

ТСР/ІР:

ПРАГМАТИЧЕСКОЕ ВВЕДЕНИЕ

ЗДРАСЬТЕ

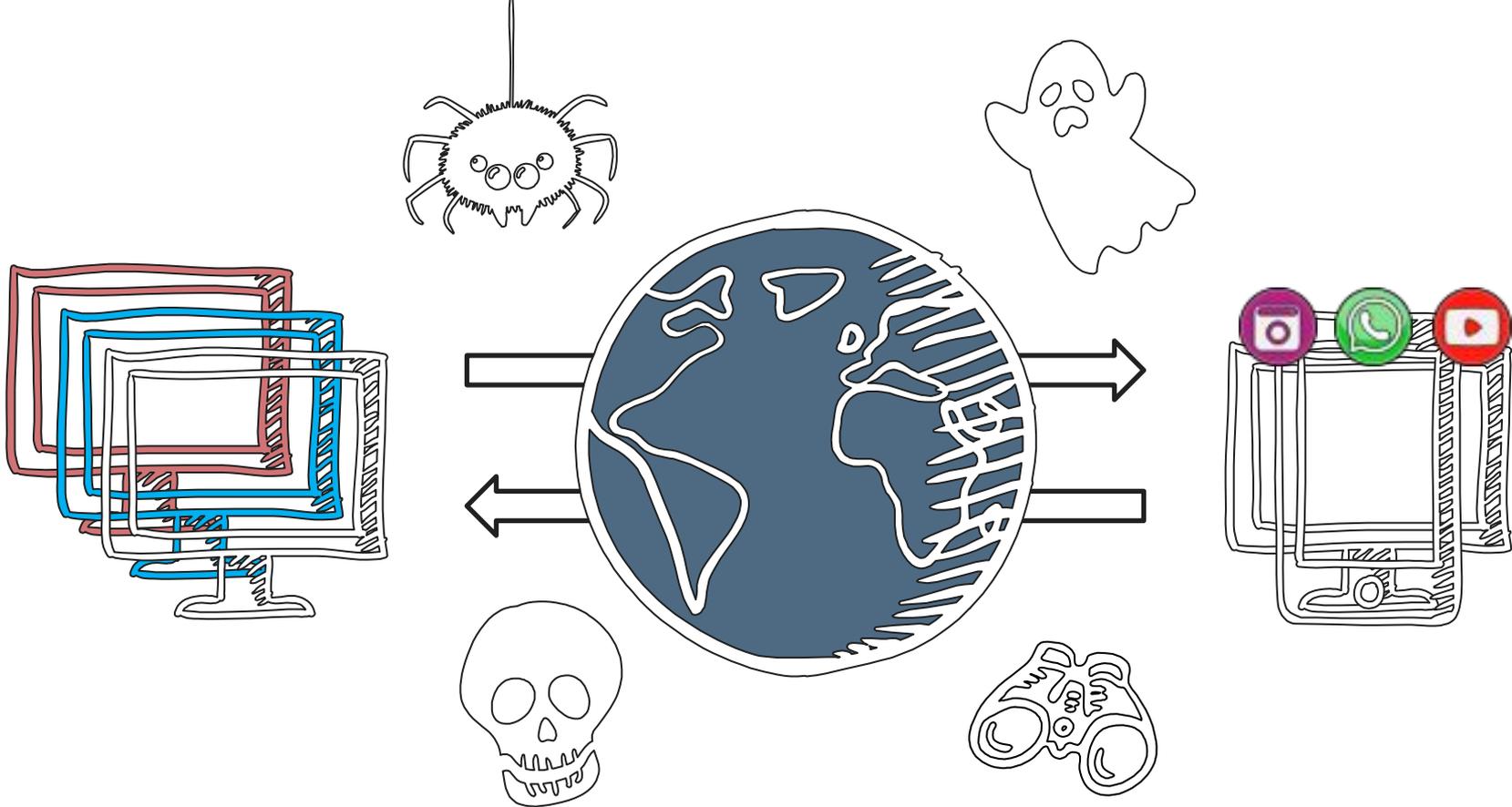
- ❖ Владимир Плизга́
- ❖ В разработке ПО с 2011 г
 - ФинТех (интернет-банк)
 - Промышленный IoT
- ❖ Люблю помогать людям
(особенно разработчикам)



INTRODUCTION

ОСНОВЫ ОСНОВ

В ЧЕМ, СОБСТВЕННО, СЛОЖНОСТЬ?



СЛОЖНОСТИ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ

- ❖ **Разнообразие** устройств
- ❖ **Многозадачность** каждого устройства
- ❖ **Непредсказуемость** среды передачи
- ❖ **Угрозы** безопасности
- ❖ *(впишите своё)*

ДЕЛЕНИЕ НА УРОВНИ
ХОРОШИЙ ВИД ДЕКОМПОЗИЦИИ



OPEN SYSTEM INTERCONNECTION

Справочная модель сетевого
взаимодействия

OSI MODEL

- ❖ Создавалась **7 лет**: с 1977 по 1984 годы
- ❖ Описывает **7 уровней** взаимодействия
- ❖ Не содержит описаний **реализации** протоколов
- ❖ Каждый уровень добавляет **заголовки**

