

Протоколы UDP и TCP

Artem Beresnev
t.me/ITSMDao

План

- ▶ Транспортный уровень
- ▶ Номера портов
- ▶ Протокол UDP
- ▶ Протокол TCP
- ▶ Установление соединения в TCP
- ▶ Управление скоростью в TCP
- ▶ Будущее TCP

Транспортный уровень ТСП\IP



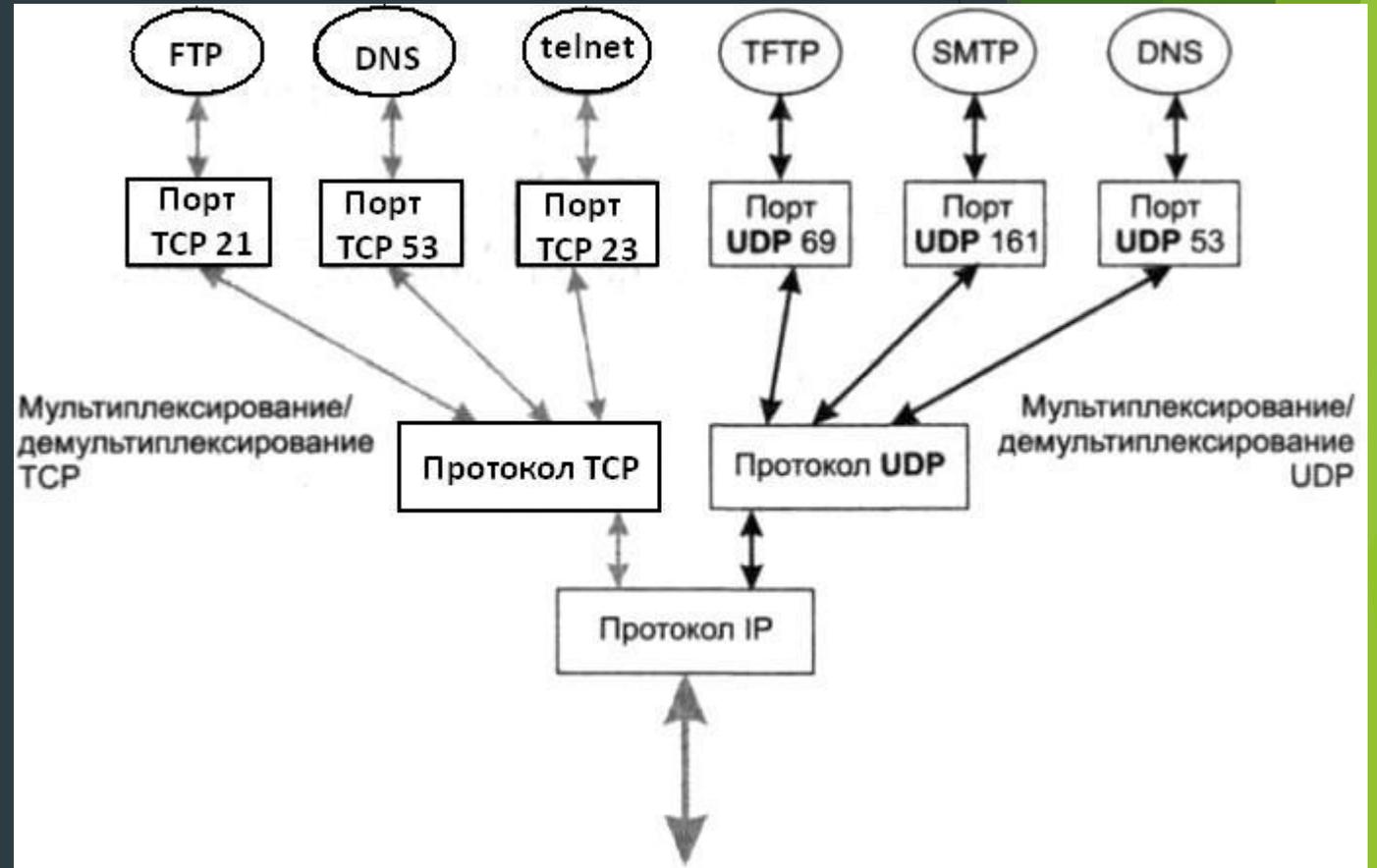
- ▶ Задачи:
 - ▶ Диспетчеризация данных между приложениями
 - ▶ Достоверность доставки (TCP)
- ▶ Адреса TCP и UDP - номера портов
- ▶ Надежная доставка - TCP
- ▶ Не надежная доставка - «чистый» UDP



Номера портов

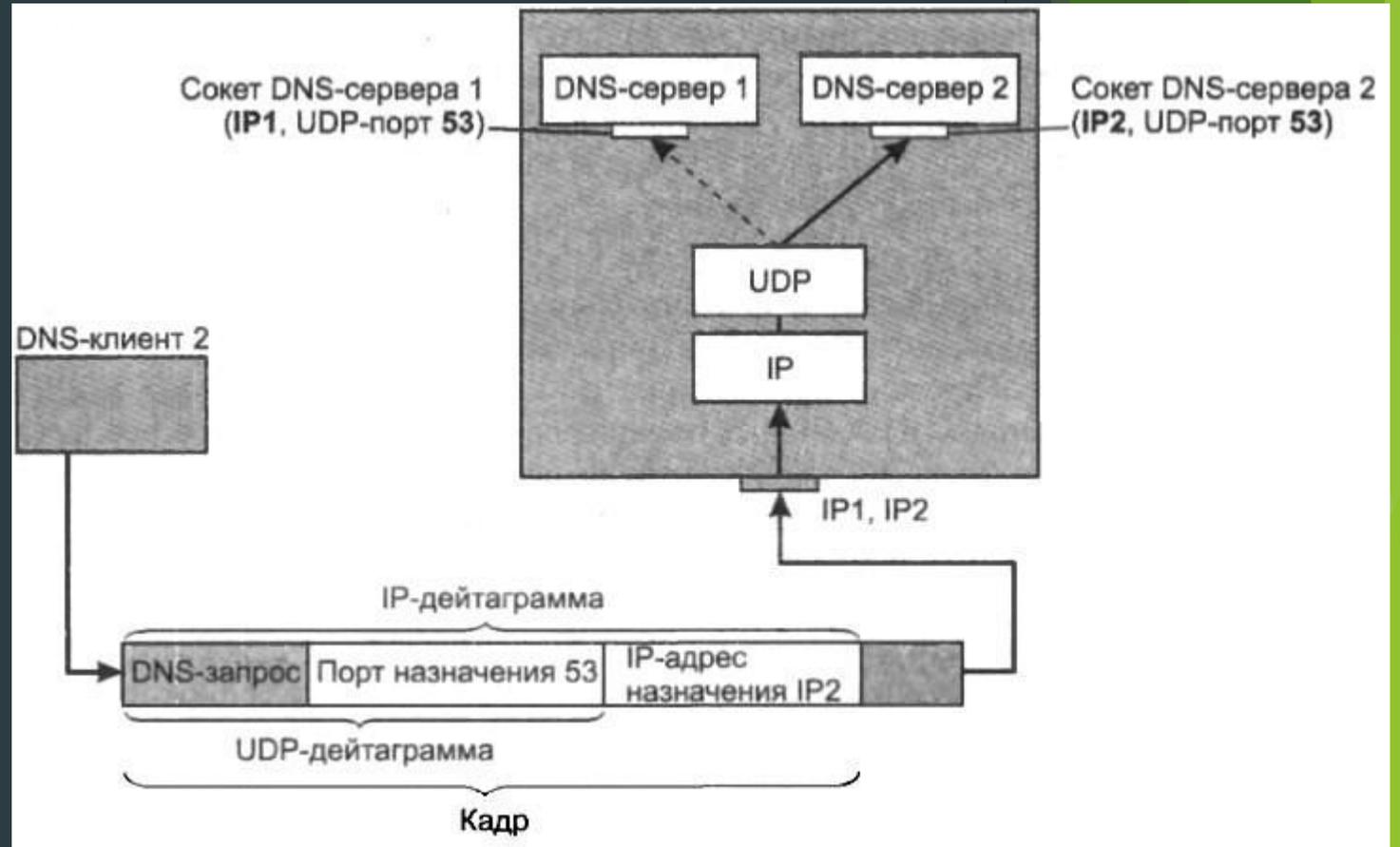
Мультиплексирование и демультимплексирование

