Архитектура распределенного web приложения. Sockets

(с прошлой лекции)

Введение

- Модель OSI
- Сокеты
- Адресация, IP, port, DNS

Клиент сервер

- Основные этапы работы сервера
- Основные этапы работы клиента
- Обработка нескольких клиентов одновременно –потоки
- Ускорение обработки – пул потоков
- Ускорение запросов – постоянное соединение
- Команды с сервера – очередь задач на отправку

Протокол взаимодействия клиента и сервера

- Сегментирование данных, кадр данных
- Необходимость наличия протокола – языка взаимодействия
- Виды протоколов – текстовые, бинарные, примеры HTTP
- Сообщение в протоколе – заголовок, тело
- Определение ошибок протокола – обработка ошибок

Надежность соединения

- Интернет ненадежен
- Восстановление соединения – задача клиента

Пример простого клиент серверного приложения на C#

Сетевая модель OSI

1. Разделение зон ответственности
2. Простота
3. Надежность

Модель OSI				
Уровень (layer)		Тип данных (PDU[1])	Функции	Примеры
Host layers	7. Прикладной (application)	Данные	Доступ к сетевым службам	HTTP, FTP, SMTP
	6. Представительский (представления) (presentation)		Представление и шифрование данных	ASCII, EBCDIC, JPEG
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP
	4. Транспортный (transport)	Сегменты (segment)/ Дейтаграммы (datagram)	Прямая связь между конечными пунктами и надежность	TCP, UDP, SCTP
Media[2] layers	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk
	2. Канальный (data link)	Биты (bit)/ Кадры (frame)	Физическая адресация	PPP, IEEE 802.2, Ethernet, DSL, ARP, L2TP
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, витая пара, коаксиальный кабель, оптический кабель

Сокет

- Сервер – поставщик услуг
- Клиент - заказчик услуг
- Сокет - канал взаимодействия между клиентом и сервером

