

Транспортный уровень OSI

Протоколы TCP и UDP

Транспортный уровень

Transport Layer

- Транспортный уровень определяет сервисы для сегментации, передачи и сборки данных для отдельных сообщений между оконечными устройствами
- Обеспечивает приложениям (или верхним уровням стека – прикладному и сеансовому) передачу данных с требуемой степенью надёжности, где критерии надёжности:
 - срочность
 - возможность восстановления прерванной связи
 - возможность исправления ошибок передачи
- Примеры протоколов: TCP, UDP
- Единица данных – *дейтаграмма/блок данных (datagram)*

Протоколы транспортного уровня стека TCP/IP

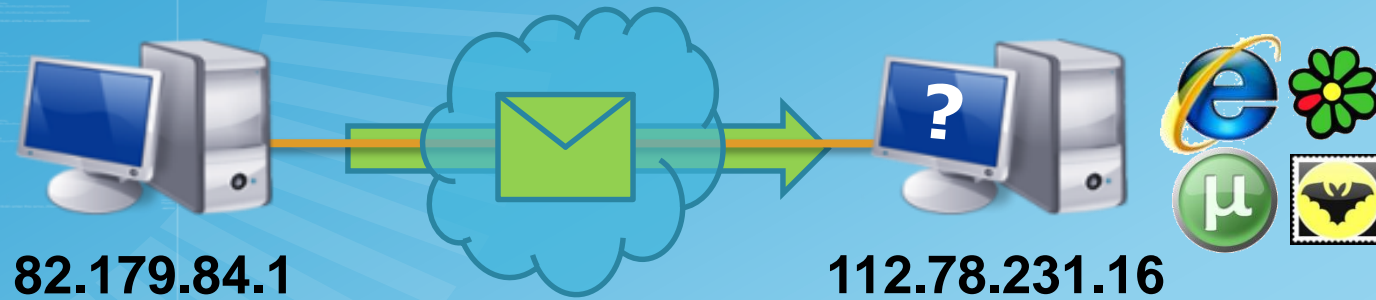
□ Transmission control protocol (TCP)

- с установкой логического соединения до передачи данных
- с подтверждением доставки данных
- надёжность в ущерб скорости

□ User datagram protocol (UDP)

- без установки логического соединения
- без подтверждения доставки данных
- скорость в ущерб надёжности

Порты и сокеты



- **Порт** – системная очередь пакетов к точке входа прикладного процесса (не путать с портами оборудования: USB, COM, PS/2 и т.д.)

Сокет
(socket)

- IP-адрес однозначно идентифицирует сетевой интерфейс в пределах составной сети
- Порт однозначно идентифицирует прикладной процесс в пределах компьютера

Способы присвоения портов

- Централизованный способ (0-1023) – для популярных общедоступных служб (HTTP, FTP, telnet, DNS, SMTP и т.д.) – т.н. «младшие порты»

- IANA (Internet assigned numbers authority)
- RFC 1700

- Локальный способ (1024-65535) – для служб, еще не ставших столь распространенными, чтобы закреплять за ними стандартные номера (т.н. «старшие порты»)