- ▶ Порт «занимается» приложением, клиентом или сервером.
- ▶ Модули протоколов TCP и UDP занимаются направлением инкапсулированных данных в нужный порт.
- ▶ Номера портов для UDP и TCP могут быть одинаковыми.



Мультиплексирование и демultipлексирование

- ▶ В случае, если на хосте несколько IP адресов, то порт для разных приложений может совпадать.
- ▶ Но различаются сокеты (!)



Значения номеров портов

- ▶ Для сервера номер порта фиксирован, для клиента – произволен.
- ▶ Для серверов:
 - ▶ RFC порты (назначается IANA RFC 6335)
 - ▶ Не RFC порты
 - ▶ Пример:
 - ▶ 80/TCP – HTTP официально
 - ▶ 81/TCP – Tor не официально
- ▶ Клиентские порты – выше 1024.

Значения номеров портов

- ▶ Размер номера порта 16 бит, следовательно максимальное значение 65535.
- ▶ Порт с номером 0 разрешен для использования в UDP, запрещен для TCP.

Безопасность и номера портов

- ▶ Сканирование портов (nmap)
- ▶ Перенос служб на не RFC порты.



Протокол UDP

Заголовок UDP



User Datagram Protocol, RFC-768

Заголовок UDP

Заголовок UDP состоит из четырех 2-байтных полей:

- ▶ номер UDP-порта отправителя;
- ▶ номер UDP-порта получателя;
- ▶ контрольная сумма;
- ▶ длина дейтаграммы

Заголовок UDP

- ▶ Контрольная сумма
 - ▶ Служит для диагностики, но не для исправления.
 - ▶ Поврежденную дейтаграмму UDP просто отбрасывает.



Протокол ТСР

Заголовок ТСП

- ▶ Transmission Control Protocol
- ▶ RFC: 793, 1323, 1644, 2018, 2581, 2582, 2861, -2873, 2883, 2923, 2988, 3465,3481 и др.
 - ▶ Устанавливает и контролирует соединение
 - ▶ Делит данные прикладного уровня на части - сегменты
 - ▶ Осуществляет передачу данных с уведомлением
 - ▶ Вычисляет контрольные суммы по всему сегменту.

Заголовок ТСР



Заголовок ТСР

- ▶ Порт отправителя
- ▶ Порт получателя

Заголовок ТСР

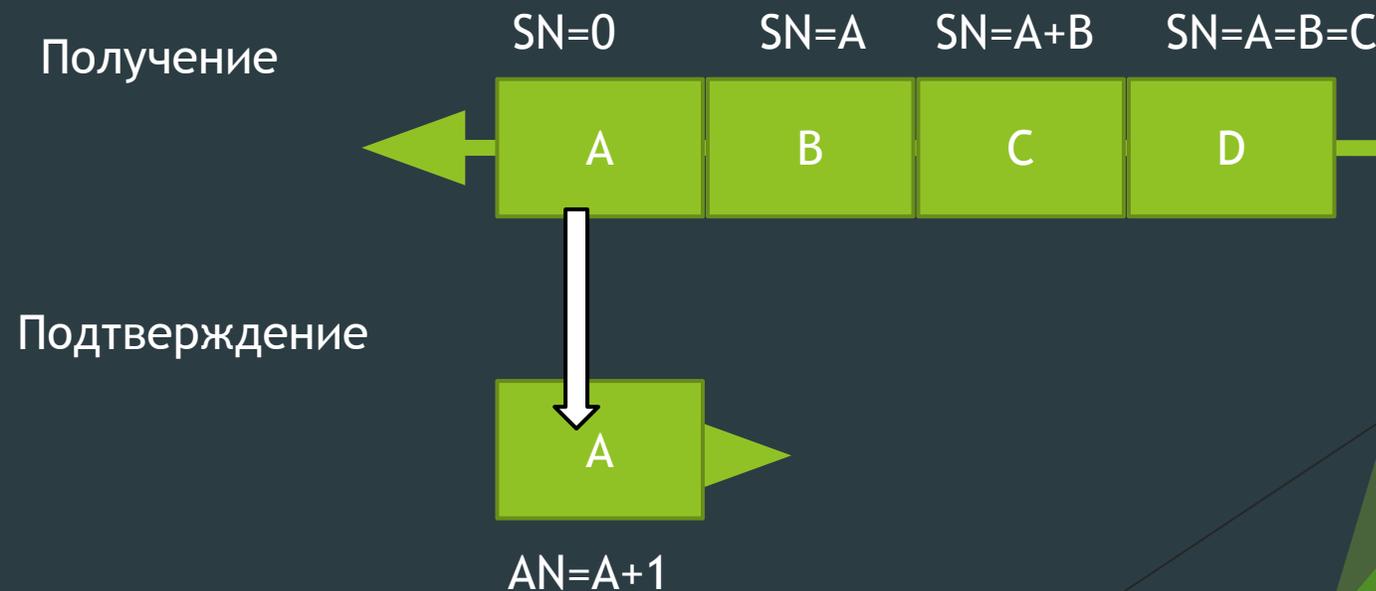
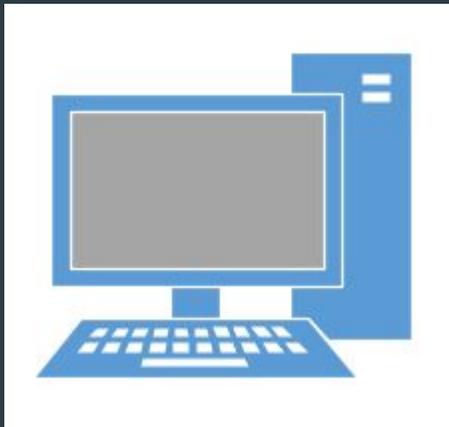