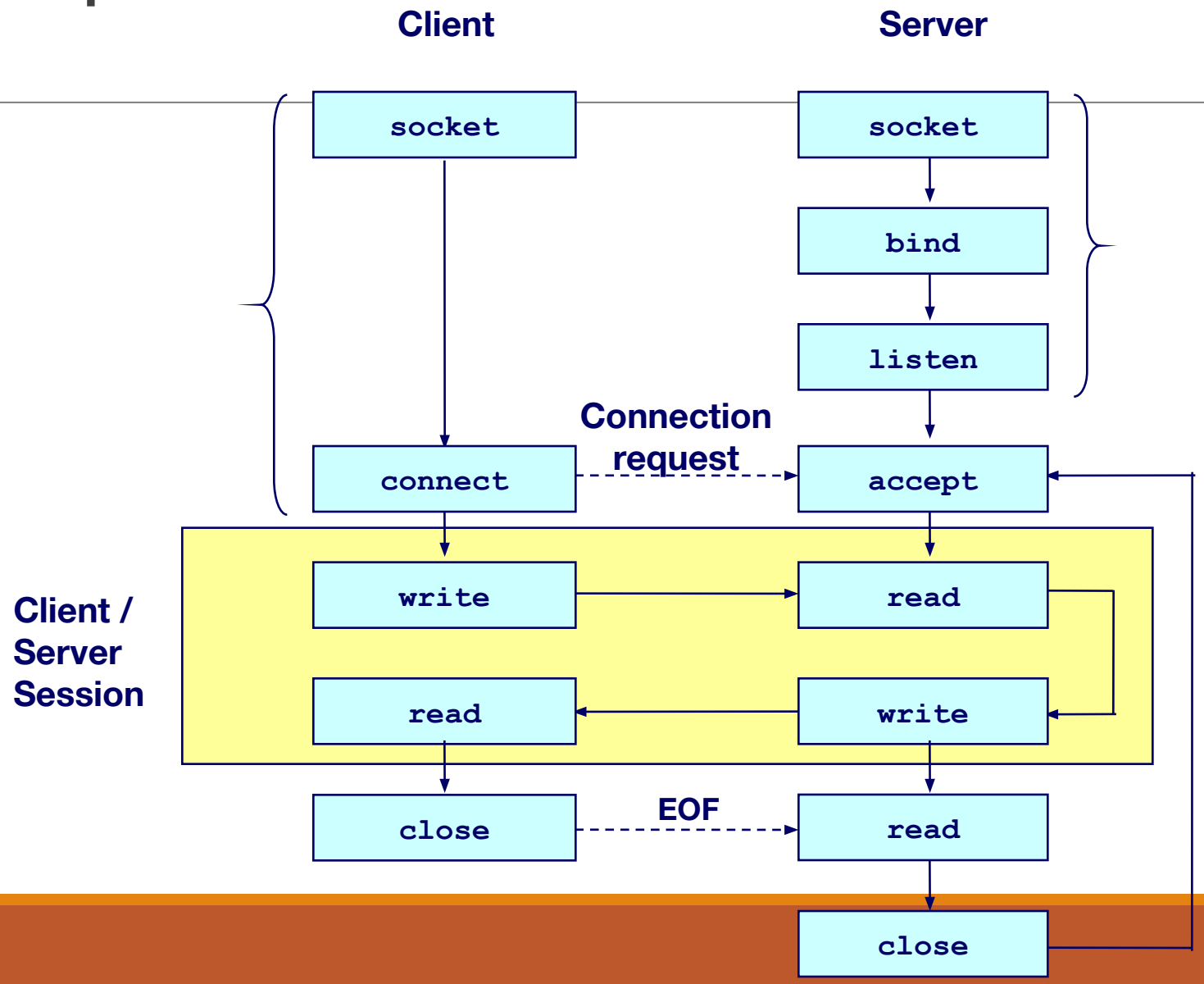
Адресация: IP, DNS

- Сервер имеет IP адрес
- DNS - *Domain Name System*
  - *www.travelline.ru – домен третьего уровня*
  - *travelline.ru – домен второго уровня*
- Привязка IP адреса к доменному имени
- Одно имя несколько адресов
  - Name: google.com
  - Addresses: 173.194.44.64, 173.194.44.70, 173.194.44.71, 173.194.44.68, 173.194.44.72, 173.194.44.69, 173.194.44.67, 173.194.44.66, 173.194.44.65, 173.194.44.78, 173.194.44.73

Адресация: Port

- Сервис (то что мы обычно называем сервером) – определяется числом - **номер порта**
  - *общеизвестные* (или *системные*, 0—1023)
  - *зарегистрированные* (или *пользовательские*, 1024—49151)
  - *динамические* (или *частные*, 49152—65535).

