiki/Unix-like)**operating systems, a process must execute with [superuser](https://en.wikipedia.org/wiki/Superuser" \o "Superuser) privileges to be able to bind a**[**network socket**](https://en.wikipedia.org/wiki/Network_socket)**to an**[**IP address**](https://en.wikipedia.org/wiki/IP_address)**using one of the well-known ports.**

## Registered ports[[edit](https://en.wikipedia.org/w/index.php?title=List_of_TCP_and_UDP_port_numbers&action=edit&section=3)]

**The range of port numbers from 1024 to 49151 are the registered ports. They are assigned by**[**IANA**](https://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority)**for specific service upon application by a requesting entity.**[**[1]**](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#cite_note-IANA-1)**On most systems, registered ports can be used by ordinary users.**

## Dynamic, private or ephemeral ports

**The range 49152–65535 (215+214 to 216−1) contains dynamic or private ports that cannot be registered with IANA.**[**[201]**](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#cite_note-201)**This range is used for private, or customized services or temporary purposes and for automatic allocation of**[**ephemeral ports**](https://en.wikipedia.org/wiki/Ephemeral_port)**.**

**UDP** ([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) *User Datagram Protocol* — протокол пользовательских [датаграмм](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D1%82%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)) — один из ключевых элементов [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP), набора сетевых протоколов для [Интернета](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые [датаграммами](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D1%82%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)) другим хостам по [IP-сети](https://ru.wikipedia.org/wiki/IP) без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. Протокол был разработан Дэвидом П. Ридом в 1980 году и официально определён в [RFC 768](https://tools.ietf.org/html/rfc768).

UDP использует простую модель передачи, без неявных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Таким образом, UDP предоставляет ненадёжный сервис, и датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в [системах реального времени](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_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8). При необходимости исправления ошибок на сетевом уровне интерфейса приложение может задействовать [TCP](https://ru.wikipedia.org/wiki/TCP) или [SCTP](https://ru.wikipedia.org/wiki/SCTP), разработанные для этой цели.

Природа UDP как протокола без сохранения состояния также полезна для серверов, отвечающих на небольшие запросы от огромного числа клиентов, например [DNS](https://ru.wikipedia.org/wiki/DNS) и [потоковые мультимедийные приложения](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D0%BE%D0%B5_%D0%BC%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0) вроде [IPTV](https://ru.wikipedia.org/wiki/IPTV), [Voice over IP](https://ru.wikipedia.org/wiki/VoIP), [протоколы туннелирования IP](https://ru.wikipedia.org/wiki/TFTP) и многие [онлайн-игры](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%B8%D0%B3%D1%80%D0%B0).

## Псевдозаголовки[[править](https://ru.wikipedia.org/w/index.php?title=UDP&veaction=edit&vesection=9) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=UDP&action=edit&section=9)]

### Псевдозаголовок для IPv4**[**[**править**](https://ru.wikipedia.org/w/index.php?title=UDP&veaction=edit&vesection=10)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=UDP&action=edit&section=10)**]**

Если UDP работает над IPv4, контрольная сумма вычисляется при помощи псевдозаголовка, который содержит некоторую информацию из заголовка IPv4. Псевдозаголовок не является настоящим IPv4-заголовком, используемым для отправления IP-пакета. В таблице приведён псевдозаголовок, используемый только для вычисления контрольной суммы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Биты** | **0 — 7** | **8 — 15** | **16 — 23** | **24 — 31** |
| **0** | Адрес источника | | | |
| **32** | Адрес получателя | | | |
| **64** | Нули | Протокол | Длина UDP | |
| **96** | Порт источника | | Порт получателя | |
| **128** | Длина | | Контрольная сумма | |
| **160+** | Данные | | | |

Адреса источника и получателя берутся из IPv4-заголовка. Значения поля «Протокол» для UDP равно 17 (0x11). Поле «Длина UDP» соответствует длине заголовка и данных.

Вычисление контрольной суммы для IPv4 необязательно, если она не используется, то значение равно 0.

### Псевдозаголовок для IPv6**[**[**править**](https://ru.wikipedia.org/w/index.php?title=UDP&veaction=edit&vesection=11)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=UDP&action=edit&section=11)**]**

При работе UDP над IPv6 контрольная сумма обязательна. Метод для её вычисления был опубликован в [RFC 2460](https://tools.ietf.org/html/rfc2460):

При вычислении контрольной суммы опять используется псевдозаголовок, имитирующий реальный IPv6-заголовок:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Биты** | **0 — 7** | **8 — 15** | **16 — 23** | **24 — 31** |
| **0** | Адрес источника | | | |
| **32** |
| **64** |
| **96** |
| **128** | Адрес получателя | | | |
| **160** |
| **192** |
| **224** |
| **256** | Длина UDP | | | |
| **288** | Нули | | | Следующий заголовок |
| **320** | Порт источника | | Порт получателя | |
| **352** | Длина | | Контрольная сумма | |
| **384+** | Данные | | | |

Адрес источника такой же, как и в IPv6-заголовке. Адрес получателя — финальный получатель; если в IPv6-пакете не содержится заголовка маршрутизации (Routing), то это будет адрес получателя из IPv6-заголовка, в противном случае, на начальном узле, это будет адрес последнего элемента заголовка маршрутизации, а на узле-получателе — адрес получателя из IPv6-заголовка. Значение «Следующий заголовок» равно значению протокола — 17 для UDP. Длина UDP — длина UDP-заголовка и данных.

## Сравнение UDP и TCP[[править](https://ru.wikipedia.org/w/index.php?title=UDP&veaction=edit&vesection=14) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=UDP&action=edit&section=14)]

*Основная статья:*[***Транспортный уровень***](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_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

[TCP](https://ru.wikipedia.org/wiki/TCP) — ориентированный на соединение протокол, что означает необходимость «рукопожатия» для установки соединения между двумя хостами. Как только соединение установлено, пользователи могут отправлять данные в обоих направлениях.

* *Надёжность* — TCP управляет подтверждением, повторной передачей и тайм-аутом сообщений. Производятся многочисленные попытки доставить сообщение. Если оно потеряется на пути, сервер вновь запросит потерянную часть. В TCP нет ни пропавших данных, ни (в случае многочисленных тайм-аутов) разорванных соединений.
* *Упорядоченность* — если два сообщения последовательно отправлены, первое сообщение достигнет приложения-получателя первым. Если участки данных прибывают в неверном порядке, TCP отправляет неупорядоченные данные в буфер до тех пор, пока все данные не могут быть упорядочены и переданы приложению.
* *Тяжеловесность* — TCP необходимо три пакета для установки сокет-соединения перед тем, как отправить данные. TCP следит за надёжностью и перегрузками.
* *Потоковость* — данные читаются как поток [байтов](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82), не передается никаких особых обозначений для границ сообщения или сегментов.