Заголовок ТСР

- ▶ Код позиции в сообщении (*sequence number*) - номер первого байта данных в сегменте. По сути это смещение сегмента относительно начала данных.
- ▶ Номер октета, который должен прийти следующим (*acknowledgment number*) - число, которое на единицу больше номера последнего успешно принятого байта.

Заголовок ТСР

- ▶ Код позиции в сообщении (sequence number)
- ▶ Номер октета, который должен прийти следующим (acknowledgment number)



Восстановление потерянных сегментов

- ▶ Реализации ТСР используют два разных способа, чтобы получатель мог запросить один пакет.
- ▶ Тройное признание. Если получатель трижды подтверждает пакет, который предшествует последнему подтвержденному серийному номеру, отправитель предполагает, что получатель запрашивает повторную передачу пакета.
- ▶ Второй способ заключается в реализации выборочных подтверждений (SACK). SACK добавляет новое поле к подтверждению ТСР, которое позволяет получателю подтвердить получение определенного набора серийных номеров.

Заголовок ТСР

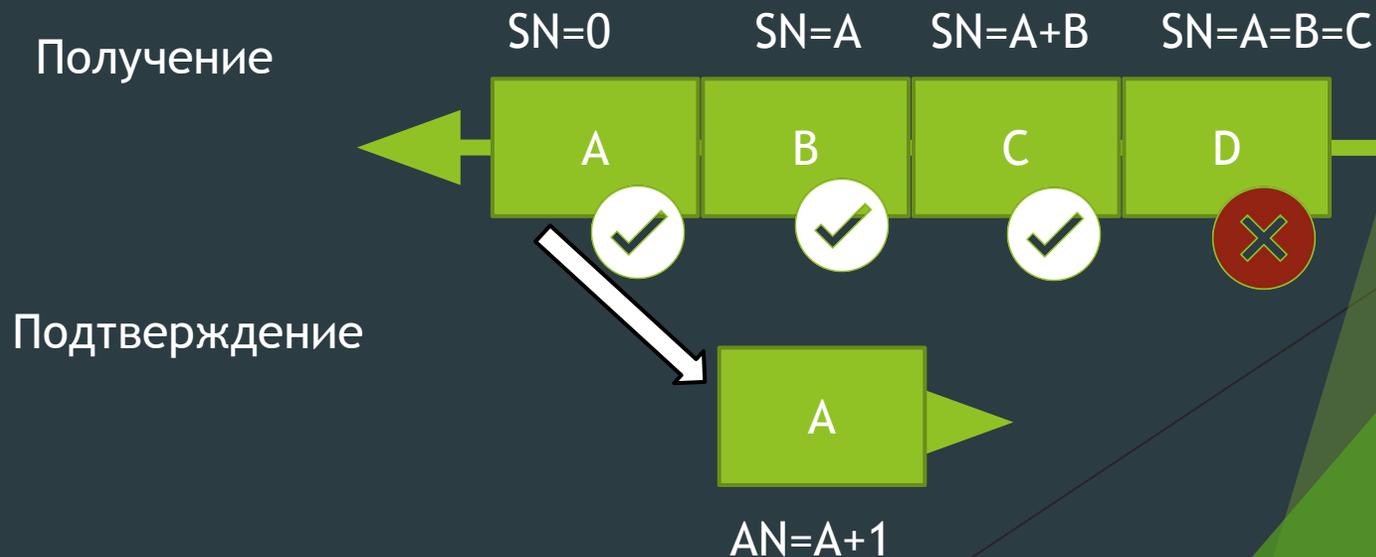


Заголовок TCP

- ▶ **HLLEN** - поле, определяющее размер заголовка пакета TCP в 4-байтных словах. Минимальный размер составляет 5 слов, а максимальный – 15, что составляет 20 и 60 байт соответственно. Смещение считается от начала заголовка TCP.
- ▶ Резерв - предназначенное для будущего использования (сейчас = 0) кроме 4 и 6 бита, где располагаются флаги управления перегрузкой
- ▶ **Флаги (обсудим далее)**
- ▶ Размер окна - определяет число сегментов, которые могут быть посланы без получения подтверждения.
- ▶ Указатель важной информации - представляет собой указатель последнего байта, содержащий информацию, которая требует немедленного реагирования.

Размер окна

- ▶ Размер окна = $A+B+C$
- ▶ Это значит, что отправитель может послать сегменты A, B, C до получения подтверждения на сегмент A
- ▶ После получения подтверждения A окно сдвигается вправо. Можно отправить D