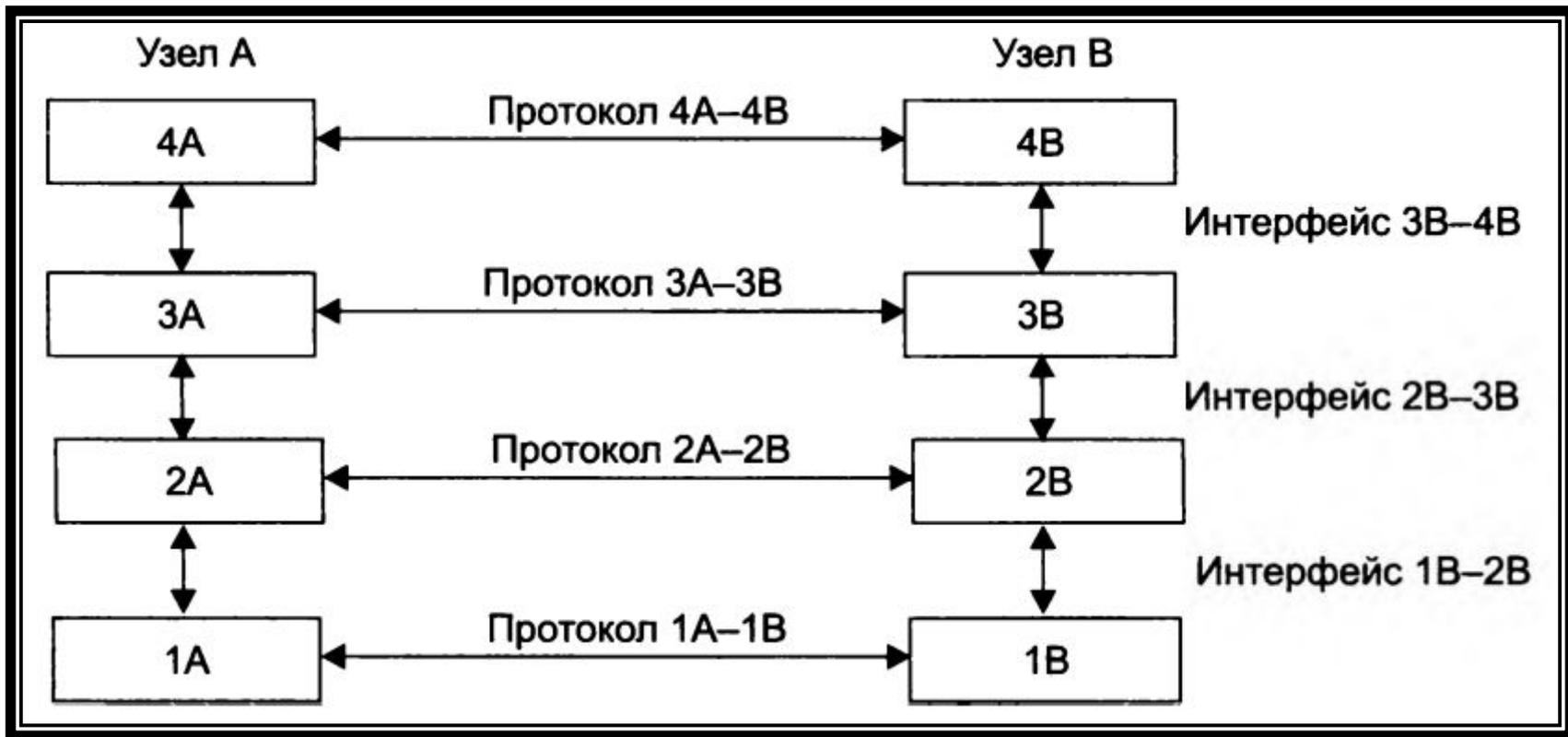
7 LEVELS OF OSI MODEL

1. Прикладной
2. Представления
3. Сеансовый
4. Транспортный
5. Сетевой
6. Канальный
7. Физический



ТЕРМИНОЛОГИЯ OSI

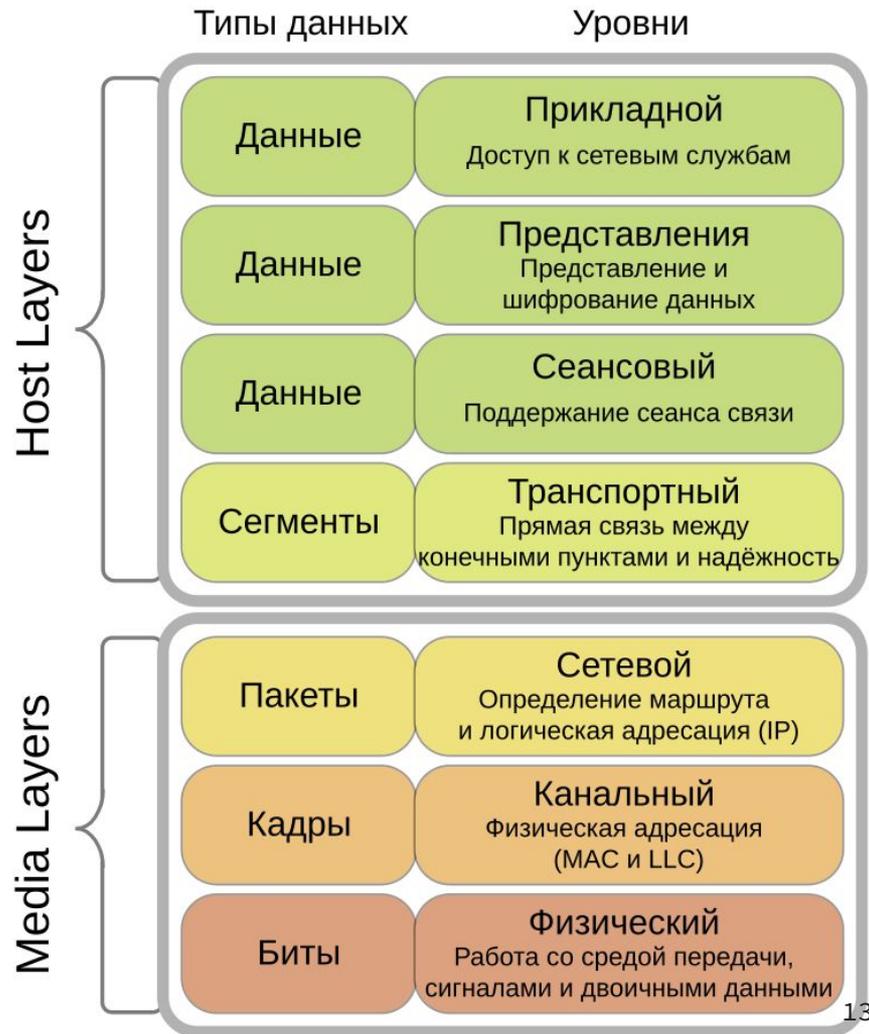
- ❖ **Уровень** – единица декомпозиции сетевого стека
- ❖ **Протокол** – язык общения между одинаковыми уровнями на **одном** этаже
- ❖ **Интерфейс** – язык общения между разными уровнями на **соседних** этажах
- ❖ **Операнд** – единица данных, с которой работает каждый уровень