UDP — более простой, основанный на сообщениях протокол без установления соединения. Протоколы такого типа не устанавливают выделенного соединения между двумя хостами. Связь достигается путём передачи информации в одном направлении от источника к получателю без проверки готовности или состояния получателя. В приложениях для голосовой связи через интернет-протокол (Voice over IP, TCP/IP) UDP имеет преимущество над TCP, в котором любое «рукопожатие» помешало бы хорошей голосовой связи. В VoIP считается, что конечные пользователи в реальном времени предоставят любое необходимое подтверждение о получении сообщения.

* *Ненадёжный* — когда сообщение посылается, неизвестно, достигнет ли оно своего назначения — оно может потеряться по пути. Нет таких понятий, как подтверждение, повторная передача, тайм-аут.
* *Неупорядоченность* — если два сообщения отправлены одному получателю, то порядок их достижения цели не может быть предугадан.
* *Легковесность* — никакого упорядочивания сообщений, никакого отслеживания соединений и т. д. Это небольшой транспортный уровень, разработанный на IP.
* *Датаграммы* — пакеты посылаются по отдельности и проверяются на целостность только если они прибыли. Пакеты имеют определенные границы, которые соблюдаются после получения, то есть операция чтения на сокете-получателе выдаст сообщение таким, каким оно было изначально послано.
* *Нет контроля перегрузок* — UDP сам по себе не избегает перегрузок. Для приложений с большой пропускной способностью возможно вызвать коллапс перегрузок, если только они не реализуют меры контроля на прикладном уровне.

## Well-known ports

**The port numbers in the range from 0 to 1023 are the *well-known ports* or *system ports*.**[**[2]**](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#cite_note-2)**They are used by system processes that provide widely used types of network services. On**[**Unix-like**](https://en.wikipedia.org/wiki/Unix-like)**operating systems, a process must execute with [superuser](https://en.wikipedia.org/wiki/Superuser" \o "Superuser) privileges to be able to bind a**[**network socket**](https://en.wikipedia.org/wiki/Network_socket)**to an**[**IP address**](https://en.wikipedia.org/wiki/IP_address)**using one of the well-known ports.**

## Registered ports[[edit](https://en.wikipedia.org/w/index.php?title=List_of_TCP_and_UDP_port_numbers&action=edit&section=3)]

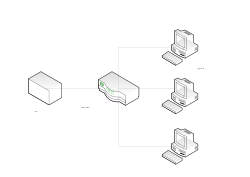
**The range of port numbers from 1024 to 49151 are the registered ports. They are assigned by**[**IANA**](https://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority)**for specific service upon application by a requesting entity.**[**[1]**](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#cite_note-IANA-1)**On most systems, registered ports can be used by ordinary users.**

## Dynamic, private or ephemeral ports

**The range 49152–65535 (215+214 to 216−1) contains dynamic or private ports that cannot be registered with IANA.**[**[201]**](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#cite_note-201)**This range is used for private, or customized services or temporary purposes and for automatic allocation of**[**ephemeral ports**](https://en.wikipedia.org/wiki/Ephemeral_port)**.**

**NAT** (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) *Network Address Translation* — «преобразование сетевых адресов») — это механизм в [сетях](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) [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), позволяющий преобразовывать [IP-адреса](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) транзитных[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)). Также имеет названия *IP Masquerading*, *Network Masquerading* и *Native Address Translation*.

## Применение:IP сеть через единственный IP-адрес[[править](https://ru.wikipedia.org/w/index.php?title=NAT&veaction=edit&vesection=6) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=NAT&action=edit&section=6)]

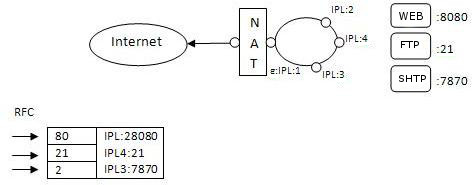
* [](https://commons.wikimedia.org/wiki/File:NAT.svg?uselang=ru)
* На рабочих станциях указанный шлюз по умолчанию или gateway

[Gateway.JPG](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Gateway.JPG)

* Преобразует служебные заголовки, формирует идентичный IP-пакет

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Ip-paket.JPG)

* Публикация локальных ресурсов во внешней IP-сети

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Ip-set.JPG)

• Экономическая выгода вследствие приобретения единственного IP-подключения, а не IP-сети.

• Сокрытие от внешнего наблюдателя структуры внутренней IP-сети.

• Организация системы с распределенной нагрузкой.

• При общем доступе через NAT прозрачно открывается доступ к внутренней структуре с защитой без использования межсетевого экрана и т. п.

• Через NAT корректно работают многие сетевые протоколы. Конструктивные реализации (общий доступ — это и есть подключение NAT) есть аппаратная реализация NAT (интегрированы межсетевые экраны).

## Функционирование[[править](https://ru.wikipedia.org/w/index.php?title=NAT&veaction=edit&vesection=1) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=NAT&action=edit&section=1)]

Преобразование адреса методом NAT может производиться почти любым [маршрутизирующим](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%86%D0%B8%D1%8F) устройством — [маршрутизатором](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), [сервером](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)) доступа, [межсетевым экраном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%8D%D0%BA%D1%80%D0%B0%D0%BD). Наиболее популярным является SNAT, суть механизма которого состоит в замене адреса источника ([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) *source*) при прохождении пакета в одну сторону и обратной замене адреса назначения ([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) *destination*) в ответном пакете. Наряду с адресами источник/назначение могут также заменяться номера [портов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_(TCP/UDP)) источника и назначения.

Принимая пакет от локального компьютера, роутер смотрит на IP-адрес назначения. Если это локальный адрес, то пакет пересылается другому локальному компьютеру. Если нет, то пакет надо переслать наружу в интернет. Но ведь обратным адресом в пакете указан локальный адрес компьютера, который из интернета будет недоступен. Поэтому роутер «на лету» транслирует (подменяет) обратный IP-адрес пакета на свой внешний (видимый из интернета) IP-адрес и меняет номер порта (чтобы различать ответные пакеты, адресованные разным локальным компьютерам). Комбинацию, нужную для обратной подстановки, роутер сохраняет у себя во временной таблице. Через некоторое время после того, как клиент и сервер закончат обмениваться пакетами, роутер сотрет у себя в таблице запись о n-ом порте за сроком давности.

Помимо source NAT (предоставления пользователям [локальной сети](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B2%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) с [внутренними адресами](https://ru.wikipedia.org/wiki/%C2%AB%D0%A1%D0%B5%D1%80%D1%8B%D0%B9%C2%BB_IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) доступа к сети [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)) часто применяется также destination NAT, когда обращения извне транслируются [межсетевым экраном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%8D%D0%BA%D1%80%D0%B0%D0%BD) на компьютер пользователя в локальной сети, имеющий [внутренний адрес](https://ru.wikipedia.org/wiki/%C2%AB%D0%A1%D0%B5%D1%80%D1%8B%D0%B9%C2%BB_IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) и потому недоступный извне сети непосредственно (без NAT).