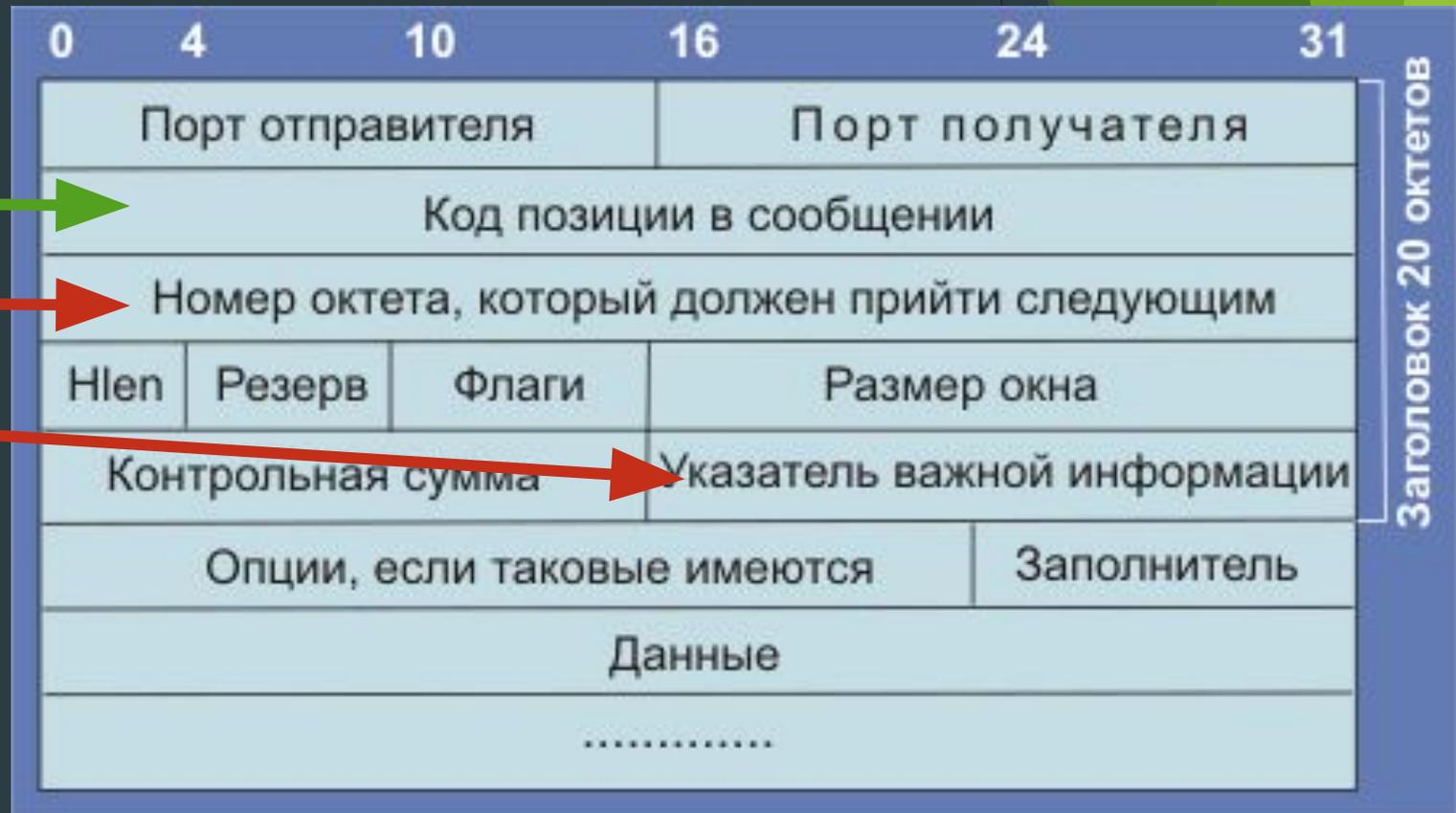


Заголовок ТСР. Флаги.

- ▶ URG — срочное сообщение;
- ▶ ACK — квитанция на принятый сегмент;
- ▶ PSH — запрос на отправку сообщения без ожидания заполнения буфера;
- ▶ RST — запрос на восстановление соединения;
- ▶ SYN — сообщение, используемое для синхронизации счетчиков переданных данных при установлении соединения;
- ▶ FIN — признак достижения передающей стороной последнего байта в потоке передаваемых данных

Заголовок TCP. Флаги.

- ▶ SYN
- ▶ ACK
- ▶ URG





Установка соединения в ТСР

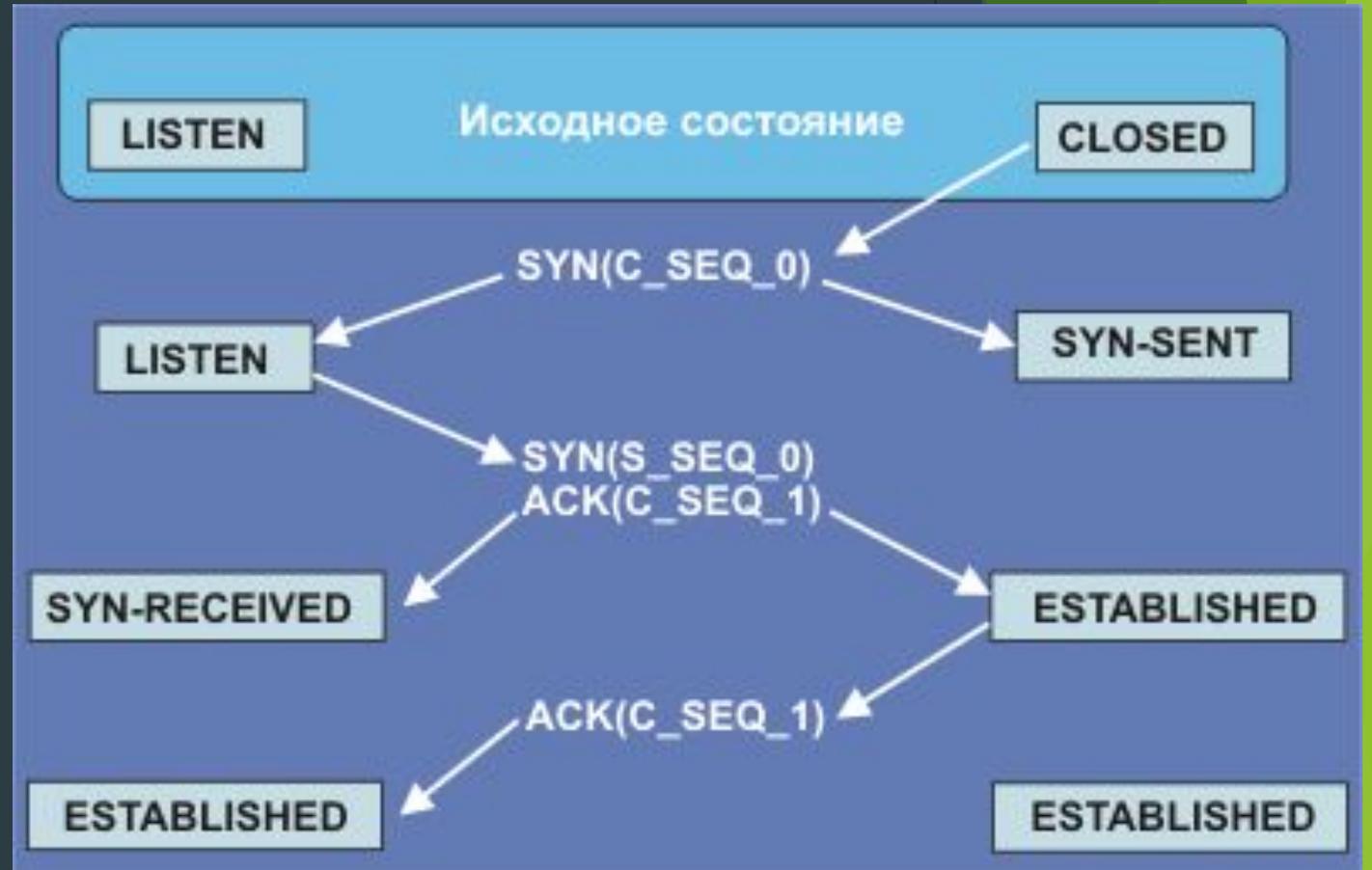
Установление соединения в ТСР

- ▶ Клиент и сервер ТСР проходят последовательно ряд состояний.
- ▶ Переход между состояниями управляется получением\отправкой сегментов с определенными флагами и значениями полей.

