Async

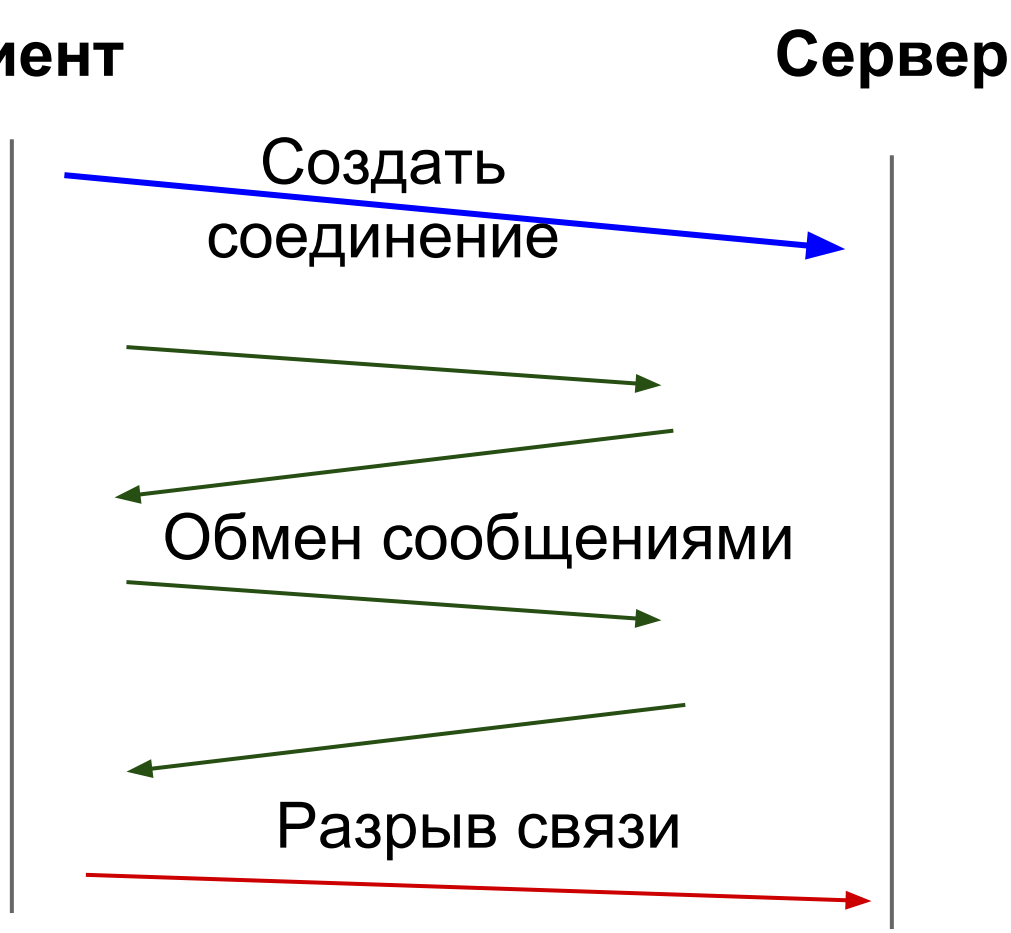
Урок 1

Сети = сокеты

- IPv4, IPv6, Unix, RAW и т.д.
- Для IPv4:
  - потоковые (TCP)
  - пакетные (UDP)

TCP, клиент-сервер

Transmission Control Protocol



Синхронные TCP сокеты

- Простые для программирования
- Легкие для изучения
- Практически непригодные для серверной стороны

TCP Сокет

Двунаправленный **байтовый** канал взаимодействия

+

средства управления этим каналом

Создание соединения

```
import socket

s = socket.socket(socket.AF_INET,
    socket.SOCK_STREAM)

s.connect(('127.0.0.1', 7777))
```

Передача данных

```
sent_size = s.send(b'binary data')
s.sendall(b'data again')

received_data = s.recv(4096)
if received_data:
    process(received_data)
else:
    do_shutdown(s)
```