РАЗНИЦА МЕЖДУ ИНТЕРФЕЙСОМ И ПРОТОКОЛОМ

ОПЕРАНДЫ УРОВНЕЙ

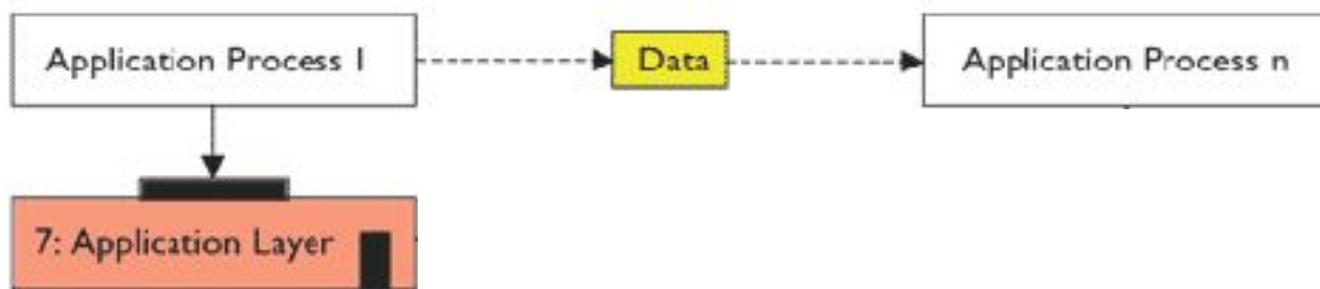
- ❖ Типы данных \leftrightarrow операнды
- ❖ Данные \leftrightarrow сообщения
- ❖ Операнд \leftrightarrow PDU
(Protocol Data Unit)