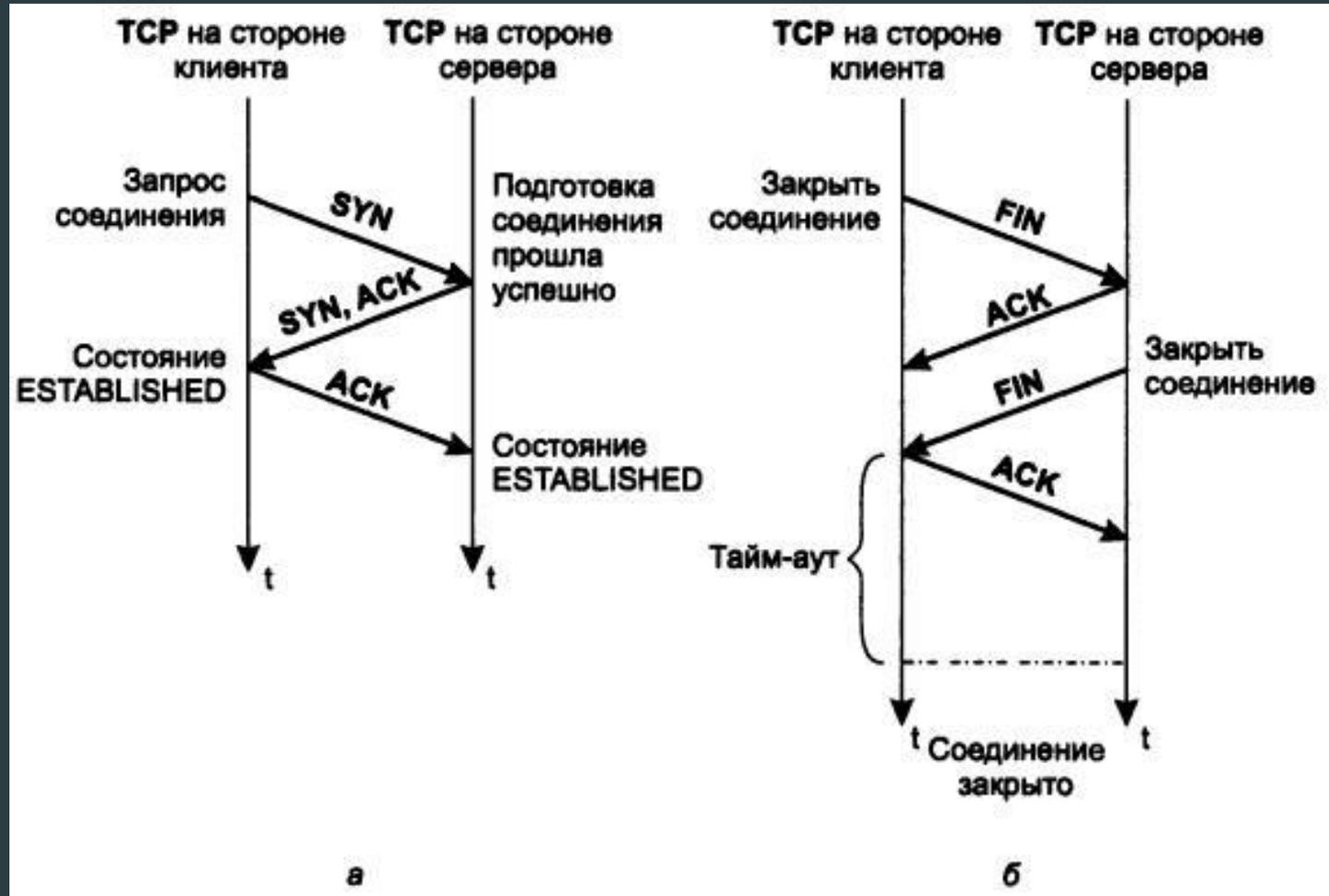
Установка соединения в TCP

Тройное рукопожатие.

- ▶ Клиент посылает SYN-сегмент (active open).
- ▶ Сервер откликается, посылая свой SYN-сегмент, с ISN (Initial Sequence Number) = 0 (passive open).
- ▶ Клиент отправляет подтверждение получения SYN-сегмента от сервера с идентификатором равным ISN (сервера)+1.



Заккрытие соединения



netstat

```
TCP      192.168.0.100:59343      5.255.255.55:443      ESTABLISHED
TCP      192.168.0.100:59370      34.218.161.49:443      ESTABLISHED
TCP      192.168.0.100:59409      51.89.64.182:80        ESTABLISHED
TCP      192.168.0.100:59413      217.69.139.59:443      ESTABLISHED
TCP      192.168.0.100:59474      52.109.88.76:443      ESTABLISHED
TCP      192.168.0.100:59478      52.109.88.76:443      ESTABLISHED
TCP      192.168.0.100:59482      193.0.170.54:443      TIME_WAIT
TCP      192.168.0.100:59483      5.61.23.11:443        TIME_WAIT
TCP      192.168.0.100:59484      94.100.180.215:443     TIME_WAIT
TCP      192.168.0.100:59485      52.109.12.20:443      ESTABLISHED
TCP      192.168.0.100:59486      31.13.72.52:443      ESTABLISHED
TCP      0.0.0.0:135             0.0.0.0:0              LISTENING
TCP      0.0.0.0:445             0.0.0.0:0              LISTENING
```



Регулирование скорости в ТСП

От чего зависит скорость ТСР?

- ▶ Скорость передачи данных по ТСР не равномерная. Она меняется в зависимости от:
 - ▶ времени от начала передачи,
 - ▶ успешности передачи,
 - ▶ общих условий передачи.