- 1024 - 49151 – зарегистрированные
- 49152 - 65535 – динамические

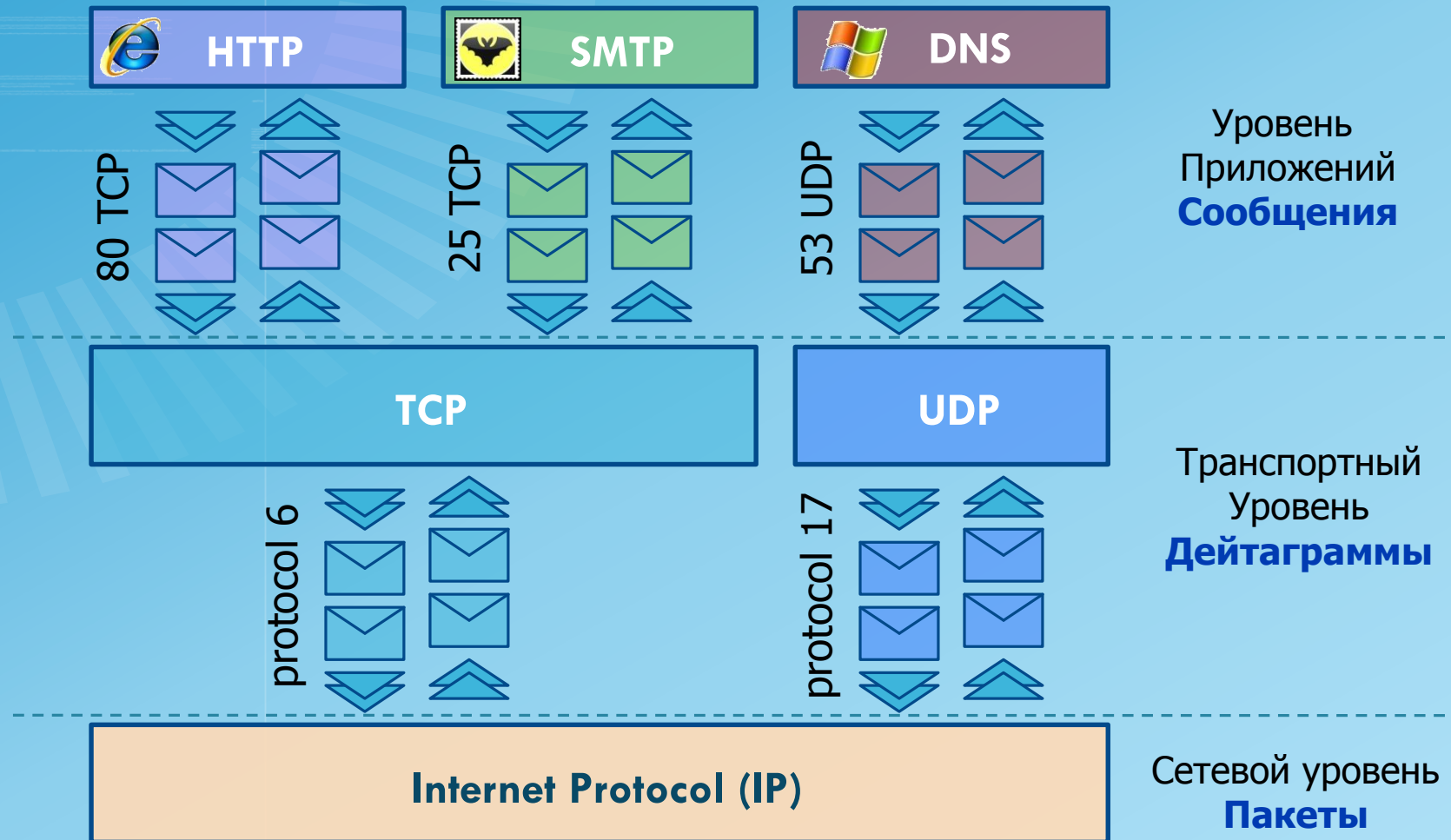
```
ssh = 22 TCP
telnet = 23 TCP
http = 80 TCP
RIP = 520 UDP
```

...

(Де)мультиплексирование (1)

- Множества TCP и UDP портов не пересекаются (независимы):
порты TCP1750 \neq UDP1750
- Для каждого порта существует 2 очереди дейтаграмм:
входящая и исходящая
- **Мультиплексирование** – процедура приёма протоколом TCP (UDP) данных, поступающих от нескольких прикладных служб (по портам), и формирования единого потока дейтаграмм
- **Демультиплексирование** – распределение протоколом TCP (UDP) поступающих с сетевого уровня пакетов между набором высокоуровневых служб, идентифицированных номерами портов

(Де)мультиплексирование (2)