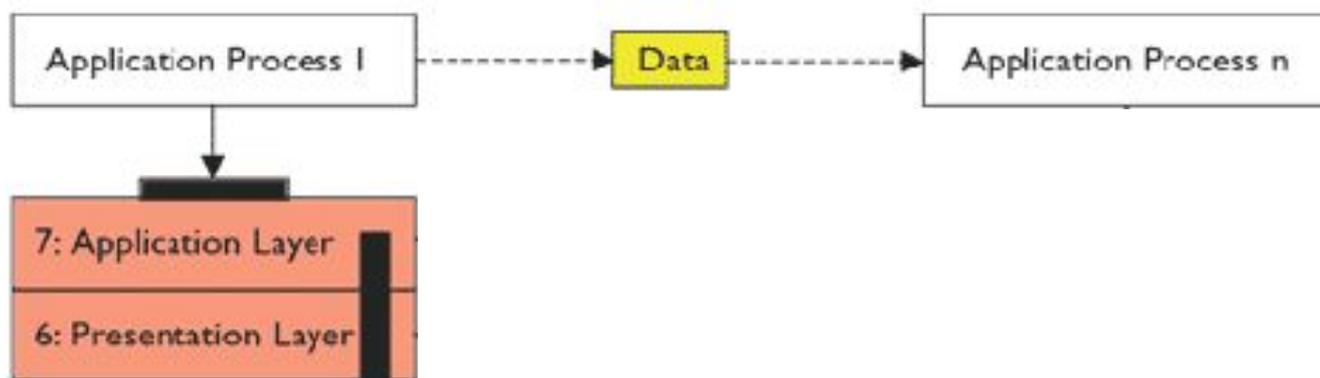


ПЕРЕДАЧА
ДАННЫХ
В OSI

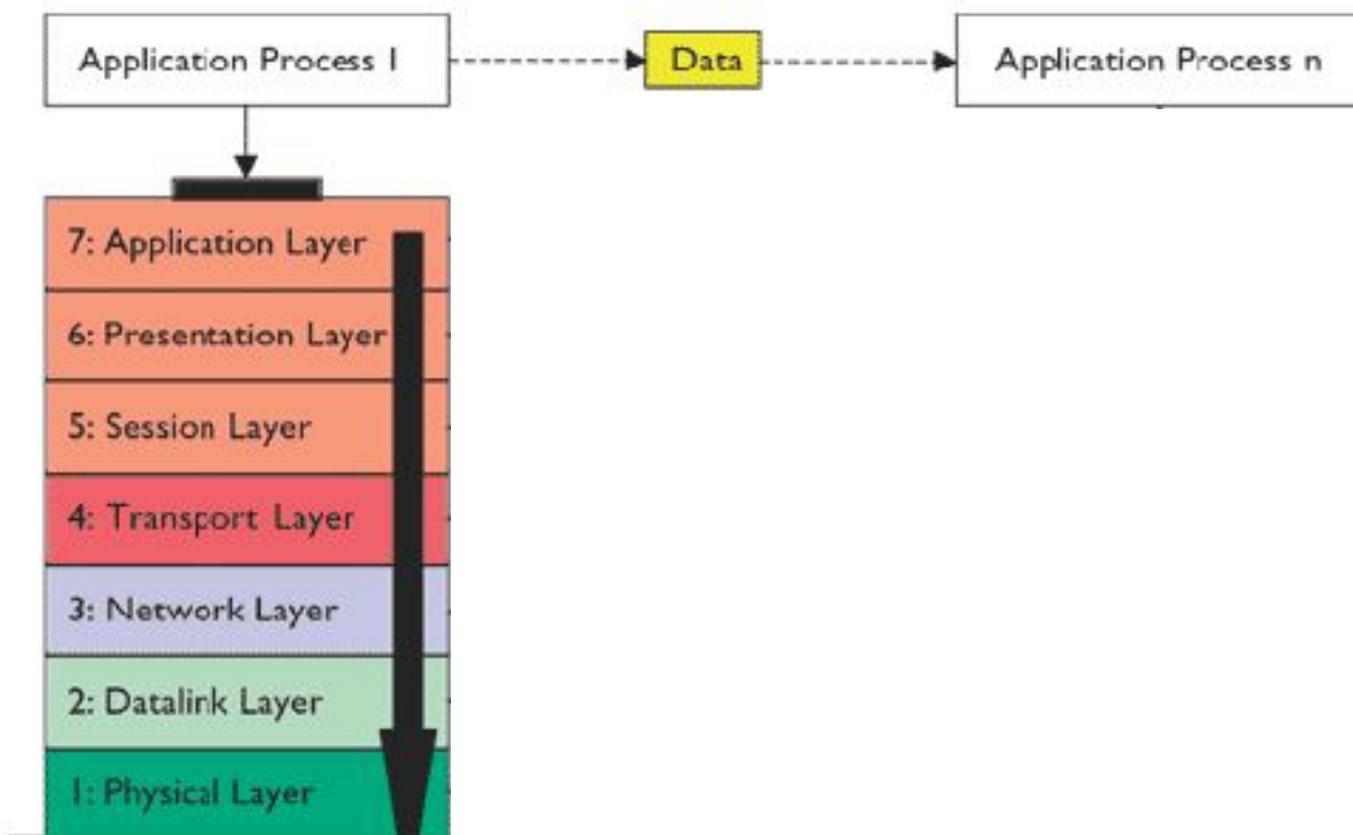
ПЕРЕДАЧА ДАННЫХ В OSI



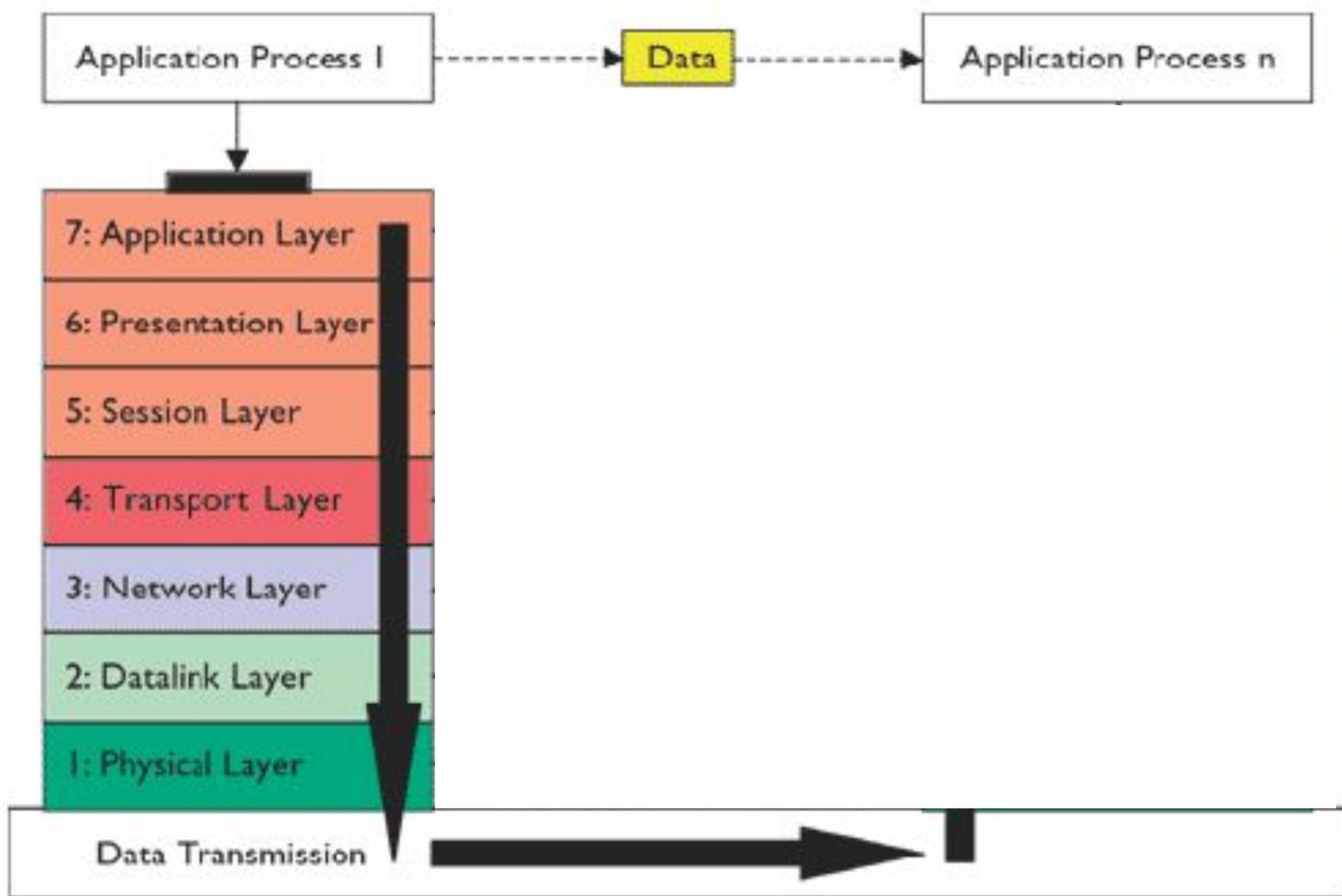
ПЕРЕДАЧА ДАННЫХ В OSI