Размер окна



Размер окна

$$W=RTT*B$$

- ▶ W - размер окна в
- ▶ B - Полоса пропускания (бит/сек)
- ▶ RTT (*round-trip time*) — Время приема-передачи - это время на отправку сигнала + время на получение подтверждения.
- ▶ Размеры окон для обоих участников обмена могут быть разными

Определение интенсивности передачи

- ▶ Трафик регулируется:
 - ▶ Со стороны получателя (контроль доставки) с помощью параметра window
 - ▶ Со стороны отправителя (контроль перегрузки) с помощью окна перегрузки cwnd (congestion window) и порога медленного старта ssthresh (Slow start threshold).
- ▶ Окно перегрузки (CWND) позволяет согласовать полную загрузку виртуального соединения и текущие возможности канала, минимизируя потери пакетов при перегрузке.
- ▶ Увеличение окна перегрузки идет по мере успешной передачи
- ▶ При неуспешной передаче окно схлопывается (при ошибках или таймауте).

TCP Congestion Control

- ▶ BIC TCP
- ▶ CUBIC TCP
- ▶ Highspeed TCP
- ▶ H-TCP
- ▶ TCP Hybla
- ▶ TCP-Illinois
- ▶ TCP Low Priority
- ▶ TCP Vegas
- ▶ TCP NewReno
- ▶ TCP Veno
- ▶ TCP Westwood+
- ▶ YeAH-TCP

Окно перегрузки 1\2

- ▶ Congestion Window – окно перегрузки
 - ▶ Чаще всего равно размеру окна передачи
 - ▶ определяет размер буфера данных, доступных для передачи.
 - ▶ поддерживается отправителем и является средством предотвращения перегрузки канала между отправителем и получателем из-за слишком большого объема трафика.
 - ▶ в современных ОС может достигать максимального значения в 65535

Окно перегрузки 2\2

- ▶ Когда соединение установлено:
 - ▶ окно перегрузки независимо на каждом хосте, устанавливается на небольшое значение, кратное MSS (Maximum Segment Size), разрешенному для этого соединения.
 - ▶ в дальнейшем, в зависимости от алгоритма управления перегрузкой, при условии, что все сегменты получены и подтверждения достигают отправителя вовремя, к размеру окна добавляется некоторая константа.
 - ▶ когда окно достигает ssthresh, окно перегрузки увеличивается линейно при каждом новом полученном подтверждении,
 - ▶ окно перегрузки схлопывается до 1 MSS (возможны другие варианты, например до половины ssthresh) при обнаруженных проблемах передачи, выключая таймаут прихода подтверждения.



Критика ТСР

Критика ТСП

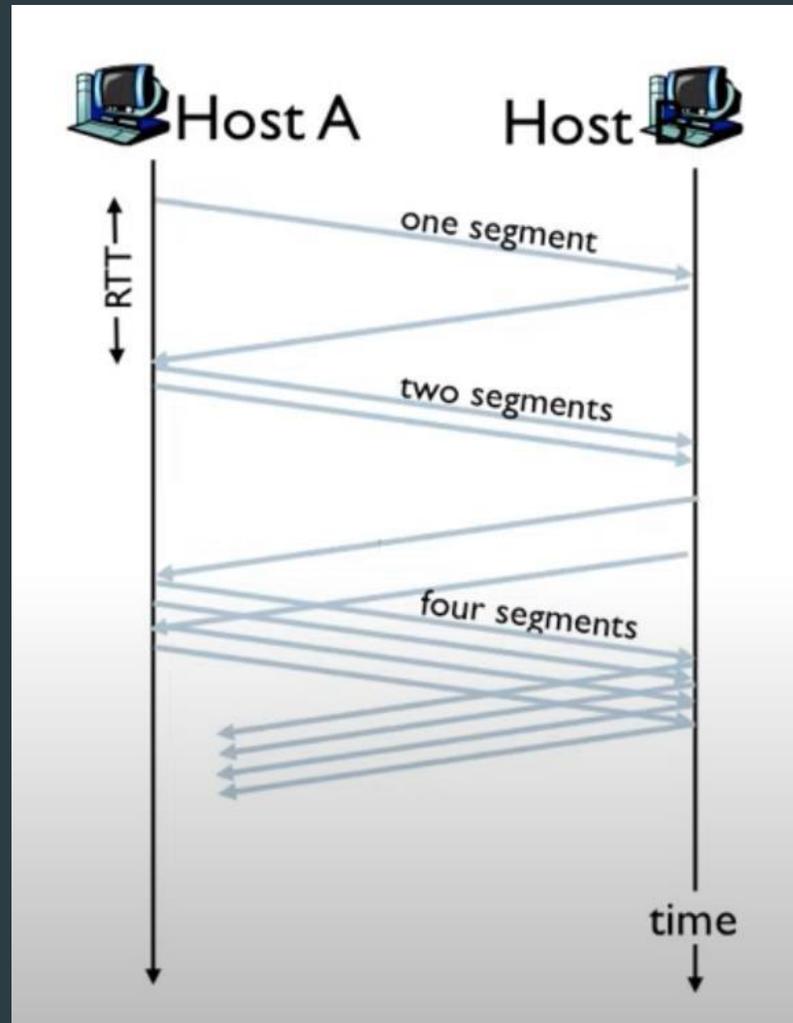
- ▶ Проблема медленного открытия соединений (ТСП, ТСП+ТЛС)
- ▶ Проблема медленного старта
- ▶ Слабая адаптация ТСП к беспроводным сетям
- ▶ Критическое влияние не пропускной способности канала, а задержки