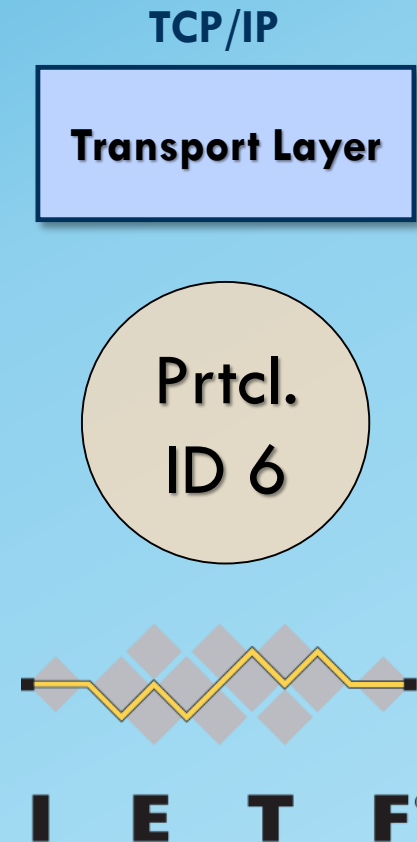


TCP – Transmission Control Protocol

■ Особенности:

- установка соединения (сессии)
- надёжная доставка (контроль доставки и автоматическая пересылка потерявшихся сегментов)
- доставка сегментов и сборка их в том же порядке, в каком происходило сегментирование
- управление потоком данных