Существует 3 базовых концепции трансляции адресов: статическая ([Static Network Address Translation](https://ru.wikipedia.org/w/index.php?title=Static_Network_Address_Translation&action=edit&redlink=1)), динамическая ([Dynamic Address Translation](https://ru.wikipedia.org/w/index.php?title=Dynamic_Address_Translation_(NAT)&action=edit&redlink=1)), [маскарадная](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D0%BA%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%BD%D0%B3)(NAPT, NAT Overload, PAT).

**Статический NAT** — Отображение незарегистрированного IP-адреса на зарегистрированный IP-адрес на основании один к одному. Особенно полезно, когда устройство должно быть доступным снаружи сети.

**Динамический NAT** — Отображает незарегистрированный IP-адрес на зарегистрированный адрес из группы зарегистрированных IP-адресов. Динамический NAT также устанавливает непосредственное отображение между незарегистрированным и зарегистрированным адресом, но отображение может меняться в зависимости от зарегистрированного адреса, доступного в пуле адресов, во время коммуникации.

**Перегруженный NAT** (NAPT, NAT Overload, PAT, маскарадинг) — форма динамического NAT, который отображает несколько незарегистрированных адресов в единственный зарегистрированный IP-адрес, используя различные порты. Известен также как [PAT](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_%D0%BF%D0%BE%D1%80%D1%82-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) (Port Address Translation). При перегрузке каждый компьютер в частной сети транслируется в тот же самый адрес, но с различным номером порта.

Механизм NAT определён в [RFC 1631](https://tools.ietf.org/html/rfc1631), [RFC 3022](https://tools.ietf.org/html/rfc3022).

### Типы NAT**[**[**править**](https://ru.wikipedia.org/w/index.php?title=NAT&veaction=edit&vesection=2)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=NAT&action=edit&section=2)**]**

Классификация NAT, часто встречающаяся в связи с [VoIP](https://ru.wikipedia.org/wiki/VoIP).[[1]](https://ru.wikipedia.org/wiki/NAT#cite_note-1) Термин «соединение» использован в значении «последовательный обмен пакетами UDP».

**Симметричный NAT** (Symmetric NAT) — Трансляция, при которой каждое соединение, инициируемое парой «внутренний адрес: внутренний порт» преобразуется в свободную уникальную случайно выбранную пару «публичный адрес: публичный порт». При этом инициация соединения из публичной сети невозможна.

**Cone NAT, Full Cone NAT** — Однозначная (взаимная) трансляция между парами «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт». Любой внешний хост может инициировать соединение с внутренним хостом (если это разрешено в правилах межсетевого экрана).

**Address-Restricted cone NAT, Restricted cone NAT** — Постоянная трансляция между парой «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт». Любое соединение, инициированное с внутреннего адреса, позволяет в дальнейшем получать ему пакеты с любого порта того публичного хоста, к которому он отправлял пакет(ы) ранее.

**Port-Restricted cone NAT** — Трансляция между парой «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт», при которой входящие пакеты проходят на внутренний хост только с одного порта публичного хоста — того, на который внутренний хост уже посылал пакет.

**The NAT Port Mapping Protocol (NAT-PMP) is a network protocol for establishingnetwork address translation (NAT) settings and port forwarding configurations automatically without user effort.**

**Трансляция порт-адрес** ([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) *Port address translation*, **PAT**) — технология трансляции сетевого адреса в зависимости от [TCP](https://ru.wikipedia.org/wiki/TCP)/[UDP](https://ru.wikipedia.org/wiki/UDP)-[порта](https://ru.wikibooks.org/w/index.php?title=%D0%9F%D0%BE%D1%80%D1%82&oldid=60877) получателя. Является частным случаем [NAT](https://ru.wikipedia.org/wiki/NAT). Также может использоваться термин DNAT (Destination NAT). То же самое что и NAT только идет привязка к определенному порту.

**Сериализация** (в программировании) — процесс перевода какой-либо [структуры данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) в последовательность [битов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82). Обратной к операции сериализации является операция *десериализации* (структуризации) — восстановление начального состояния структуры данных из битовой последовательности.

Сериализация используется для передачи [объектов](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) по сети и для сохранения их в [файлы](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB). Например, нужно создать распределённое [приложение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%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), разные части которого должны обмениваться данными со сложной структурой. В таком случае для типов данных, которые предполагается передавать, пишется код, который осуществляет сериализацию и десериализацию. Объект заполняется нужными [данными](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5), затем вызывается код сериализации, в результате получается, например, [XML](https://ru.wikipedia.org/wiki/XML)-документ. Результат сериализации передаётся принимающей стороне по, скажем, [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) или [HTTP](https://ru.wikipedia.org/wiki/HTTP). Приложение-получатель создаёт объект того же типа и вызывает код десериализации, в результате получая объект с теми же данными, что были в объекте приложения-отправителя. По такой схеме работает, например, сериализация объектов через [SOAP](https://ru.wikipedia.org/wiki/SOAP) в [Microsoft .NET](https://ru.wikipedia.org/wiki/Microsoft_.NET).

**JSON** ([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) *JavaScript Object Notation*, обычно произносится как [/ˈdʒeɪsən/](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0_%D0%9C%D0%A4%D0%90_%D0%B4%D0%BB%D1%8F_%D0%B0%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%BE%D0%B3%D0%BE_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0)[[1]](https://ru.wikipedia.org/wiki/JSON#cite_note-Pronunciation-1)) — [текстовый формат](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82) [обмена данными](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D0%B5%D0%BD_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8), основанный на [JavaScript](https://ru.wikipedia.org/wiki/JavaScript). Как и многие другие текстовые форматы, JSON легко читается людьми. Формат JSON был разработан [Дугласом Крокфордом](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D0%BA%D1%84%D0%BE%D1%80%D0%B4,_%D0%94%D1%83%D0%B3%D0%BB%D0%B0%D1%81)[[2]](https://ru.wikipedia.org/wiki/JSON#cite_note-2).

Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта [ECMA-262](https://ru.wikipedia.org/wiki/ECMA-262) [1999 года](https://ru.wikipedia.org/wiki/1999_%D0%B3%D0%BE%D0%B4)), формат считается независимым от языка и может использоваться практически с любым [языком программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Для многих языков существует готовый код для создания и обработки данных в формате JSON.

**Gson** (also known as Google Gson) is an [open source](https://en.wikipedia.org/wiki/Open_source) [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) library to [serialize](https://en.wikipedia.org/wiki/Serialize) and deserialize Java objects to (and from) [JSON](https://en.wikipedia.org/wiki/JSON).