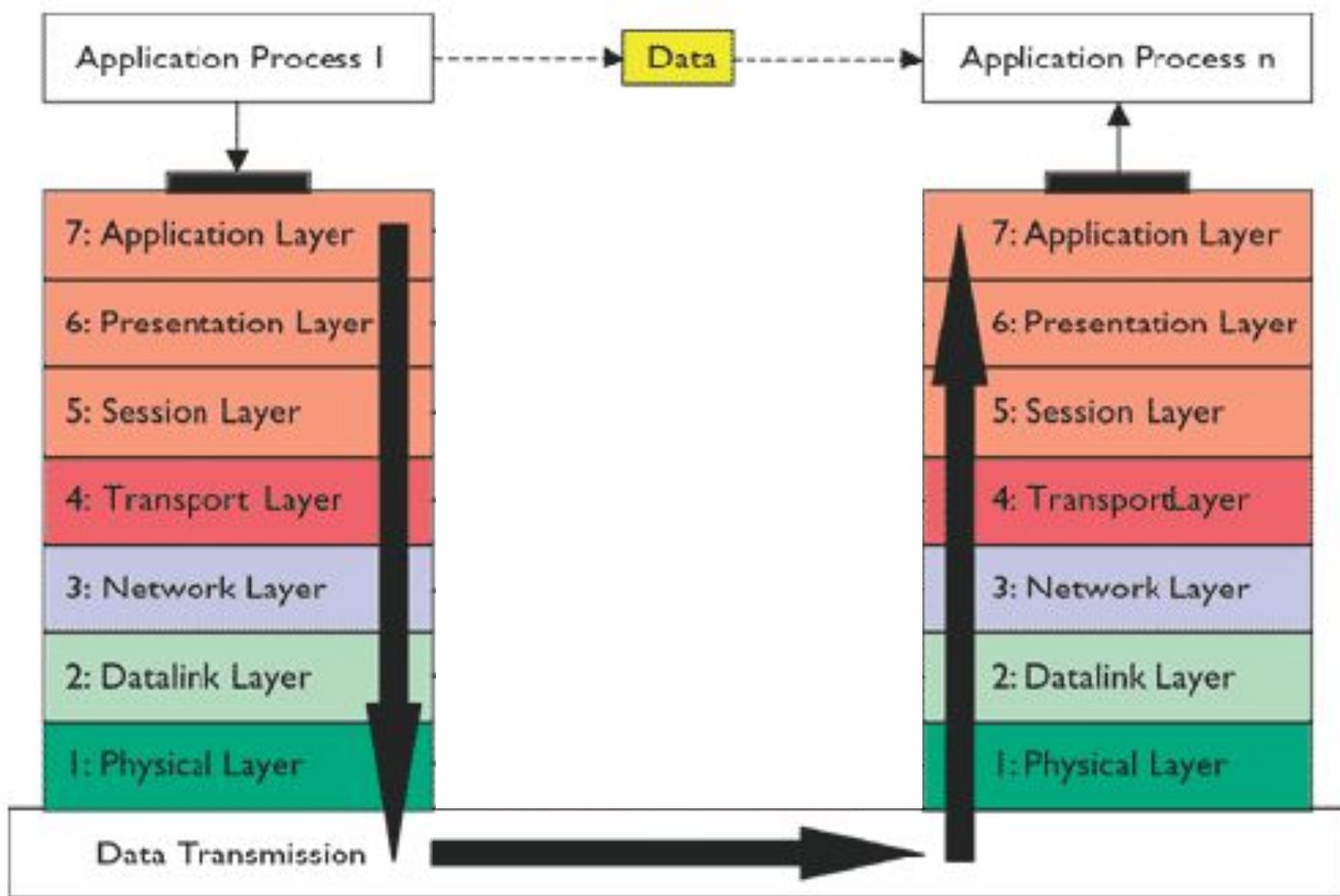
ПЕРЕДАЧА
ДАНЫХ
В OSI



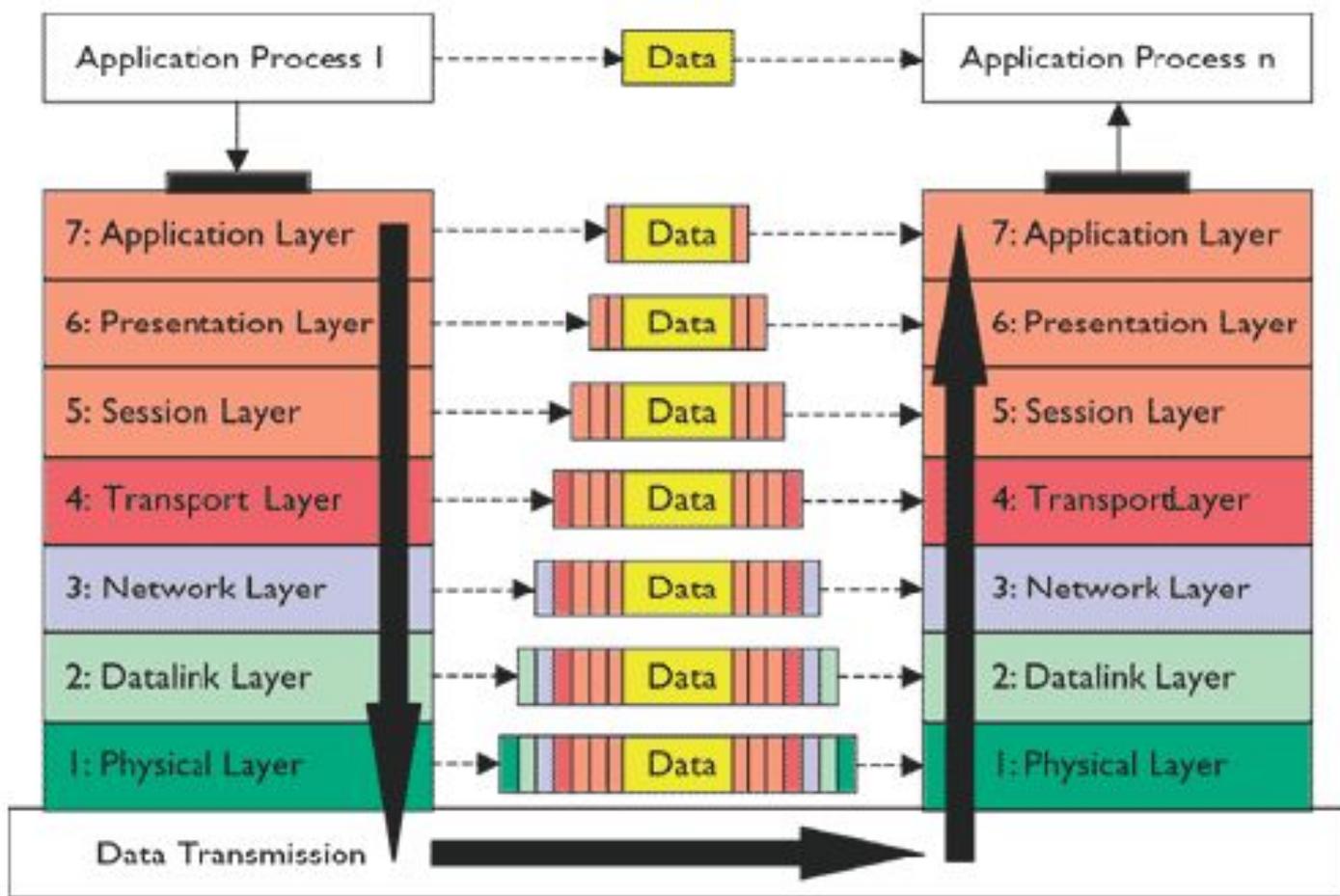
ПЕРЕДАЧА
ДАНЫХ
В OSI



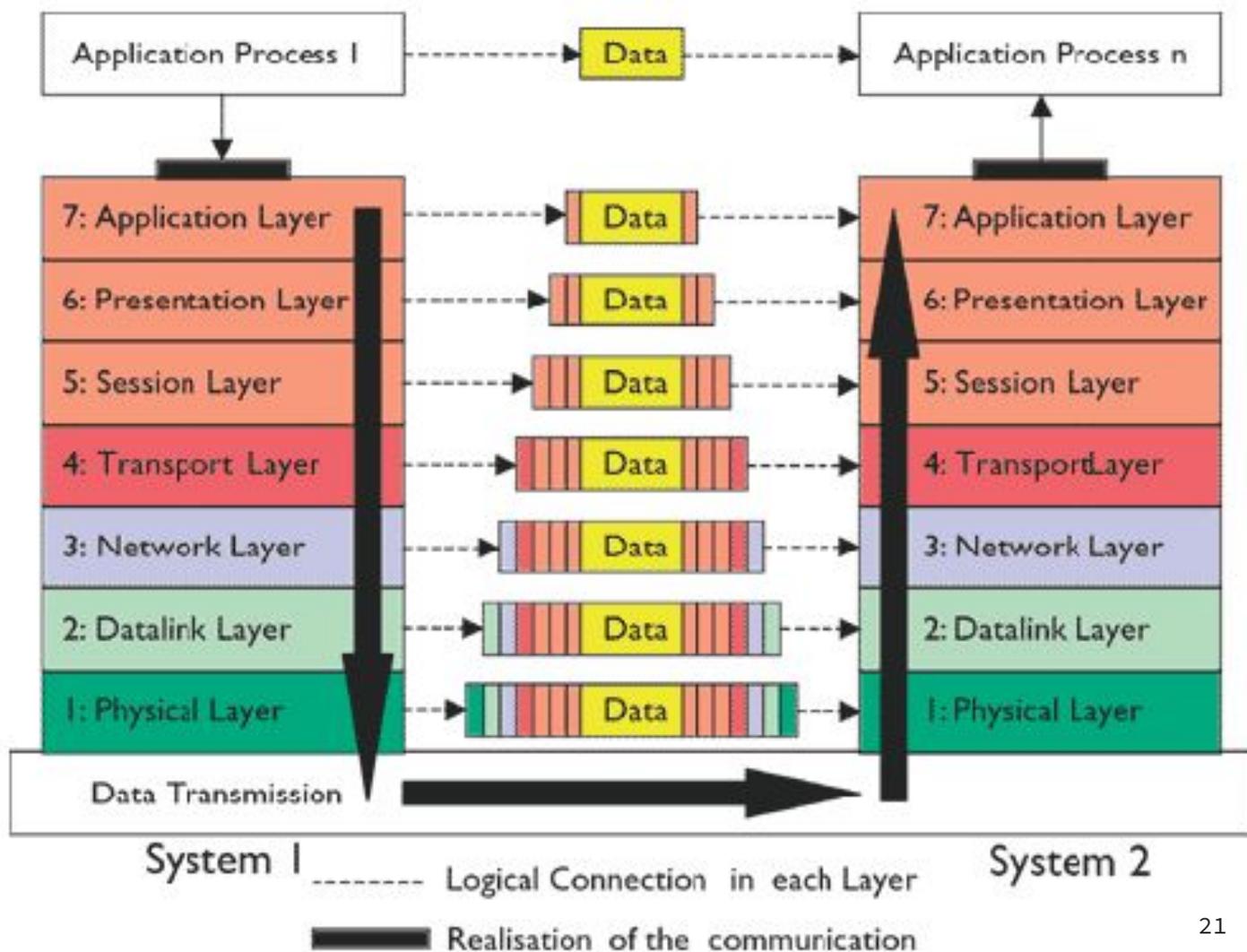
ПЕРЕДАЧА
ДАНЫХ
В OSI



ПЕРЕДАЧА
ДАНЫХ
В OSI



ПЕРЕДАЧА
ДАНЫХ
В OSI





“

Пока комитеты *ISO* спорили
о своих стандартах,
за их спиной менялась
вся концепция организации сетей
и по всему миру внедрялся
протокол **TCP/IP**.

[Evi Nemeth](#)

TCP/IP

Стек протоколов здорового человека

TCP/IP: 45+ YEARS OF SCALABILITY

- ❖ Самый распространенный сетевой стек на Земле
- ❖ Включает только 4 уровня:
 - ❖ Прикладной
 - ❖ Транспортный
 - ❖ Сетевой
 - ❖ Физический (канальный)