■ RFC 793



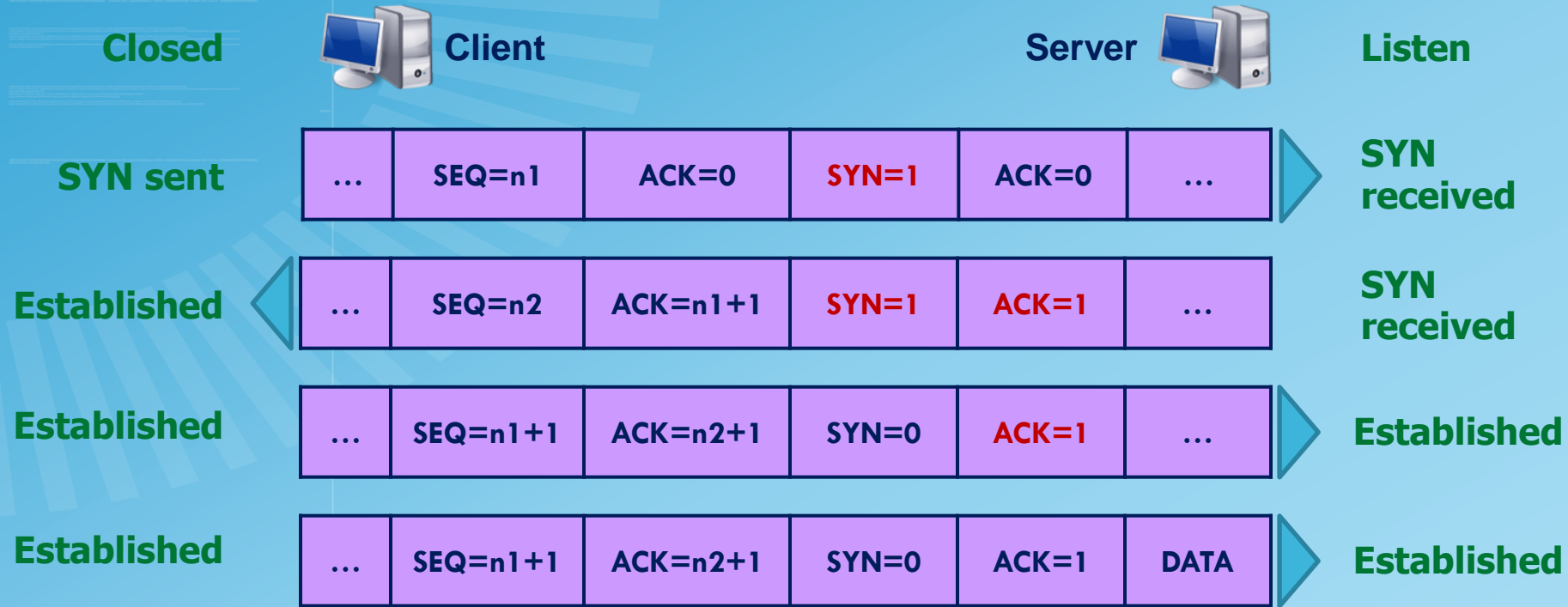
TCP-сегмент

	биты 0-3	4-7	8	9	10	11	12	13	14	15	16-18	19-31	
0	Source port (SP) – номер порта отправителя										Destination port (SP) – номер порта получателя		
32	Sequence number (SEQ, номер последовательности) – последний порядковый номер байта, переданный в рамках данного соединения (исходящий поток)												
64	Acknowledgement number (ACK, номер подтверждения) – последний порядковый номер байта в рамках данного соединения, корректное получение которого подтверждается в данном сегменте (входящий поток)												
96	Data offset – смещение данных	Reserved	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window size (размер окна) – количество байт, которое получатель готов принять		
128	Checksum – контрольная сумма										Urgent pointer – указатель важности		
160	Options (только если Data offset>5) – дополнительные параметры												
160 (192+)	DATA – данные верхнего уровня												

Логические соединения

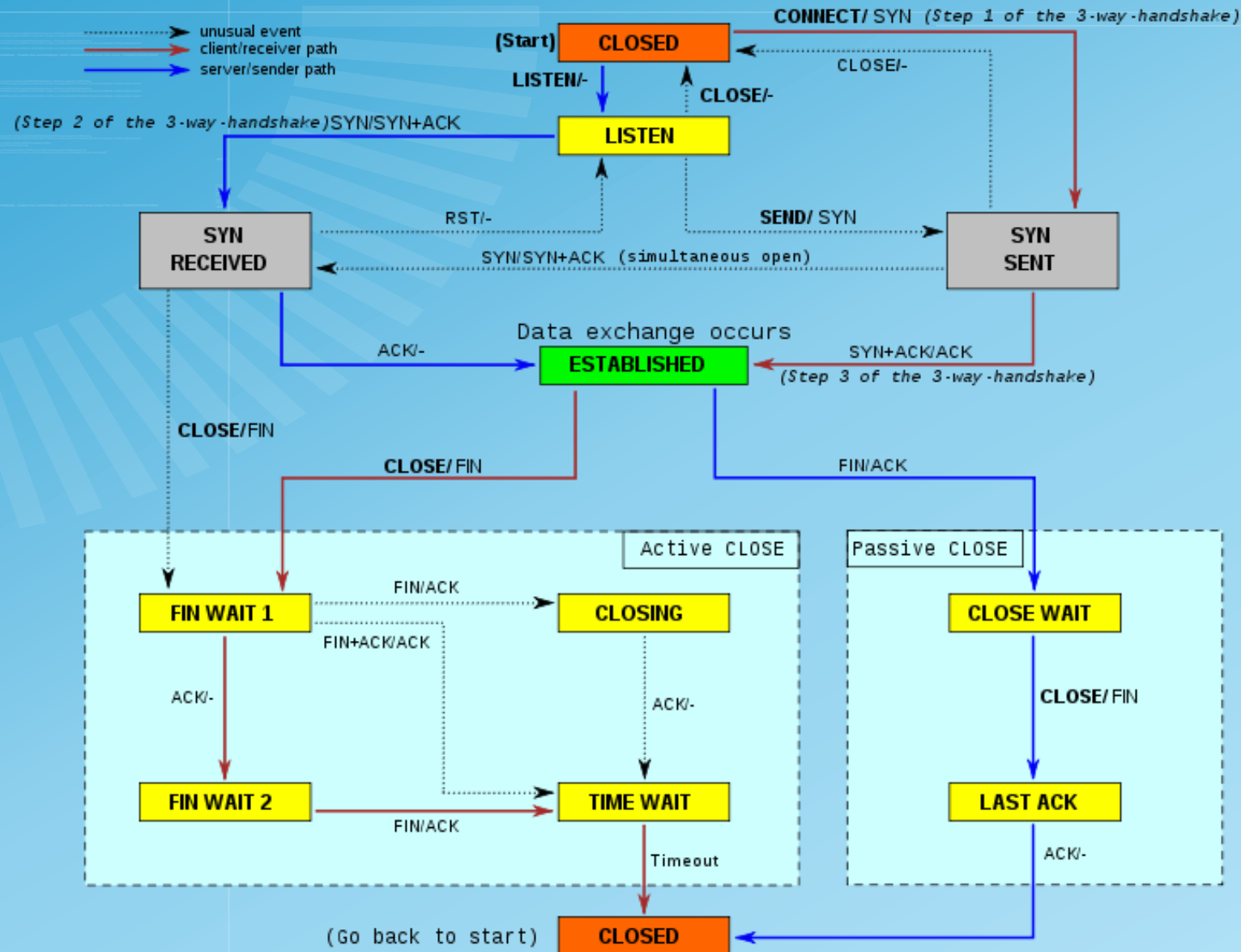
- **Сокет** (socket) – набор идентифицирующих параметров. Включает IP-адрес и номер порта.
 - Каждый взаимодействующий процесс однозначно идентифицируется сокетом в пределах составной сети.
- Соединение идентифицируется парой сокетов $\{(IP1, n1), (IP2, n2)\}$ и представляет собой договоренность о параметрах процедуры обмена данными между процессами:
 - Максимальный размер принимаемого сегмента [Байт]
 - Максимальный объем данных, который разрешено передавать до получения подтверждения о доставке предыдущего набора данных (размер «окна») [Байт]
 - Начальный порядковый номер байта, с которого начинается отсчёт потока данных в рамках соединения