**Jackson** is a high-performance [JSON](https://en.wikipedia.org/wiki/JSON) processor for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). Developers of it, excel the combination of fast, correct, lightweight, and ergonomic attributes of the library.[[1]](https://en.wikipedia.org/wiki/Jackson_(API)#cite_note-:jjpw-1)[[2]](https://en.wikipedia.org/wiki/Jackson_(API)#cite_note-:tjc-2)

## Implemention[[edit](https://en.wikipedia.org/w/index.php?title=Jackson_(API)&action=edit&section=1)]

Jackson provide many ways of working including simple [POJO](https://en.wikipedia.org/wiki/Plain_Old_Java_Object) converted to/from JSON for simple cases. Jackson also provides a set of annotations for mapping too.[[3]](https://en.wikipedia.org/wiki/Jackson_(API)#cite_note-:jc-3)[[1]](https://en.wikipedia.org/wiki/Jackson_(API)#cite_note-:jjpw-1)

## Usage Example[[edit](https://en.wikipedia.org/w/index.php?title=Jackson_(API)&action=edit&section=2)]

Sample code for reading and writing with POJOs may look like as follows:

**public** **class** **ReadWriteJackson** {

**public** **static** void main(String[] args) **throws** IOException {

ObjectMapper mapper = **new** ObjectMapper();

String jsonInput = "{\"id\":0,\"firstName\":\"Robin\",\"lastName\":\"Wilson\"}";

Person q = mapper.readValue(jsonInput, Person.class);

System.out.println("Read and parsed Person from JSON: " + q);

Person p = **new** Person("Roger", "Rabbit");

System.out.print("Person object " + p + " as JSON = ");

mapper.writeValue(System.out, p);

}

}

**POJO** ([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) ***Plain Old Java Object***) — «старый добрый Java-объект», простой [Java](https://ru.wikipedia.org/wiki/Java)-объект, не унаследованный от какого-то специфического объекта и не реализующий никаких служебных интерфейсов сверх тех, которые нужны для бизнес-модели.

Термин, придуманный [Мартином Фаулером](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D1%83%D0%BB%D0%B5%D1%80,_%D0%9C%D0%B0%D1%80%D1%82%D0%B8%D0%BD) c сотоварищами в пику EJB ([Enterprise JavaBeans](https://ru.wikipedia.org/wiki/Enterprise_JavaBeans)), так как отсутствие звучного термина для простых объектов приводило к тому, что молодые Java-программисты пренебрежительно к ним относились, считая что только EJB «спасут мир».

Концепция POJO появилась как результат поиска путей облегчения и упрощения методов программирования для задач, требовавших от [бизнес-объектов](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82)разносторонней функциональности, включая хранение объектов в базе данных, веб-интерфейс и т.п. Решение было найдено в виде платформ (библиотек), основанных на [интроспекции](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D1%80%D0%BE%D1%81%D0%BF%D0%B5%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)).

**JavaBeans** — классы в языке [Java](https://ru.wikipedia.org/wiki/Java), написанные по определённым правилам. Они используются для объединения нескольких [объектов](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) в один ([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) *bean* — фасоль, кофейное зерно) для удобной передачи данных.

Спецификация [Sun Microsystems](https://ru.wikipedia.org/wiki/Sun_Microsystems) определяет JavaBeans как повторно используемые программные компоненты, которыми можно управлять, используя графические конструкторы и средства [IDE](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8).

JavaBeans обеспечивают основу для многократно используемых, встраиваемых и модульных [компонентов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) [ПО](https://ru.wikipedia.org/wiki/%D0%9F%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). Компоненты JavaBeans могут принимать различные формы, но наиболее широко они применяются в элементах графического пользовательского [интерфейса](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81). Одна из целей создания JavaBeans — взаимодействие с похожими компонентными [структурами](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0). Например, [Windows-программа](https://ru.wikipedia.org/w/index.php?title=Windows-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0&action=edit&redlink=1), при наличии соответствующего [моста](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%81%D1%82_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) или [объекта-обёртки](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), может использовать компонент JavaBeans так, будто бы он является компонентом [COM](https://ru.wikipedia.org/wiki/Component_Object_Model) или [ActiveX](https://ru.wikipedia.org/wiki/ActiveX).

## Правила описания JavaBean[[править](https://ru.wikipedia.org/w/index.php?title=JavaBeans&veaction=edit&vesection=1) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=JavaBeans&action=edit&section=1)]

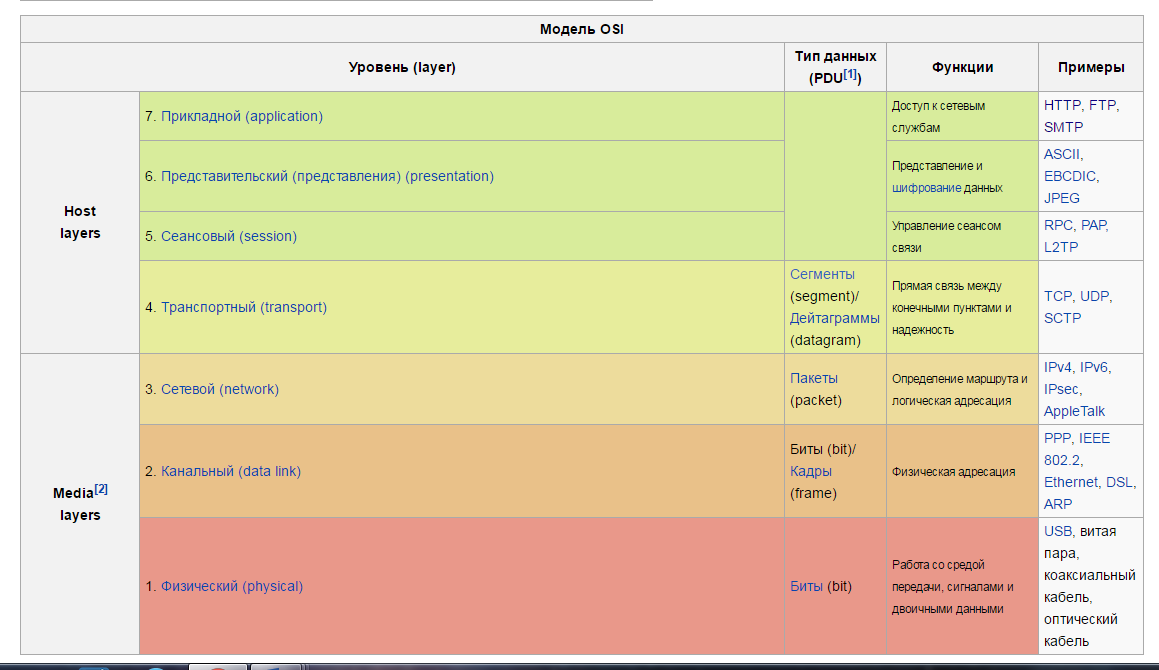
Чтобы [класс](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) мог работать как bean, он должен соответствовать определённым соглашениям об именах методов, конструкторе и поведении. Эти соглашения дают возможность создания инструментов, которые могут использовать, замещать и соединять JavaBeans.

Правила описания гласят:

* Класс должен иметь [конструктор](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) без параметров, с модификатором доступа public. Такой конструктор позволяет инструментам создать объект без дополнительных сложностей с параметрами.
* Свойства класса должны быть доступны через get, set и другие методы (так называемые [методы доступа](https://ru.wikipedia.org/wiki/Accessor)), которые должны подчиняться стандартному соглашению об именах. Это легко позволяет инструментам автоматически определять и обновлять содержание bean’ов. Многие инструменты даже имеют специализированные редакторы для различных типов свойств.
* Класс должен быть [сериализуем](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F#Java). Это даёт возможность надёжно сохранять, хранить и восстанавливать состояние bean независимым от платформы и виртуальной машины способом.
* Класс должен иметь переопределенные методы equals(), hashCode() и toString().