https://www.youtube.com/watch?v=RqZ9_Yq-ibM

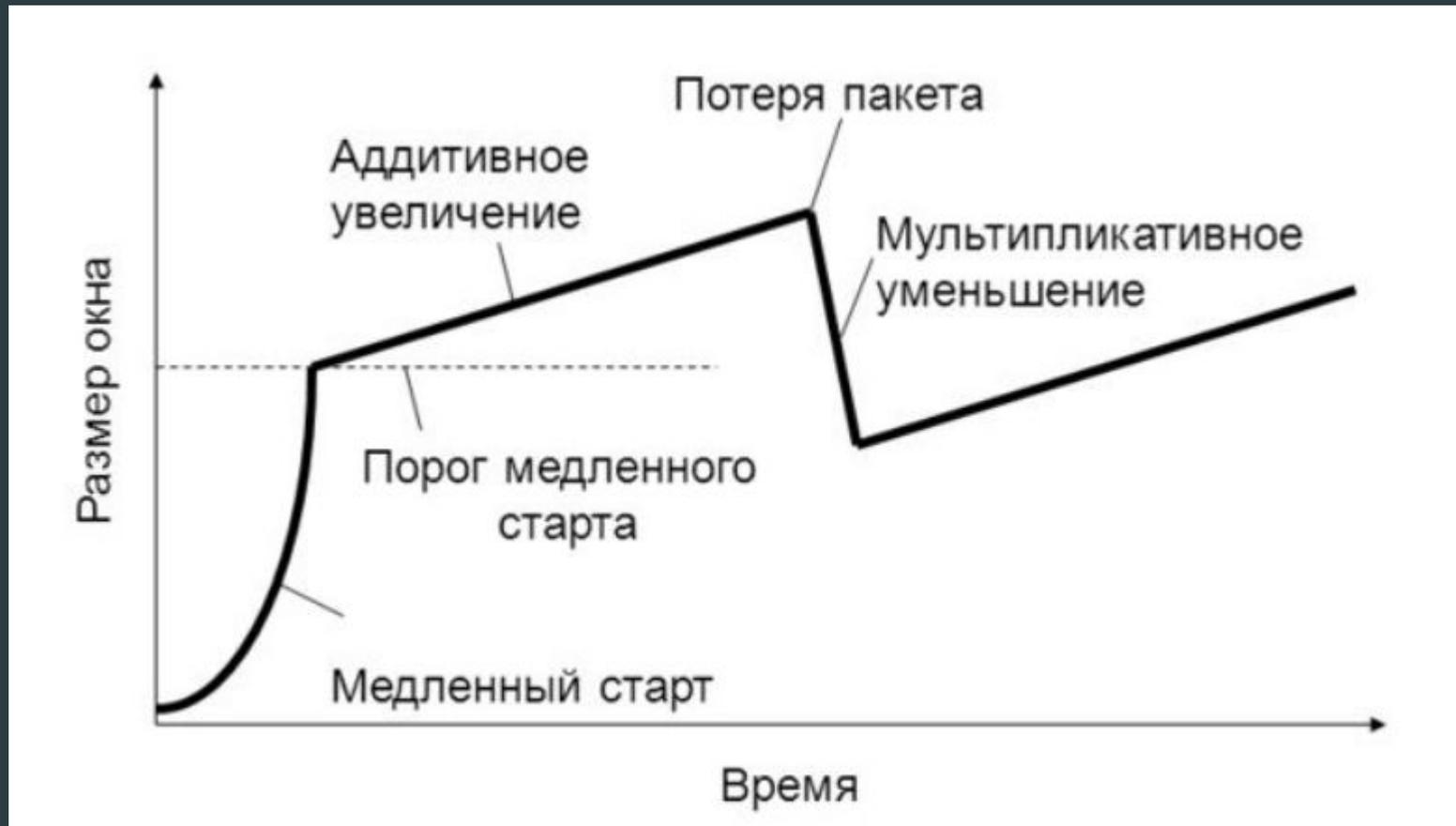
<https://www.youtube.com/watch?v=1lh0bL2Zp1c>

Медленный старт

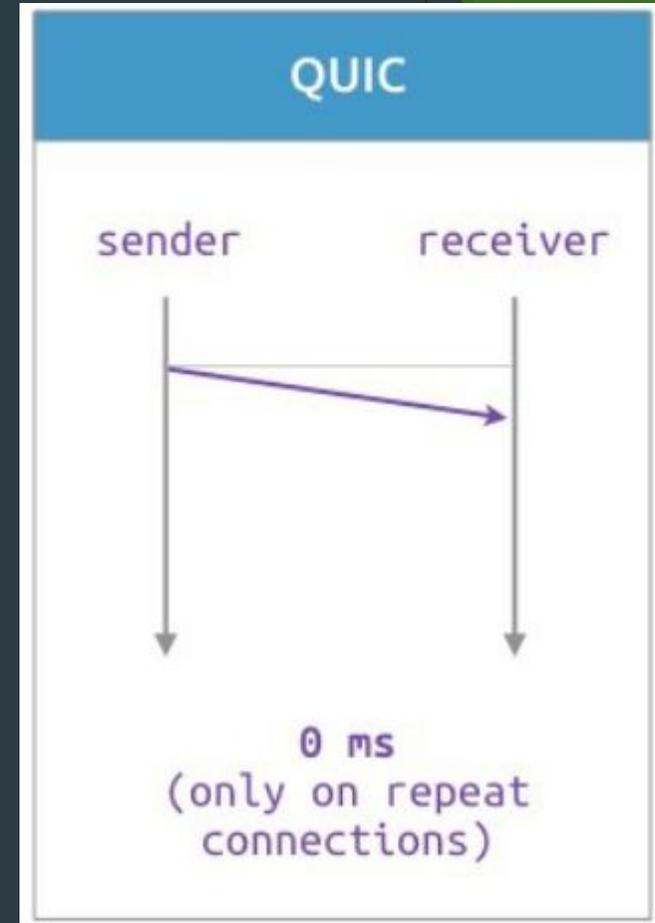
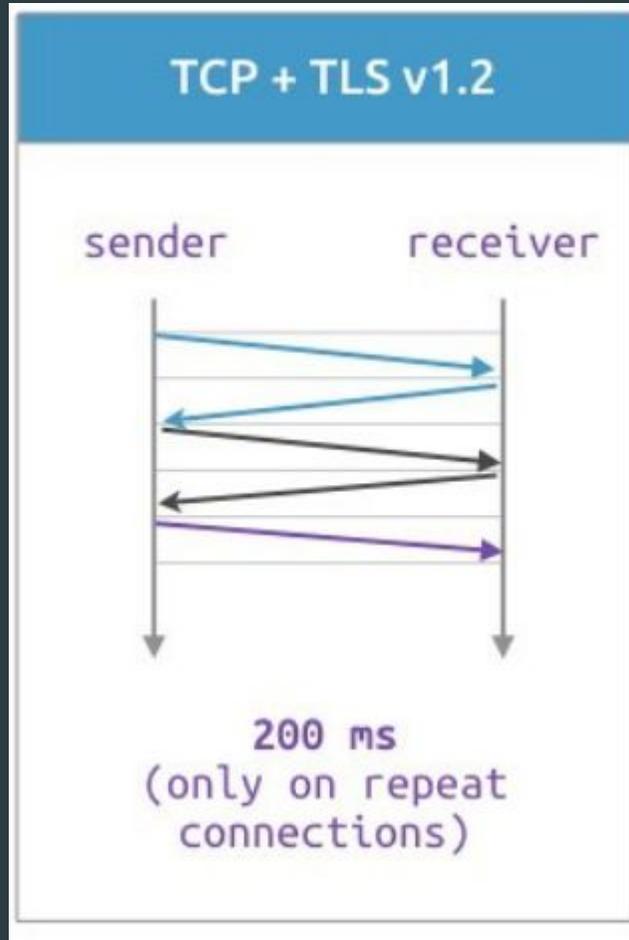
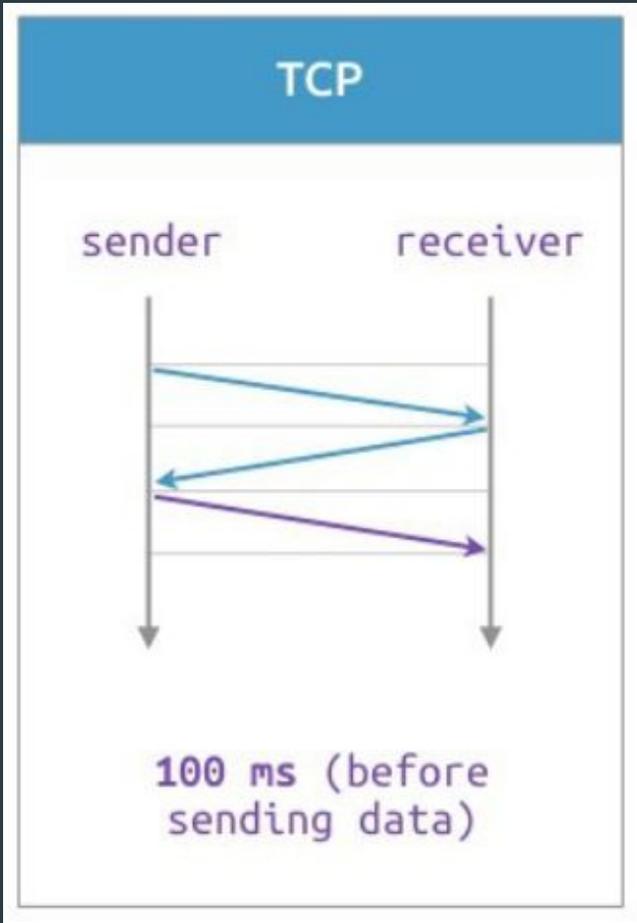
- ▶ $cwnd=1$ MSS
- ▶ Пусть мы будем удваивать $cwnd$