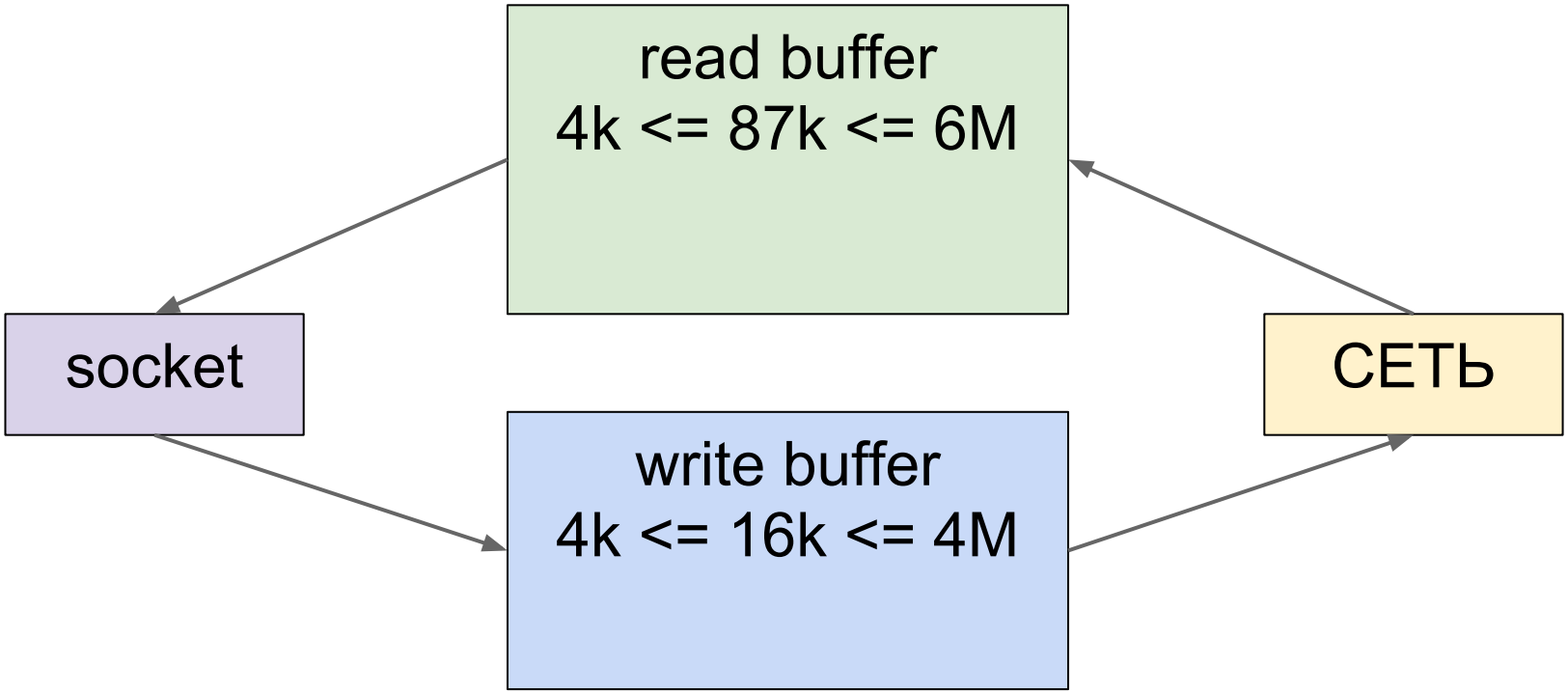
Прием соединения

```
s = socket.socket(socket.AF_INET,
    socket.SOCK_STREAM)

s.bind(('127.0.0.1', 7777))
s.listen(10)
client, addr = s.accept()

data = client.recv(4096)
```

Завершение соединения

```
s.close()


s.sendall(b'closing socket')
s.shutdown(socket.SHUT_WR)
data = s.recv(4096)
if data:
    process_final_answer(data)
s.close()
```