Так как требования в основном изложены в виде соглашения, а не [интерфейса](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), некоторые разработчики рассматривают JavaBeans, как [Plain Old Java Objects](https://ru.wikipedia.org/wiki/POJO), которые следуют определённым правилам именования.

**Сетевая модель OSI** ([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) *open systems interconnection basic reference model* — **базовая**[**эталонная модель**](https://ru.wikipedia.org/wiki/%D0%AD%D1%82%D0%B0%D0%BB%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C)**взаимодействия открытых систем,** сокр. **ЭМВОС**; 1978 год) — [сетевая модель](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C) стека [сетевых протоколов](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) OSI/ISO (ГОСТ Р ИСО/МЭК 7498-1-99).



* тип передающей среды (медный кабель, оптоволокно, радиоэфир и др.),
* тип модуляции сигнала,
* сигнальные уровни логических дискретных состояний (нуля и единицы).

### Прикладной уровень**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=3)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=3)**]**

*Основная статья:*[***Прикладной уровень***](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Прикладной уровень (уровень приложений; [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) *application layer*) — верхний уровень модели, обеспечивающий взаимодействие пользовательских приложений с сетью:

* позволяет приложениям использовать сетевые службы:
  + удалённый доступ к файлам и базам данных,
  + пересылка электронной почты;
* отвечает за передачу служебной информации;
* предоставляет приложениям информацию об ошибках;
* формирует запросы к уровню представления.

Протоколы прикладного уровня: [RDP](https://ru.wikipedia.org/wiki/Remote_Desktop_Protocol), [HTTP](https://ru.wikipedia.org/wiki/HTTP), [SMTP](https://ru.wikipedia.org/wiki/SMTP), [SNMP](https://ru.wikipedia.org/wiki/SNMP), [POP3](https://ru.wikipedia.org/wiki/Post_Office_Protocol), [FTP](https://ru.wikipedia.org/wiki/FTP), [XMPP](https://ru.wikipedia.org/wiki/XMPP), [OSCAR](https://ru.wikipedia.org/wiki/OSCAR), [Modbus](https://ru.wikipedia.org/wiki/Modbus), [SIP](https://ru.wikipedia.org/wiki/SIP), [TELNET](https://ru.wikipedia.org/wiki/TELNET) и другие.

### Уровень представления**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=4)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=4)**]**

*Основная статья:*[***Представительский уровень***](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Представительский уровень (уровень представления; [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) *presentation layer*) обеспечивает преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с прикладного уровня, на уровне представления преобразуются в формат для передачи по сети, а полученные из сети данные преобразуются в формат приложений. На этом уровне может осуществляться сжатие/распаковка или шифрование/дешифрование, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

Уровень представлений обычно представляет собой промежуточный протокол для преобразования информации из соседних уровней. Это позволяет осуществлять обмен между приложениями на разнородных компьютерных системах прозрачным для приложений образом. Уровень представлений обеспечивает форматирование и преобразование кода. Форматирование кода используется для того, чтобы гарантировать приложению поступление информации для обработки, которая имела бы для него смысл. При необходимости этот уровень может выполнять перевод из одного формата данных в другой.

Уровень представлений имеет дело не только с форматами и представлением данных, он также занимается структурами данных, которые используются программами. Таким образом, уровень 6 обеспечивает организацию данных при их пересылке.

Чтобы понять, как это работает, представим, что имеются две системы. Одна использует для представления данных расширенный двоичный код обмена информацией[EBCDIC](https://ru.wikipedia.org/wiki/EBCDIC), например, это может быть [мейнфрейм](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B9%D0%BD%D1%84%D1%80%D0%B5%D0%B9%D0%BC) компании [IBM](https://ru.wikipedia.org/wiki/IBM), а другая — американский стандартный код обмена информацией [ASCII](https://ru.wikipedia.org/wiki/ASCII) (его использует большинство других производителей компьютеров). Если этим двум системам необходимо обменяться информацией, то нужен уровень представлений, который выполнит преобразование и осуществит перевод между двумя различными форматами.

Другой функцией, выполняемой на уровне представлений, является шифрование данных, которое применяется в тех случаях, когда необходимо защитить передаваемую информацию от доступа несанкционированными получателями. Чтобы решить эту задачу, процессы и коды, находящиеся на уровне представлений, должны выполнить преобразование данных. На этом уровне существуют и другие подпрограммы, которые сжимают тексты и преобразовывают графические изображения в битовые потоки, так, что они могут передаваться по сети.

Стандарты уровня представлений также определяют способы представления графических изображений. Для этих целей может использоваться формат [PICT](https://ru.wikipedia.org/w/index.php?title=PICT&action=edit&redlink=1) — формат изображений, применяемый для передачи графики QuickDraw между программами.

Другим форматом представлений является тэгированный формат файлов изображений [TIFF](https://ru.wikipedia.org/wiki/TIFF), который обычно используется для растровых изображений с высоким[разрешением](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B5%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%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0)). Следующим стандартом уровня представлений, который может использоваться для графических изображений, является стандарт, разработанный Объединенной экспертной группой по фотографии (Joint Photographic Expert Group); в повседневном пользовании этот стандарт называют просто [JPEG](https://ru.wikipedia.org/wiki/JPEG).

Существует другая группа стандартов уровня представлений, которая определяет представление звука и кинофрагментов. Сюда входят интерфейс электронных музыкальных инструментов ([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) *Musical Instrument Digital Interface*, [MIDI](https://ru.wikipedia.org/wiki/MIDI)) для цифрового представления музыки, разработанный Экспертной группой по кинематографии стандарт [MPEG](https://ru.wikipedia.org/wiki/MPEG), используемый для сжатия и кодирования видеороликов на компакт-дисках, хранения в оцифрованном виде и передачи со скоростями до 1,5 Мбит/с, и [QuickTime](https://ru.wikipedia.org/wiki/QuickTime) — стандарт, описывающий звуковые и видео элементы для программ, выполняемых на компьютерах Macintosh и PowerPC.

Протоколы уровня представления: AFP — [Apple Filing Protocol](https://ru.wikipedia.org/wiki/Apple_Filing_Protocol), ICA — [Independent Computing Architecture](https://ru.wikipedia.org/wiki/Independent_Computing_Architecture), LPP — Lightweight Presentation Protocol, NCP — [NetWare Core Protocol](https://ru.wikipedia.org/wiki/NetWare_Core_Protocol), NDR — [Network Data Representation](https://ru.wikipedia.org/w/index.php?title=Network_Data_Representation&action=edit&redlink=1), XDR — [eXternal Data Representation](https://ru.wikipedia.org/wiki/External_Data_Representation" \o "External Data Representation), X.25 PAD — [Packet Assembler/Disassembler Protocol](https://ru.wikipedia.org/wiki/X.25).