Тройное рукопожатие



- Тройное рукопожатие (3-way handshake) – процесс установки TCP-соединения, представляющий собой последовательный обмен сегментами с флагами SYN (запрос на синхронизацию номеров SEQ) и ACK (подтверждение синхронизации)

Диаграмма состояний ТСП

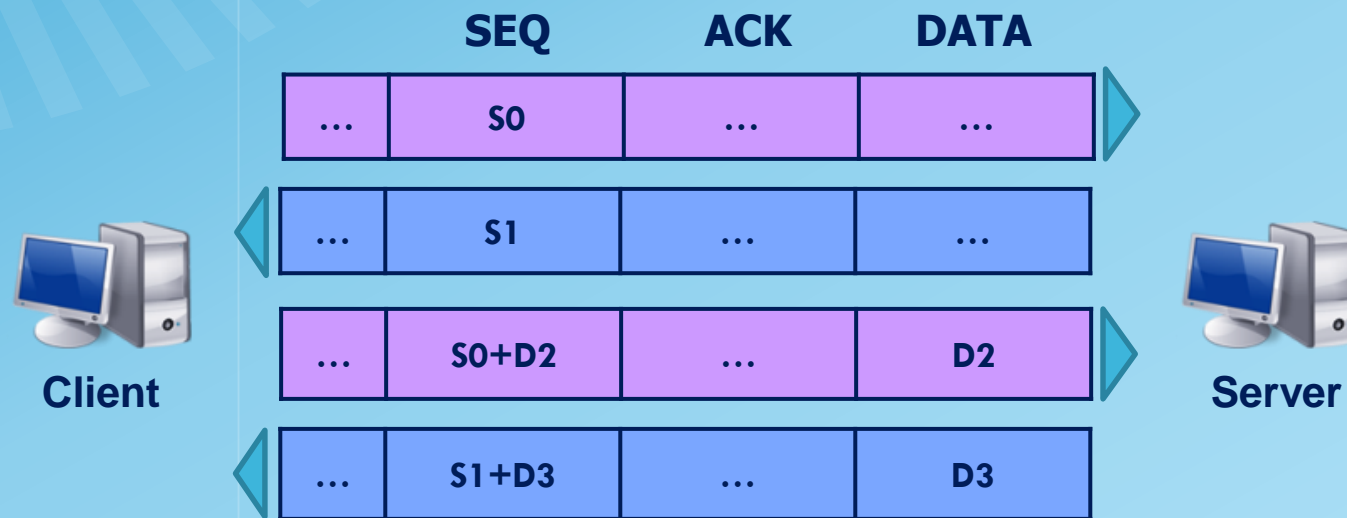


Квитирование

- **Квитирование** – в протоколе TCP процесс подтверждения доставки данных с использованием полей сегмента SEQ (номер байта в последовательности) и ACK (номер байта-подтверждения, не путать с флагом ACK)
- Учет переданных байт осуществляется в обоих направлениях независимо