Буфера



Настройки

```
opt = s.getsockopt(level, name)
s.setsockopt(level, name, opt)
```

Десятки платформозависимых настроек соединения

Используются относительно редко

В основном это управление буферами, `TCP_CORK`, `TCP_NODELAY`

TIME_WAIT

После закрытия сокета его адрес освобождается **закрывающей стороной** через 1-4 минуты

```
s.setsockopt(SOL_SOCKET,
    SO_REUSEADDR, 1)
```

На маках рекомендуют еще и

```
s.setsockopt(SOL_SOCKET,
    SO_REUSEPORT, 1)
```

Запись после закрытия

```
l_onoff = 1
l_linger = 0
s.setsockopt(socket.SOL_SOCKET,
    socket.SO_LINGER,
    struct.pack('ii', l_onoff,
                l_linger))
s.close()
```

Обрыв соединения

TCP ***шлёт пакеты*** только если ***есть новые данные*** или ***нужно подвердить доставку***

~~SO_KEEPALIVE~~

Пользовательский keep alive

Ошибки

- Исключение `OSError` (начиная с Python 3.3)
- Причины
  - Адрес не найден
  - Порт не обслуживается
  - Таймаут
  - Разрыв соединения
  - Прочее

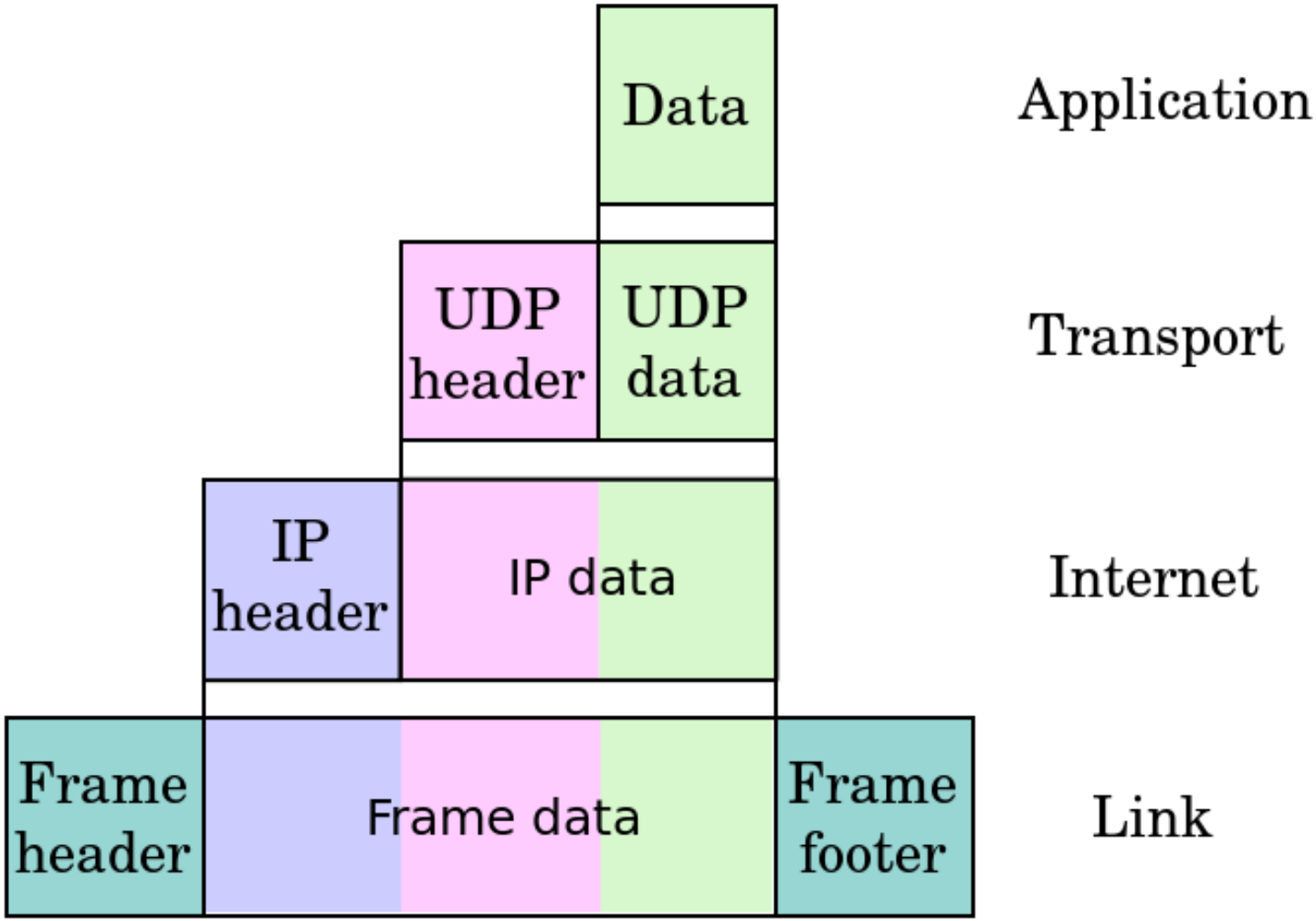
Пакеты

Пользовательские кадры сообщений

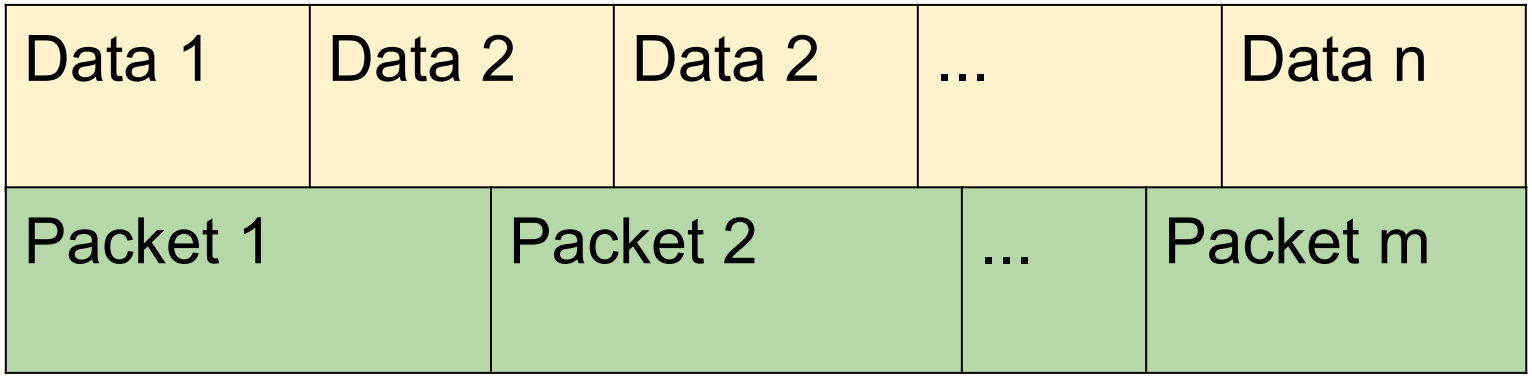
- длина
- тип
- данные
- контрольная сумма

Пара сокетов обменивается **бесконечными** потоками **байтов**, которые порциями проталкиваются через сеть

Под капотом


Data
Application
UDP header
UDP data
Transport
IP header
IP data
Internet
Frame header
Frame data
Frame footer
Link

TCP поток



Минимальный пакет по стандарту 512 байт
На практике минимум 4 килобайта

Несколько сокетов за раз

```
import select

s1.setblocking(False)
s2.setblocking(False)

reads, writes, errs = select.select(
    [s1, s2], [s2], [], 1.0)
for s in reads:
    data = s.recv(4096)
```

Задание

Две программы

- сервер
- клиент

Обмен сообщениями

- подключение
- посылка строки
- эхо-ответ
- отключение

Обработка исключений, проверка пакетов

Юниттесты

Следующий урок

Создание event loop
Обработка запускаемых по таймеру событий