### Сеансовый уровень**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=5)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=5)**]**

*Основная статья:*[***Сеансовый уровень***](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)

Сеансовый 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) *session layer*) модели обеспечивает поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений.

Протоколы сеансового уровня: ADSP ([AppleTalk Data Stream Protocol](https://ru.wikipedia.org/wiki/AppleTalk)), ASP ([AppleTalk Session Protocol](https://ru.wikipedia.org/wiki/AppleTalk)), H.245 ([Call Control Protocol for Multimedia Communication](https://ru.wikipedia.org/wiki/H.245)), ISO-SP (OSI Session Layer Protocol (X.225, ISO 8327)), iSNS ([Internet Storage Name Service](https://ru.wikipedia.org/wiki/Internet_Storage_Name_Service)), L2F ([Layer 2 Forwarding Protocol](https://ru.wikipedia.org/wiki/Layer_2_Forwarding_Protocol)), L2TP ([Layer 2 Tunneling Protocol](https://ru.wikipedia.org/wiki/L2TP)), NetBIOS ([Network Basic Input Output System](https://ru.wikipedia.org/wiki/NetBIOS)), PAP ([Password Authentication Protocol](https://ru.wikipedia.org/wiki/Password_Authentication_Protocol)), PPTP ([Point-to-Point Tunneling Protocol](https://ru.wikipedia.org/wiki/PPTP)), RPC ([Remote Procedure Call Protocol](https://ru.wikipedia.org/wiki/Remote_procedure_call)), RTCP ([Real-time Transport Control Protocol](https://ru.wikipedia.org/wiki/RTCP)), SMPP ([Short Message Peer-to-Peer](https://ru.wikipedia.org/wiki/SMPP)), SCP ([Session Control Protocol](https://ru.wikipedia.org/w/index.php?title=Session_Control_Protocol&action=edit&redlink=1)), ZIP ([Zone Information Protocol](https://ru.wikipedia.org/wiki/Zone_Information_Protocol)), SDP ([Sockets Direct Protocol](https://ru.wikipedia.org/wiki/Sockets_Direct_Protocol))..

### Транспортный уровень**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=6)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=6)**]**

*Основная статья:*[***Транспортный уровень***](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_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Транспортный 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) *transport layer*) модели предназначен для обеспечения надёжной передачи данных от отправителя к получателю. При этом уровень надёжности может варьироваться в широких пределах. Существует множество классов протоколов транспортного уровня, начиная от протоколов, предоставляющих только основные транспортные функции (например, функции передачи данных без подтверждения приема), и заканчивая протоколами, которые гарантируют доставку в пункт назначения нескольких пакетов данных в надлежащей последовательности, мультиплексируют несколько потоков данных, обеспечивают механизм управления потоками данных и гарантируют достоверность принятых данных. Например, [UDP](https://ru.wikipedia.org/wiki/UDP) ограничивается контролем целостности данных в рамках одной датаграммы и не исключает возможности потери пакета целиком или дублирования пакетов, нарушение порядка получения пакетов данных; [TCP](https://ru.wikipedia.org/wiki/TCP) обеспечивает надёжную непрерывную передачу данных, исключающую потерю данных или нарушение порядка их поступления или дублирования, может перераспределять данные, разбивая большие порции данных на фрагменты и наоборот, склеивая фрагменты в один пакет.

Протоколы транспортного уровня: ATP ([AppleTalk Transaction Protocol](https://ru.wikipedia.org/wiki/AppleTalk)), CUDP ([Cyclic UDP](https://ru.wikipedia.org/w/index.php?title=Cyclic_UDP&action=edit&redlink=1)), DCCP ([Datagram Congestion Control Protocol](https://ru.wikipedia.org/wiki/Datagram_Congestion_Control_Protocol)), FCP ([Fiber Channel Protocol](https://ru.wikipedia.org/wiki/Fibre_Channel)), IL ([IL Protocol](https://ru.wikipedia.org/wiki/IL_Protocol)), NBF ([NetBIOS Frames protocol](https://ru.wikipedia.org/w/index.php?title=NetBIOS_Frames_protocol&action=edit&redlink=1)), NCP ([NetWare Core Protocol](https://ru.wikipedia.org/wiki/NetWare_Core_Protocol)), SCTP ([Stream Control Transmission Protocol](https://ru.wikipedia.org/wiki/Stream_Control_Transmission_Protocol)), SPX ([Sequenced Packet Exchange](https://ru.wikipedia.org/wiki/IPX/SPX)), SST ([Structured Stream Transport](https://ru.wikipedia.org/w/index.php?title=Structured_Stream_Transport&action=edit&redlink=1)), TCP ([Transmission Control Protocol](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol)), UDP ([User Datagram Protocol](https://ru.wikipedia.org/wiki/User_Datagram_Protocol)).

### Сетевой уровень**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=7)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=7)**]**

*Основная статья:*[***Сетевой уровень***](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Сетевой 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) *network layer*) модели предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и «заторов» в сети.

Протоколы сетевого уровня маршрутизируют данные от источника к получателю. Работающие на этом уровне устройства ([маршрутизаторы](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)) условно называют устройствами третьего уровня (по номеру уровня в модели OSI).

Протоколы сетевого уровня: IP/IPv4/IPv6 ([Internet Protocol](https://ru.wikipedia.org/wiki/Internet_Protocol)), IPX ([Internetwork Packet Exchange](https://ru.wikipedia.org/wiki/Internetwork_Packet_Exchange), протокол межсетевого обмена), X.25 (частично этот протокол реализован на уровне 2), CLNP (сетевой протокол без организации соединений), IPsec ([Internet Protocol Security](https://ru.wikipedia.org/wiki/Internet_Protocol_Security)). Протоколы маршрутизации - RIP ([Routing Information Protocol](https://ru.wikipedia.org/wiki/Routing_Information_Protocol)), OSPF ([Open Shortest Path First](https://ru.wikipedia.org/wiki/Open_Shortest_Path_First)).

### Канальный уровень**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=8)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=8)**]**

*Основная статья:*[***Канальный уровень***](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Канальный 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) *data link layer*) 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](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B4%D1%80_(%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8)), проверяет их на целостность и, если нужно, исправляет ошибки (формирует повторный запрос поврежденного кадра) и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием.

Спецификация [IEEE 802](https://ru.wikipedia.org/wiki/IEEE_802) разделяет этот уровень на два подуровня: [MAC](https://ru.wikipedia.org/wiki/Media_Access_Control) ([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) *media access control*) регулирует доступ к разделяемой физической среде, [LLC](https://ru.wikipedia.org/wiki/Logical_link_control)([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) *logical link control*) обеспечивает обслуживание сетевого уровня.

На этом уровне работают [коммутаторы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D1%82%D0%B0%D1%82%D0%BE%D1%80), [мосты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BC%D0%BE%D1%81%D1%82) и другие устройства.Эти устройства используют адресацию второго уровня (по номеру уровня в модели OSI).

