**Transmission Control Protocol** (TCP, 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).

В [стеке протоколов TCP/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) выполняет функции [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 надёжную передачу потока байтов от одного процесса к другому. TCP реализует управление потоком, управление перегрузкой, рукопожатие, надёжную передачу.



**Содержание**

* [1Заголовок сегмента TCP](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%97%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D0%BA_%D1%81%D0%B5%D0%B3%D0%BC%D0%B5%D0%BD%D1%82%D0%B0_TCP)
	+ [1.1Порт источника, Порт назначения](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9F%D0%BE%D1%80%D1%82_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B0,_%D0%9F%D0%BE%D1%80%D1%82_%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)
	+ [1.2Порядковый номер](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9F%D0%BE%D1%80%D1%8F%D0%B4%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BD%D0%BE%D0%BC%D0%B5%D1%80)
	+ [1.3Номер подтверждения](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9D%D0%BE%D0%BC%D0%B5%D1%80_%D0%BF%D0%BE%D0%B4%D1%82%D0%B2%D0%B5%D1%80%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F)
	+ [1.4Длина заголовка (смещение данных)](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%94%D0%BB%D0%B8%D0%BD%D0%B0_%D0%B7%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BA%D0%B0_(%D1%81%D0%BC%D0%B5%D1%89%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85))
	+ [1.5Зарезервировано](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%97%D0%B0%D1%80%D0%B5%D0%B7%D0%B5%D1%80%D0%B2%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BE)
	+ [1.6Флаги (управляющие биты)](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A4%D0%BB%D0%B0%D0%B3%D0%B8_(%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D1%8E%D1%89%D0%B8%D0%B5_%D0%B1%D0%B8%D1%82%D1%8B))
	+ [1.7Размер окна](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A0%D0%B0%D0%B7%D0%BC%D0%B5%D1%80_%D0%BE%D0%BA%D0%BD%D0%B0)
	+ [1.8Контрольная сумма](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%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)
	+ [1.9Указатель важности](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A3%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C_%D0%B2%D0%B0%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D0%B8)
	+ [1.10Опции](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9E%D0%BF%D1%86%D0%B8%D0%B8)
* [2Механизм действия протокола](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9C%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC_%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%B0)
	+ [2.1Состояния сеанса TCP](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A1%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_%D1%81%D0%B5%D0%B0%D0%BD%D1%81%D0%B0_TCP)
	+ [2.2Установка соединения](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0_%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F)
	+ [2.3Передача данных](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%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)
	+ [2.4Завершение соединения](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%97%D0%B0%D0%B2%D0%B5%D1%80%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F)
* [3Известные проблемы](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%98%D0%B7%D0%B2%D0%B5%D1%81%D1%82%D0%BD%D1%8B%D0%B5_%D0%BF%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC%D1%8B)
	+ [3.1Максимальный размер сегмента](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9C%D0%B0%D0%BA%D1%81%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%80_%D1%81%D0%B5%D0%B3%D0%BC%D0%B5%D0%BD%D1%82%D0%B0)
	+ [3.2Обнаружение ошибок при передаче данных](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9E%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BE%D1%88%D0%B8%D0%B1%D0%BE%D0%BA_%D0%BF%D1%80%D0%B8_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)
	+ [3.3Атаки на протокол](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%90%D1%82%D0%B0%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)
* [4Реализация](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A0%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)
	+ [4.1Псевдозаголовок](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9F%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D0%B7%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D0%BA)
	+ [4.2Освобождение от расчёта контрольной суммы](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9E%D1%81%D0%B2%D0%BE%D0%B1%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BE%D1%82_%D1%80%D0%B0%D1%81%D1%87%D1%91%D1%82%D0%B0_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C%D0%BD%D0%BE%D0%B9_%D1%81%D1%83%D0%BC%D0%BC%D1%8B)
* [5См. также](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A1%D0%BC._%D1%82%D0%B0%D0%BA%D0%B6%D0%B5)
* [6Ссылки](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8)
* [7Литература](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol#%D0%9B%D0%B8%D1%82%D0%B5%D1%80%D0%B0%D1%82%D1%83%D1%80%D0%B0)

Заголовок сегмента TCP[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=1) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=1)]

|  |
| --- |
| **Структура заголовка** |
| [**Бит**](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82) | **0 — 3** | **4 — 9** | **10 — 15** | **16 — 31** |
| **0** | Порт источника, **Source Port**  | Порт назначения, **Destination Port** |
| **32** | Порядковый номер, **Sequence Number (SN)** |
| **64** | Номер подтверждения, **Acknowledgment Number (ACK SN)** |
| **96** | Длина заголовка | Зарезервировано | Флаги | Размер Окна |
| **128** | Контрольная сумма | Указатель важности |
| **160** | Опции (необязательное, но используется практически всегда) |
| **160/192+** | Данные |

**Порт источника, Порт назначения**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=2) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=2)]

Эти 16-битные поля содержат номера [портов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_%28TCP/UDP%29) — числа, которые определяются по [специальному списку](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BF%D0%BE%D1%80%D1%82%D0%BE%D0%B2_TCP_%D0%B8_UDP).