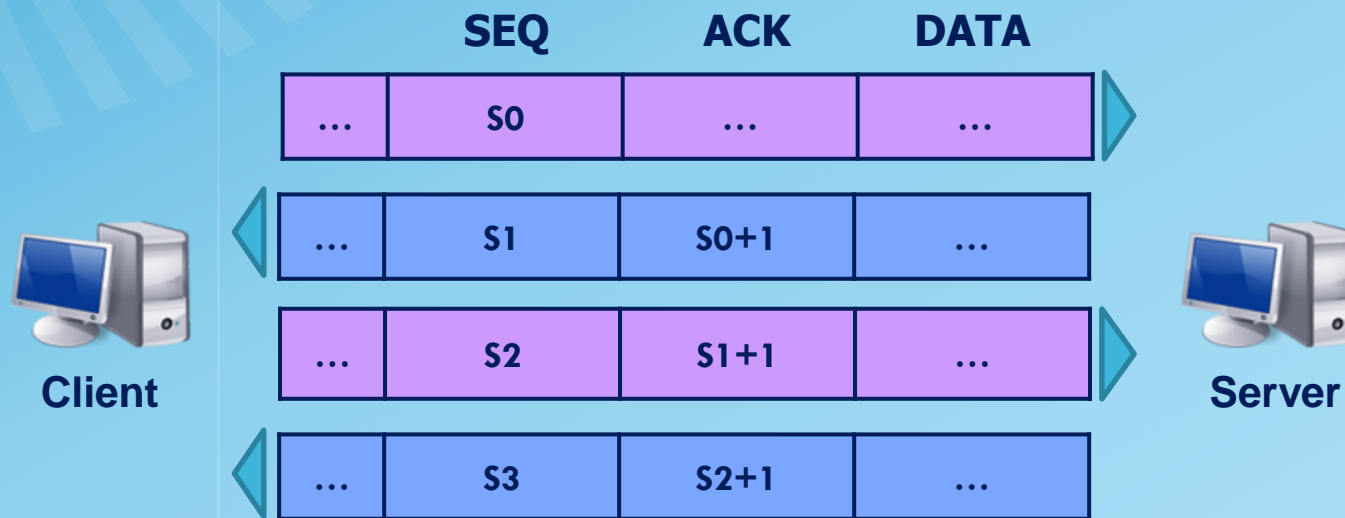
Квитирование – SEQ

- Поле SEQ указывает последний порядковый номер переданного байта
- При безошибочной передаче
$$SEQ_n = SEQ_{n-1} + DATA_n$$