Протоколы канального уровня: [ARCnet](https://ru.wikipedia.org/wiki/ARCnet" \o "ARCnet), [ATM](https://ru.wikipedia.org/wiki/Asynchronous_Transfer_Mode), [Controller Area Network](https://ru.wikipedia.org/wiki/Controller_Area_Network) (CAN), [Econet](https://ru.wikipedia.org/w/index.php?title=Econet&action=edit&redlink=1" \o "Econet (страница отсутствует)), [IEEE 802.3](https://ru.wikipedia.org/wiki/IEEE_802.3) ([Ethernet](https://ru.wikipedia.org/wiki/Ethernet)), [Ethernet Automatic Protection Switching](https://ru.wikipedia.org/wiki/Ethernet_Automatic_Protection_Switching) (EAPS), [Fiber Distributed Data Interface](https://ru.wikipedia.org/wiki/Fiber_Distributed_Data_Interface) (FDDI), [Frame Relay](https://ru.wikipedia.org/wiki/Frame_Relay), [High-Level Data Link Control](https://ru.wikipedia.org/wiki/High-Level_Data_Link_Control) (HDLC), [IEEE 802.2](https://ru.wikipedia.org/w/index.php?title=IEEE_802.2&action=edit&redlink=1) (предоставляет функции LLC для подуровня IEEE 802 MAC), [Link Access Procedures, D channel](https://ru.wikipedia.org/w/index.php?title=Link_Access_Procedures,_D_channel&action=edit&redlink=1) (LAPD), [IEEE 802.11](https://ru.wikipedia.org/wiki/IEEE_802.11) [wireless LAN](https://ru.wikipedia.org/wiki/Wireless_LAN), [LocalTalk](https://ru.wikipedia.org/w/index.php?title=LocalTalk&action=edit&redlink=1" \o "LocalTalk (страница отсутствует)), [Multiprotocol Label Switching](https://ru.wikipedia.org/wiki/Multiprotocol_Label_Switching) (MPLS), [Point-to-Point Protocol](https://ru.wikipedia.org/wiki/Point-to-Point_Protocol) (PPP), [Point-to-Point Protocol over Ethernet](https://ru.wikipedia.org/wiki/PPPoE) (PPPoE),[Serial Line Internet Protocol](https://ru.wikipedia.org/wiki/SLIP) (SLIP, устарел), [StarLan](https://ru.wikipedia.org/wiki/StarLan" \o "StarLan), [Token ring](https://ru.wikipedia.org/wiki/Token_ring), [Unidirectional Link Detection](https://ru.wikipedia.org/w/index.php?title=UDLD&action=edit&redlink=1) (UDLD), [x.25](https://ru.wikipedia.org/wiki/X.25), [ARP](https://ru.wikipedia.org/wiki/ARP).

В программировании этот уровень представляет [драйвер](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B0%D0%B9%D0%B2%D0%B5%D1%80) сетевой платы, в [операционных системах](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) имеется программный интерфейс взаимодействия канального и сетевого уровней между собой. Это не новый уровень, а просто реализация модели для конкретной ОС. Примеры таких интерфейсов: [ODI](https://ru.wikipedia.org/wiki/ODI), [NDIS](https://ru.wikipedia.org/wiki/NDIS), [UDI](https://ru.wikipedia.org/w/index.php?title=UDI&action=edit&redlink=1).

### Физический уровень**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=9)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=9)**]**

*Основная статья:*[***Физический слой***](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%81%D0%BB%D0%BE%D0%B9)

Физический 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) *physical layer*) — нижний уровень модели, который определяет метод передачи данных, представленных в двоичном виде, от одного устройства (компьютера) к другому. Составлением таких методов занимаются разные организации, в том числе: [Институт инженеров по электротехнике и электронике](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%BE%D0%B2_%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B8_%D0%B8_%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%B8%D0%BA%D0%B8),[Альянс электронной промышленности](https://ru.wikipedia.org/wiki/Electronic_Industries_Alliance), [Европейский институт телекоммуникационных стандартов](https://ru.wikipedia.org/wiki/%D0%95%D0%B2%D1%80%D0%BE%D0%BF%D0%B5%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82_%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D1%85_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BE%D0%B2) и другие. Осуществляют передачу электрических или оптических сигналов в кабель или в радиоэфир и, соответственно, их приём и преобразование в биты данных в соответствии с [методами кодирования цифровых сигналов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

На этом уровне также работают [концентраторы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%BD%D1%86%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80), [повторители](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D1%82%D0%B5%D0%BB%D1%8C_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B5_%D0%BE%D0%B1%D0%BE%D1%80%D1%83%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) сигнала и [медиаконвертеры](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D0%B0%D0%BA%D0%BE%D0%BD%D0%B2%D0%B5%D1%80%D1%82%D0%B5%D1%80).

Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом. К физическому уровню относятся физические, электрические и механические интерфейсы между двумя системами. Физический уровень определяет такие виды сред передачи данных как [оптоволокно](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B2%D0%BE%D0%BB%D0%BE%D0%BA%D0%BD%D0%BE), [витая пара](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%82%D0%B0%D1%8F_%D0%BF%D0%B0%D1%80%D0%B0), [коаксиальный кабель](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B0%D0%BA%D1%81%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%B0%D0%B1%D0%B5%D0%BB%D1%8C), спутниковый канал передач данных и т. п. Стандартными типами сетевых интерфейсов, относящимися к физическому уровню, являются: [V.35](https://ru.wikipedia.org/wiki/V.35), [RS-232](https://ru.wikipedia.org/wiki/RS-232), [RS-485](https://ru.wikipedia.org/wiki/RS-485), [RJ-11](https://ru.wikipedia.org/wiki/RJ-45), [RJ-45](https://ru.wikipedia.org/wiki/RJ-45), разъемы [AUI](https://ru.wikipedia.org/wiki/AUI) и [BNC](https://ru.wikipedia.org/wiki/BNC-%D0%BA%D0%BE%D0%BD%D0%BD%D0%B5%D0%BA%D1%82%D0%BE%D1%80).