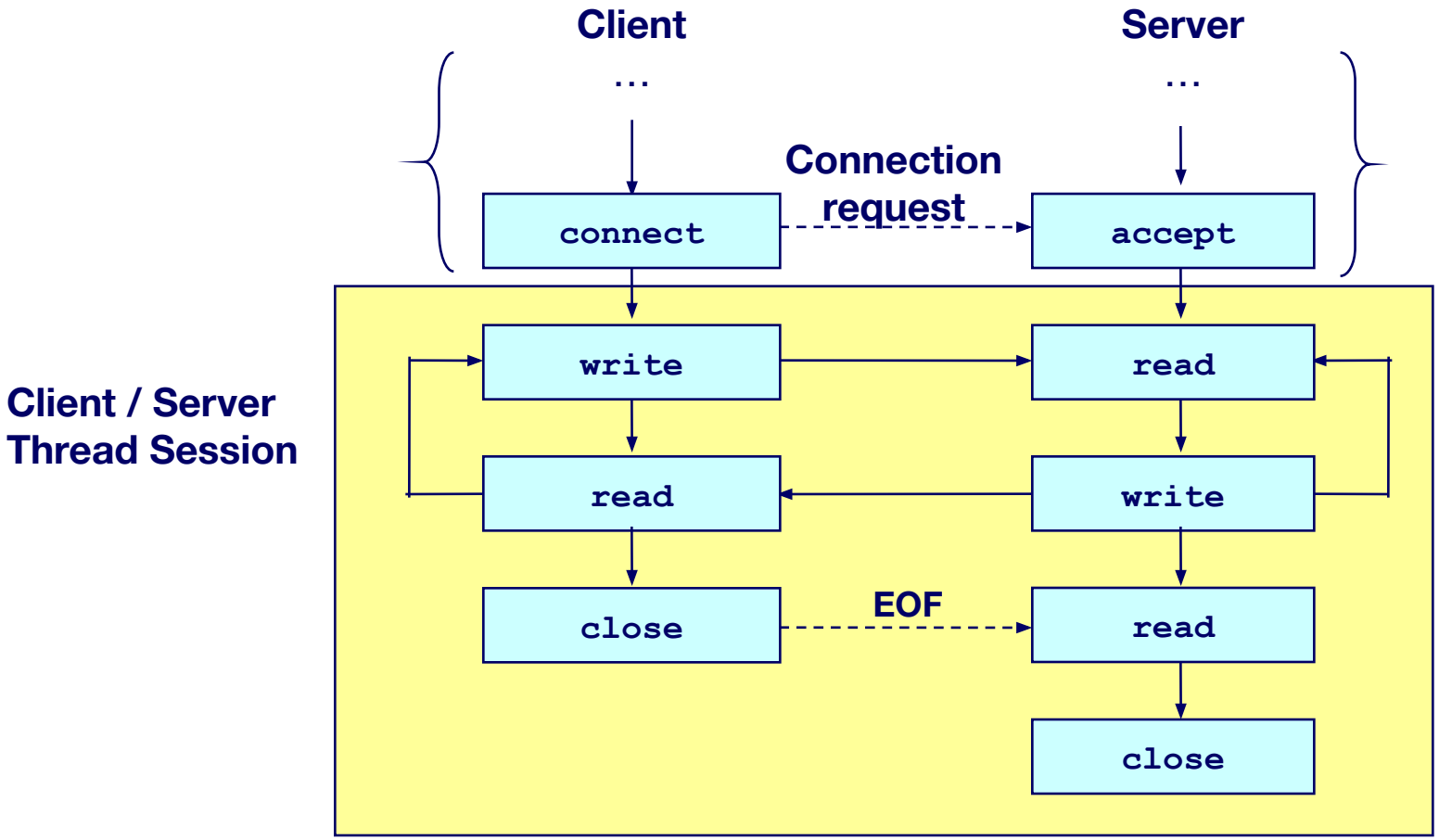
Общая схема


Client
Server
socket
socket
bind
listen
Connection request
connect
accept
Client / Server Session
write
read
read
write
close
EOF
read
close

Обслуживание нескольких клиентов

- Обработка в потоке
- Пул потоков
  - поддерживается серверами «из коробки»
  - имеет ограничения (IIS, application)

Постоянное соединение



TCP фрагментирование

- TCP не гарантирует, что сообщение придет сразу и целиком
  - Одна запись “Hello\n”
  - Множественное чтение
    - “He”
    - “llo\n”
- Нужен разделитель
  - IRC команды разделяются переносом строки
- Буферизированное чтение

Свой протокол взаимодействия

- Языком общения между клиентов и сервером служит протокол
  - FTP - файлы
  - HTTP - интернет
  - RTP – медиа данные
- TCP – гарантирует целостность ваших сообщений
- Заголовок сообщения
  - Код сервера
  - Идентификатор сообщения
  - Длина сообщения
  - Тело сообщения
  - Контрольная сумма

Надежность соединения

- TCP - гарантия доставки и целостности данных
- Интернет ненадежен
- Пользователь не должен видеть, что соединение рвется
- Повторение запросов в случае разрыва
  - Идемпотентность запросов

Современные веб приложения

- Клиент
  - браузер, игровой движок, торрент
- Балансировщик
  - NGINX, HaProxy
- Фронтэнд
  - сервер со статикой
- Бэкенд
  - сервер с логикой
- База данных
  - хранилище данных