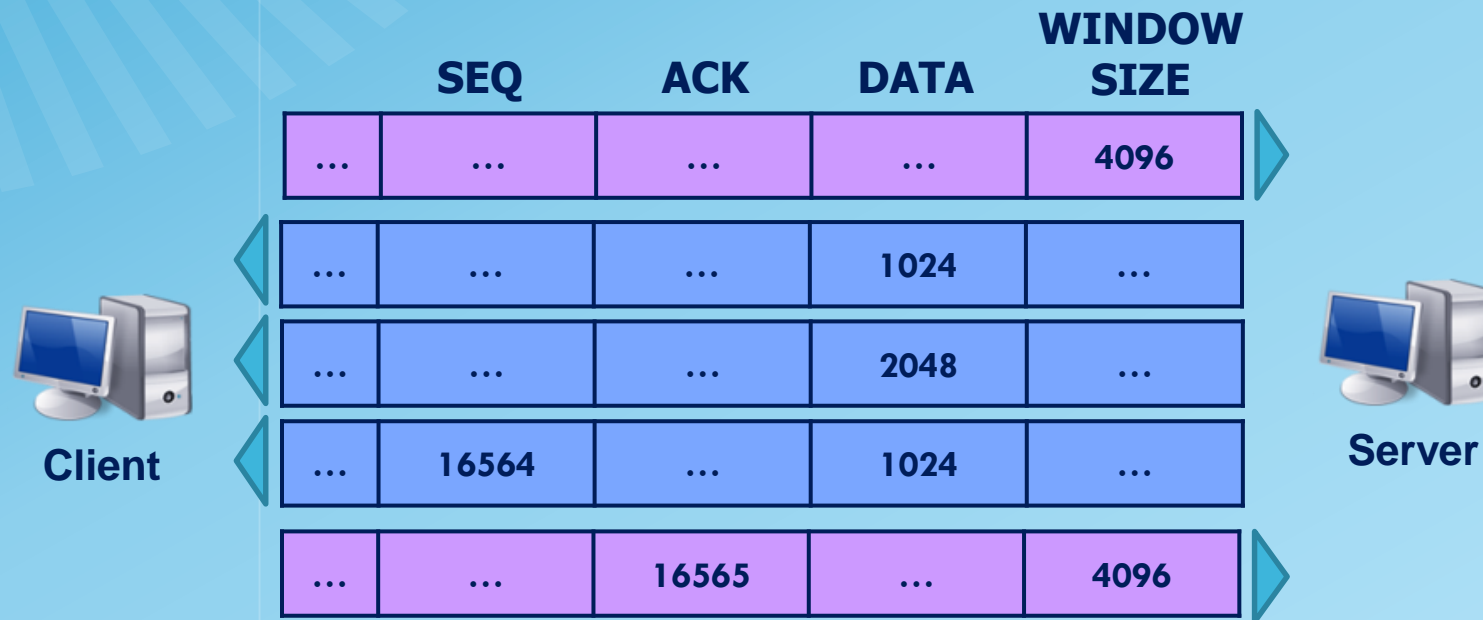
Квитирование – АСК

- Поле АСК содержит последний порядковый номер успешно принятого байта, увеличенный на 1, то есть первый порядковый номер ожидаемого байта
- При безошибочной передаче $ACK = SEQ_{rcvd} + 1$



TCP окно

- **TCP окно** – алгоритм управления интенсивностью потока данных, основанный на изменении максимального количества данных, которое получатель готов принять и подтвердить одним ответным сегментом
- Размер окна всегда назначает получатель, исходя из статистики количества ошибок

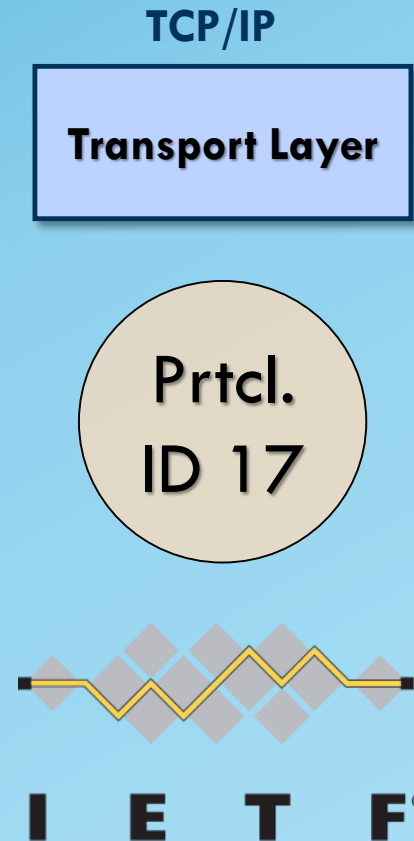


UDP – User datagram protocol

□ Особенности:

- отсутствие сессий
- ненадёжная доставка «best effort»
- отсутствие упорядочивания сегментов (дейтаграмм)
- отсутствие контроля за скоростью передачи
- максимально возможная скорость передачи данных за счёт всего вышеперечисленного

□ RFC 768



UDP дейтаграмма

	0-15	16-31
0	Source port (SP) – номер порта отправителя	Destination port (SP) – номер порта получателя
32	Length – полная длина дейтаграммы в байтах (до 65535)	Checksum – контрольная сумма
64	Data – данные верхнего уровня	