Протоколы физического уровня: [IEEE 802.15 (Bluetooth)](https://ru.wikipedia.org/wiki/Bluetooth), [IRDA](https://ru.wikipedia.org/wiki/Infrared_Data_Association), [EIA](https://ru.wikipedia.org/wiki/Electronic_Industries_Alliance) [RS-232](https://ru.wikipedia.org/wiki/RS-232), [EIA-422](https://ru.wikipedia.org/wiki/EIA-422), [EIA-423](https://ru.wikipedia.org/wiki/RS-423), [RS-449](https://ru.wikipedia.org/wiki/RS-449), [RS-485](https://ru.wikipedia.org/wiki/RS-485), [DSL](https://ru.wikipedia.org/wiki/Digital_subscriber_line), [ISDN](https://ru.wikipedia.org/wiki/Integrated_Services_Digital_Network), [SONET/SDH](https://ru.wikipedia.org/wiki/Synchronous_optical_network), [802.11](https://ru.wikipedia.org/wiki/802.11) [Wi-Fi](https://ru.wikipedia.org/wiki/Wi-Fi), [Etherloop](https://ru.wikipedia.org/wiki/Etherloop" \o "Etherloop), [GSM](https://ru.wikipedia.org/wiki/GSM) [Um radio interface](https://ru.wikipedia.org/w/index.php?title=Um_Interface&action=edit&redlink=1), [ITU](https://ru.wikipedia.org/wiki/International_Telecommunication_Union) и [ITU-T](https://ru.wikipedia.org/wiki/ITU-T), [TransferJet](https://ru.wikipedia.org/w/index.php?title=TransferJet&action=edit&redlink=1" \o "TransferJet (страница отсутствует)), [ARINC 818](https://ru.wikipedia.org/w/index.php?title=ARINC_818&action=edit&redlink=1), [G.hn](https://ru.wikipedia.org/w/index.php?title=G.hn&action=edit&redlink=1)/[G.9960](https://ru.wikipedia.org/w/index.php?title=G.9960&action=edit&redlink=1).

## Соответствие модели OSI и других моделей сетевого взаимодействия[[править](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=10) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=10)]

Поскольку наиболее востребованными и практически используемыми стали протоколы (например TCP/IP), разработанные с использованием других моделей сетевого взаимодействия, далее необходимо описать возможное включение отдельных протоколов других моделей в различные уровни модели OSI.

### Семейство TCP/IP**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=11)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=11)**]**

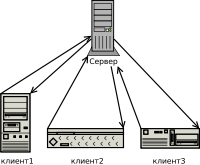
Семейство [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP) имеет три транспортных протокола: TCP, полностью соответствующий OSI, обеспечивающий проверку получения данных; [UDP](https://ru.wikipedia.org/wiki/UDP), отвечающий транспортному уровню только наличием порта, обеспечивающий обмен [датаграммами](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D1%82%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) между приложениями, не гарантирующий получения данных; и [SCTP](https://ru.wikipedia.org/wiki/SCTP), разработанный для устранения некоторых недостатков TCP, в который добавлены некоторые новшества. В семействе TCP/IP есть ещё около двухсот протоколов, самым известным из которых является служебный протокол [ICMP](https://ru.wikipedia.org/wiki/ICMP), используемый для внутренних нужд обеспечения работы; остальные также не являются транспортными протоколами.

### Семейство IPX/SPX**[**[**править**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&vesection=12)**|**[**править вики-текст**](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=12)**]**

В семействе [IPX/SPX](https://ru.wikipedia.org/wiki/IPX/SPX) порты появляются в протоколе сетевого уровня IPX, обеспечивая обмен [датаграммами](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D1%82%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) между приложениями (операционная система резервирует часть сокетов для себя). Протокол SPX, в свою очередь, дополняет IPX всеми остальными возможностями транспортного уровня в полном соответствии с OSI.

В качестве адреса хоста ICX использует идентификатор, образованный из четырёхбайтного номера сети (назначаемого [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)) и MAC-адреса сетевого адаптера.

**Клиент-сервер** ([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) *Client-server*) — 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](https://ru.wikipedia.org/wiki/%D0%9F%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). Обычно они взаимодействуют через компьютерную сеть посредством [сетевых протоколов](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)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](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2) посредством [HTTP](https://ru.wikipedia.org/wiki/HTTP),[FTP](https://ru.wikipedia.org/wiki/FTP), [BitTorrent](https://ru.wikipedia.org/wiki/BitTorrent_(%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)" \o "BitTorrent (протокол)), [потоковое мультимедиа](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D0%BE%D0%B5_%D0%BC%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0) или работа с [базами данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)) или сервисных функций (например, работа с [электронной почтой](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), общение посредством [систем мгновенного обмена сообщениями](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BC%D0%B3%D0%BD%D0%BE%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%D0%B0_%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F%D0%BC%D0%B8), просмотр [web-страниц](https://ru.wikipedia.org/wiki/Web-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) во [всемирной паутине](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0)).

[](https://commons.wikimedia.org/wiki/File:Two-tier_architecture.ru.svg?uselang=ru)

Пример двухуровневой архитектуры

## Преимущества[[править](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80&veaction=edit&vesection=1) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80&action=edit&section=1)]

* 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. На сервере проще организовать контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

## Многоуровневая архитектура клиент-сервер[[править](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80&veaction=edit&vesection=3) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80&action=edit&section=3)]

**Многоуровневая архитектура клиент-сервер** — разновидность архитектуры клиент-сервер, в которой функция обработки данных вынесена на один или несколько отдельных серверов. Это позволяет разделить функции хранения, обработки и представления данных для более эффективного использования возможностей серверов и клиентов.

Частные случаи многоуровневой архитектуры:

* [Трёхуровневая архитектура](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0)
* [Клиент](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)) (*слой клиента*) — это интерфейсный (обычно [графический](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F)) компонент комплекса, предоставляемый конечному пользователю. Этот уровень не должен иметь прямых связей с базой данных (по требованиям безопасности и масштабируемости), быть нагруженным основной [бизнес-логикой](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0) (по требованиям[масштабируемости](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D0%BE%D1%81%D1%82%D1%8C)) и хранить состояние приложения (по требованиям надёжности). На этот уровень обычно выносится только простейшая бизнес-логика: интерфейс [авторизации](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), алгоритмы шифрования, проверка вводимых значений на допустимость и соответствие формату, несложные операции с данными (сортировка, группировка, подсчёт значений), уже загруженными на терминал.
* [Сервер приложений](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%B9) (*средний слой*, *связующий слой*) располагается на втором уровне, на нём сосредоточена бо́льшая часть бизнес-логики. Вне его остаются только фрагменты, экспортируемые на клиента (терминалы), а также элементы логики, погруженные в базу данных (хранимые процедуры и триггеры). Реализация данного компонента обеспечивается [связующим программным обеспечением](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%8F%D0%B7%D1%83%D1%8E%D1%89%D0%B5%D0%B5_%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). Серверы приложений проектируются таким образом, чтобы добавление к ним дополнительных экземпляров обеспечивало [горизонтальное масштабирование](https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D1%80%D0%B8%D0%B7%D0%BE%D0%BD%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BC%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) производительности программного комплекса и не требовало внесения изменений в программный код приложения.
* Сервер баз данных (*слой данных*) обеспечивает хранение данных и выносится на отдельный уровень, реализуется, как правило, средствами [систем управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), подключение к этому компоненту обеспечивается только с уровня сервера приложений.
* В простейших конфигурациях все компоненты или часть из них могут быть совмещены на одном вычислительном узле. В продуктивных конфигурациях как правило используется выделенный вычислительный узел для сервера баз данных или [кластер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80_(%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BE%D0%B2)) серверов баз данных, для серверов приложений — выделенная группа вычислительных узлов, к которым непосредственно подключаются клиенты (терминалы).