Управление окном перегрузки



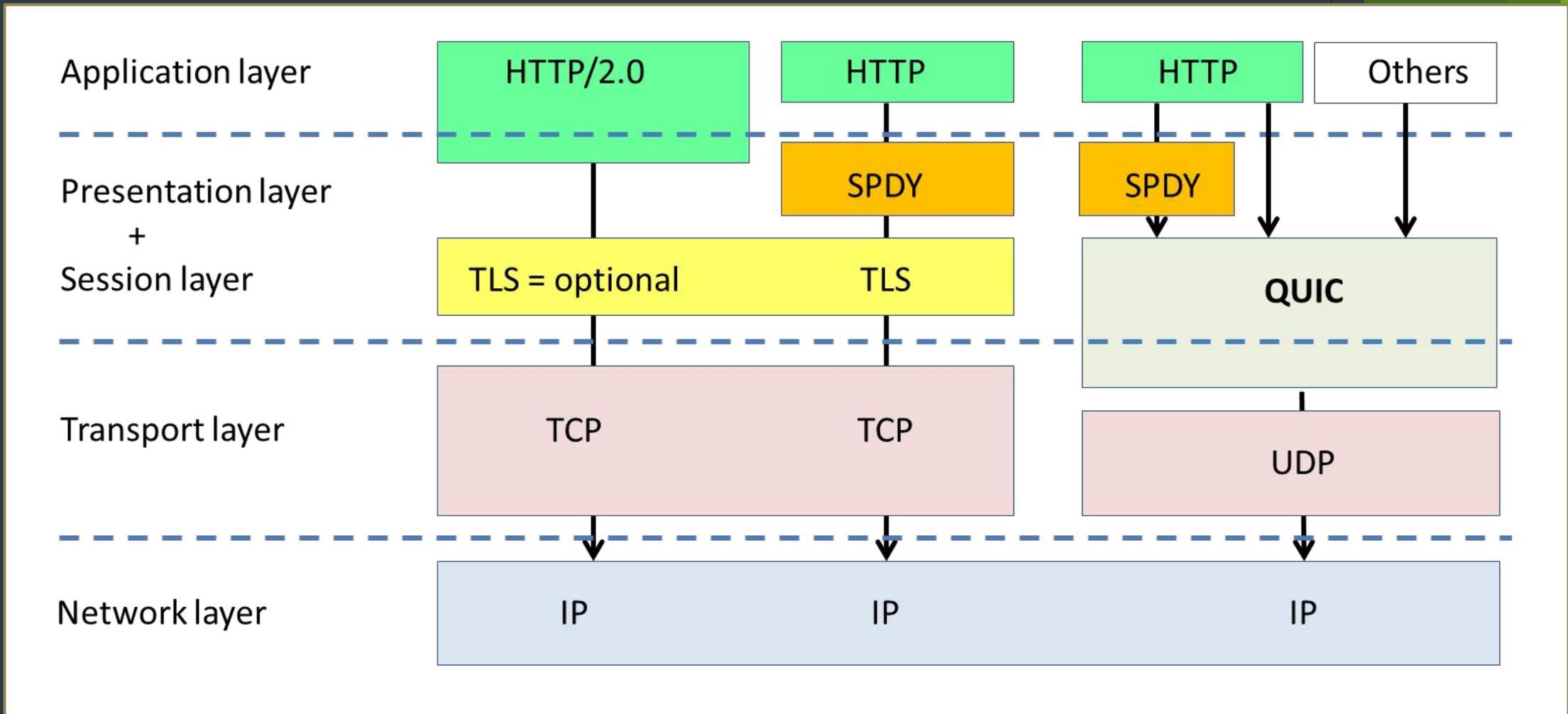
Переход на QUIC



https://infostart.ru/journal/news/tekhnologii/novyy-protokol-http-3-bolshe-ne-budet-ispolzovat-tcp-na-ocheredi-quic_1284071/

- TCP handshake
- Data
- TLS v1.2 setup

Переход на UDP



Выводы

- ▶ Транспортный уровень
- ▶ Номера портов
- ▶ Протокол UDP
- ▶ Протокол TCP
- ▶ Установление соединения в TCP
- ▶ Управление скоростью в TCP
- ▶ Будущее TCP - постепенный переход на QUIC в ряде приложений