**Порт источника** идентифицирует приложение клиента, с которого отправлены пакеты. Ответные данные передаются клиенту на основании этого номера.

**Порт назначения** идентифицирует порт, на который отправлен пакет.

**Порядковый номер**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=3) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=3)]

**Sequence number** (32 бита) — измеряется в байтах, и каждый переданный байт полезных данных (payload) увеличивает это значение на 1.

Если установлен флаг SYN (идёт установление сессии), то поле содержит изначальный порядковый номер — ISN (Initial Sequence Number). В целях безопасности это значение генерируется случайным образом и может быть равно от 0 до 232-1 (4294967295). Первый байт полезных данных в устанавливающейся сессии будет иметь номер ISN+1.

В противном случае, если SYN не установлен, первый байт данных, передаваемый в данном пакете, имеет этот порядковый номер.

Поскольку поток TCP в общем случае может быть длиннее, чем число различных состояний этого поля, то все операции с порядковым номером должны выполняться по модулю 232. Это накладывает практическое ограничение на использование TCP. Если скорость передачи коммуникационной системы такова, чтобы в течение MSL (максимального времени жизни сегмента) произошло переполнение порядкового номера, то в сети может появиться два сегмента с одинаковым номером, относящихся к разным частям потока, и приёмник получит некорректные данные.

**Номер подтверждения**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=4) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=4)]

**Acknowledgment Number (ACK SN)** (32 бита) — если установлен флаг ACK, то это поле содержит порядковый номер октета, который отправитель данного сегмента желает получить. Это означает, что все предыдущие октеты (с номерами от ISN+1 до ACK-1 включительно) были успешно получены.

Каждая сторона подсчитывает свой Sequence number для переданных данных и отдельно Acknowledgement number для полученных данных. Соответственно Sequence number каждой из сторон соответствует Acknowledgement number другой стороны.

**Длина заголовка (смещение данных)**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=5) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=5)]

Длина заголовка (Data offset) занимает 4 бита и указывает значение длины заголовка, измеренное в 32-битовых словах. Минимальный размер составляет 20 байт (пять 32-битовых слов), а максимальный — 60 байт (пятнадцать 32-битовых слов). Длина заголовка определяет смещение полезных данных относительно начала сегмента. Например, Data offset равное 1111 говорит о том, что заголовок занимает пятнадцать 32-битных слова (15 строк\*32 бита в каждой строке/8 бит = 60 байт).

**Зарезервировано**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=6) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=6)]

Зарезервировано (6 бит) для будущего использования и должно устанавливаться в ноль. Из них два (5-й и 6-й) уже определены:

* **CWR** (Congestion Window Reduced) — Поле «Окно перегрузки уменьшено» — флаг установлен отправителем, чтобы указать, что получен пакет с установленным флагом ECE ([RFC 3168](https://tools.ietf.org/html/rfc3168))
* **ECE** (ECN-Echo) — Поле «Эхо ECN» — указывает, что данный узел способен на ECN (явное уведомление перегрузки) и для указания отправителю о перегрузках в сети ([RFC 3168](https://tools.ietf.org/html/rfc3168))

**Флаги (управляющие биты)**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=7) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=7)]

Это поле содержит 6 битовых флагов:

* **URG** — поле *«Указатель важности»* задействовано ([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) *Urgent pointer field is significant*)
* **ACK** — поле *«Номер подтверждения»* задействовано ([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) *Acknowledgement field is significant*)
* **PSH** — ([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) *Push function*) инструктирует получателя протолкнуть данные, накопившиеся в приёмном буфере, в приложение пользователя
* **RST** — оборвать соединения, сбросить буфер (очистка буфера) ([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) *Reset the connection*)
* **SYN** — синхронизация номеров последовательности ([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) *Synchronize sequence numbers*)
* **FIN** ([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) *final*, бит) — флаг, будучи установлен, указывает на завершение соединения ([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) *FIN bit used for connection termination*).

**Размер окна**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=8) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=8)]

Window Size определяет количество байт данных (payload), после передачи которых отправитель ожидает подтверждения от получателя, что данные получены. Иначе говоря, получатель пакета располагает для приёма данных буфером длиной "размер окна" байт.

По умолчанию размер окна измеряется в байтах, поэтому ограничен 216 (65535) байтами. Однако благодаря TCP опции Window scale option этот размер может быть увеличен до 1 Гбайта. Чтобы задействовать эту опцию, обе стороны должны согласовать это в своих SYN сегментах.

**Контрольная сумма**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=9) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=9)]

Поле контрольной суммы — это 16-битное дополнение к сумме всех 16-битных слов заголовка (включая псевдозаголовок) и данных. Если сегмент, по которому вычисляется контрольная сумма, имеет длину не кратную 16-битам, то длина сегмента увеличивается до кратной 16-ти, за счёт дополнения к нему справа нулевых битов заполнения. Биты заполнения (0) не передаются в сообщении и служат только для расчёта контрольной суммы. При расчёте контрольной суммы значение самого поля контрольной суммы принимается равным 0.

**Указатель важности**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=10) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=10)]

16-битовое значение положительного смещения от порядкового номера в данном сегменте. Это поле указывает порядковый номер октета, которым заканчиваются важные (urgent) данные. Поле принимается во внимание только для пакетов с установленным флагом URG. Используется для [внеполосных данных](https://ru.wikipedia.org/wiki/%D0%92%D0%BD%D0%B5%D0%BF%D0%BE%D0%BB%D0%BE%D1%81%D0%BD%D1%8B%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5).

**Опции**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=11) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=11)]

Могут применяться в некоторых случаях для расширения протокола. Иногда используются для тестирования. На данный момент в опции практически всегда включают 2 байта [NOP](https://ru.wikipedia.org/wiki/NOP) (в данном случае 0x01) и 10 байт, задающих [timestamps](https://ru.wikipedia.org/wiki/Timestamp%22%20%5Co%20%22Timestamp). Вычислить длину поля опции можно через значение поля смещения.

Механизм действия протокола[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=12) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=12)]

В отличие от традиционной альтернативы — UDP, который может сразу же начать передачу пакетов, TCP устанавливает соединения, которые должны быть созданы перед передачей данных. TCP-соединение можно разделить на 3 стадии:

* Установка соединения
* Передача данных
* Завершение соединения

**Состояния сеанса TCP**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=13) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=13)]



Упрощённая диаграмма состояний TCP. Более подробно в [TCP EFSM diagram](http://www.medianet.kent.edu/techreports/TR2005-07-22-tcp-EFSM.pdf) (на английском языке)

|  |
| --- |
| **Состояния сеанса TCP** |
| **CLOSED** | Начальное состояние узла. Фактически фиктивное |
| **LISTEN** | Сервер ожидает запросов установления соединения от клиента |
| **SYN-SENT** | Клиент отправил запрос серверу на установление соединения и ожидает ответа |
| **SYN-RECEIVED** | Сервер получил запрос на соединение, отправил ответный запрос и ожидает подтверждения |
| **ESTABLISHED** | Соединение установлено, идёт передача данных |
| **FIN-WAIT-1** | Одна из сторон (назовём её узел-1) завершает соединение, отправив сегмент с флагом FIN |
| **CLOSE-WAIT** | Другая сторона (узел-2) переходит в это состояние, отправив, в свою очередь сегмент ACK и продолжает одностороннюю передачу |
| **FIN-WAIT-2** | Узел-1 получает ACK, продолжает чтение и ждёт получения сегмента с флагом FIN |
| **LAST-ACK** | Узел-2 заканчивает передачу и отправляет сегмент с флагом FIN |
| **TIME-WAIT** | Узел-1 получил сегмент с флагом FIN, отправил сегмент с флагом ACK и ждёт 2\*MSL секунд, перед окончательным закрытием соединения |
| **CLOSING** | Обе стороны инициировали закрытие соединения одновременно: после отправки сегмента с флагом FIN узел-1 также получает сегмент FIN, отправляет ACK и находится в ожидании сегмента ACK (подтверждения на свой запрос о разъединении) |

**Установка соединения**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=14) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=14)]

Процесс начала сеанса TCP (также называемый «рукопожатие» ([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) *handshake*)), состоит из трёх шагов.

1. Клиент, который намеревается установить соединение, посылает серверу сегмент с номером последовательности и флагом SYN.

* Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буферы и управляющие структуры памяти) для обслуживания нового клиента.
* В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN и ACK, и переходит в состояние SYN-RECEIVED.
* В случае неудачи сервер посылает клиенту сегмент с флагом RST.

2. Если клиент получает сегмент с флагом SYN, то он запоминает номер последовательности и посылает сегмент с флагом ACK.

* Если клиент одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED.
* Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться.
* Если клиент не получает ответа в течение 10 секунд, то он повторяет процесс соединения заново.

3. Если сервер в состоянии SYN-RECEIVED получает сегмент с флагом ACK, то он переходит в состояние ESTABLISHED.

* В противном случае после тайм-аута он закрывает сокет и переходит в состояние CLOSED.

Процесс называется «трёхэтапным согласованием» ([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) *three way handshake*), так как несмотря на то что возможен процесс установления соединения с использованием четырёх сегментов (SYN в сторону сервера, ACK в сторону клиента, SYN в сторону клиента, ACK в сторону сервера), на практике для экономии времени используется три сегмента.

Пример базового 3-этапного согласования:

TCP A TCP B

 1. CLOSED LISTEN

 2. SYN-SENT --> <SEQ=100><CTL=SYN> --> SYN-RECEIVED

 3. ESTABLISHED <-- <SEQ=300><ACK=101><CTL=SYN,ACK> <-- SYN-RECEIVED

 4. ESTABLISHED --> <SEQ=101><ACK=301><CTL=ACK> --> ESTABLISHED

 5. ESTABLISHED <-- <SEQ=301><ACK=101><CTL=ACK> <-- ESTABLISHED

В строке 2 TCP A начинает передачу сегмента SYN, говорящего об использовании номеров последовательности, начиная со 100. В строке 3 TCP B передаёт SYN и подтверждение для принятого SYN в адрес TCP A. Надо отметить, что поле подтверждения показывает ожидание TCP B приёма номера последовательности 101, подтверждающего SYN с номером 100.

В строке 4 TCP A отвечает пустым сегментом с подтверждением ACK для сегмента SYN от TCP B; в строке 5 TCP B передаёт некоторые данные. Отметим, что номер подтверждения сегмента в строке 5 (ACK=101) совпадает с номером последовательности в строке 4 (SEQ=101), поскольку ACK не занимает пространства номеров последовательности (если это сделать, придётся подтверждать подтверждения — ACK для ACK).

Существуют экспериментальные расширения протокола TCP, сокращающие количество пакетов при установлении соединения, например [TCP Fast Open](https://ru.wikipedia.org/w/index.php?title=TCP_Fast_Open&action=edit&redlink=1)[[en]](https://en.wikipedia.org/wiki/TCP_Fast_Open). Ранее также существовало расширение [T/TCP](https://ru.wikipedia.org/wiki/T/TCP). Для прозрачного шифрования данных предлагается использовать расширение [tcpcrypt](https://ru.wikipedia.org/wiki/Tcpcrypt%22%20%5Co%20%22Tcpcrypt).

**Передача данных**[[править](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&veaction=edit&section=15) | [править код](https://ru.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=15)]

*См. также:*[*Алгоритм Нейгла*](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9D%D0%B5%D0%B9%D0%B3%D0%BB%D0%B0)*и*[*Медленный старт*](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%81%D1%82%D0%B0%D1%80%D1%82)

При обмене данными приёмник использует номер последовательности, содержащийся в получаемых сегментах, для восстановления их исходного порядка. Приёмник уведомляет передающую сторону о номере последовательности, до которой он успешно получил данные, включая его в поле «номер подтверждения». Все получаемые данные, относящиеся к промежутку подтверждённых последовательностей, игнорируются. Если полученный сегмент содержит номер последовательности больший, чем ожидаемый, то данные из сегмента буферизируются, но номер подтверждённой последовательности не изменяется. Если впоследствии будет принят сегмент, относящийся к ожидаемому номеру последовательности, то порядок данных будет автоматически восстановлен исходя из номеров последовательностей в сегментах.

Для того, чтобы передающая сторона не отправляла данные интенсивнее, чем их может обработать приёмник, TCP содержит средства управления потоком. Для этого используется поле «окно». В сегментах, направляемых от приёмника передающей стороне, в поле «окно» указывается текущий размер приёмного буфера. Передающая сторона сохраняет размер окна и отправляет данных не более, чем указал приёмник. Если приёмник указал нулевой размер окна, то передача данных в направлении этого узла не происходит, пока приёмник не сообщит о большем размере окна.

В некоторых случаях передающее приложение может явно затребовать протолкнуть данные до некоторой последовательности принимающему приложению, не буферизируя их. Для этого используется флаг PSH. Если в полученном сегменте обнаруживается флаг PSH, то реализация TCP отдаёт все буферизированные на текущий момент данные принимающему приложению. «Проталкивание» используется, например, в интерактивных приложениях. В сетевых терминалах нет смысла ожидать ввода пользователя после того, как он закончил набирать команду. Поэтому последний сегмент, содержащий команду, обязан содержать флаг PSH, чтобы приложение на принимающей стороне смогло начать её выполнение.