Не регламентирует
конкретные
протоколы

OSI



TCP/IP



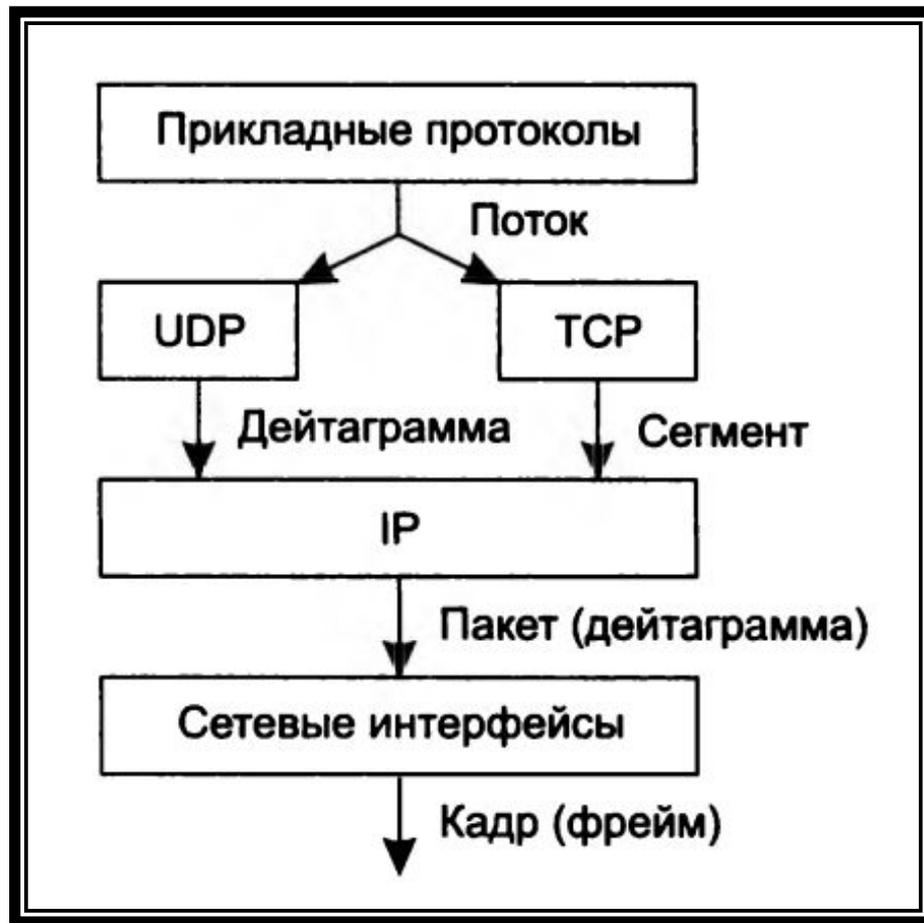
OSI **VS** TCP/IP

Соотношение уровней

(⚠ примерное)

ОСНОВНЫЕ ПРОТОКОЛЫ И ОПЕРАНДЫ* TCP/IP

*операнд \Leftrightarrow PDU:
Protocol Data Unit





15	1 000 000
14	500 000
13	250 000
12	125 000
11	64 000
10	32 000

К какому уровню стека TCP/IP относятся Wi-Fi и Bluetooth?

А Прикладной

С Транспортный

В Канальный

Д 80-ый





15	1 000 000
14	500 000
13	250 000
12	125 000
11	64 000
10	32 000

К какому уровню стека TCP/IP относятся Wi-Fi и Bluetooth?

А Прикладной

С Транспортный

В Канальный

Д 80-ый



WI-FI

- ❖ Не протокол, а технология
- ❖ Основана на наборе спецификаций IEEE **802.11x**
- ❖ По модели OSI находится на **физическом** уровне
- ❖ По модели TCP-/IP находится на **канальном** уровне

BLUETOOTH

- ❖ Не протокол, а технология
- ❖ Основана на спецификации IEEE **802.15.1**
- ❖ По модели OSI находится на **физическом** уровне
- ❖ По модели TCP-/IP находится на **канальном** уровне

WI-FI

- ❖ Не протокол, а технология
- ❖ Основана на наборе спецификаций IEEE **802.11x**
- ❖ По модели OSI находится на **физическом** уровне
- ❖ По модели TCP-/IP находится на **канальном** уровне

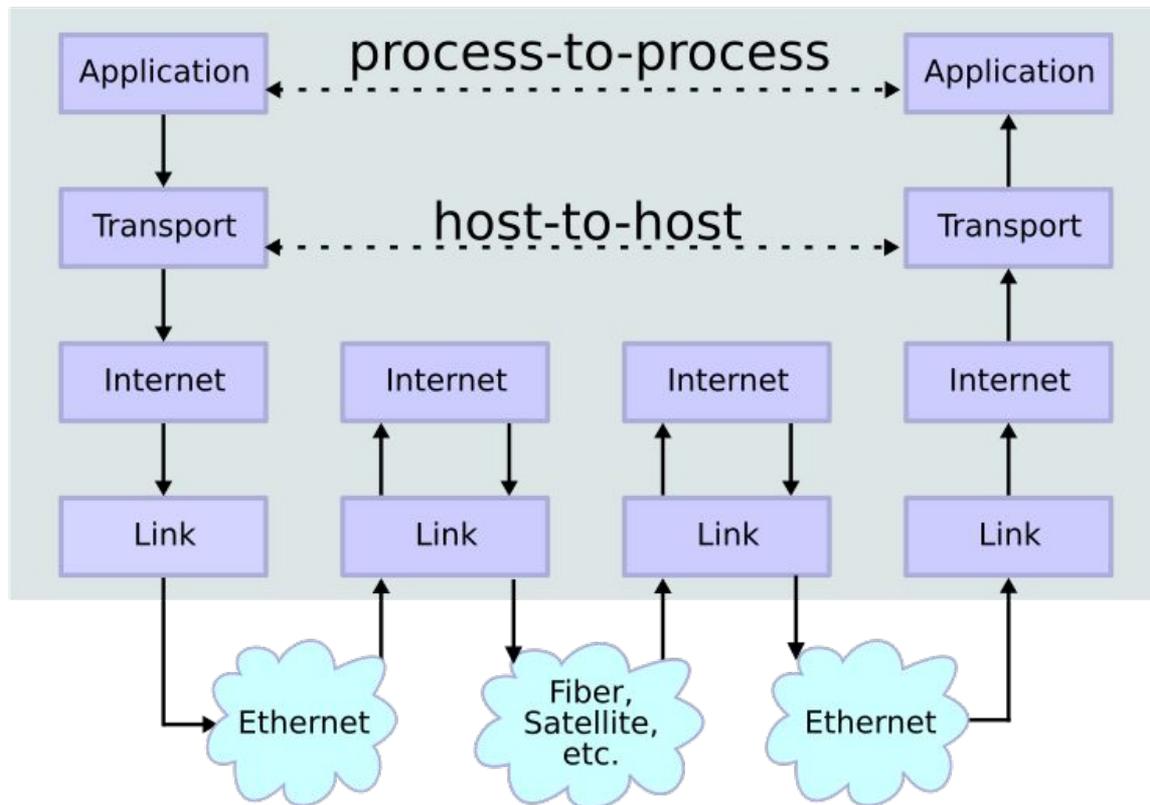
BLUETOOTH



Харальд I Синезубый

ПЕРЕДАЧА
ДАННЫХ
В TCP/IP

Data Flow



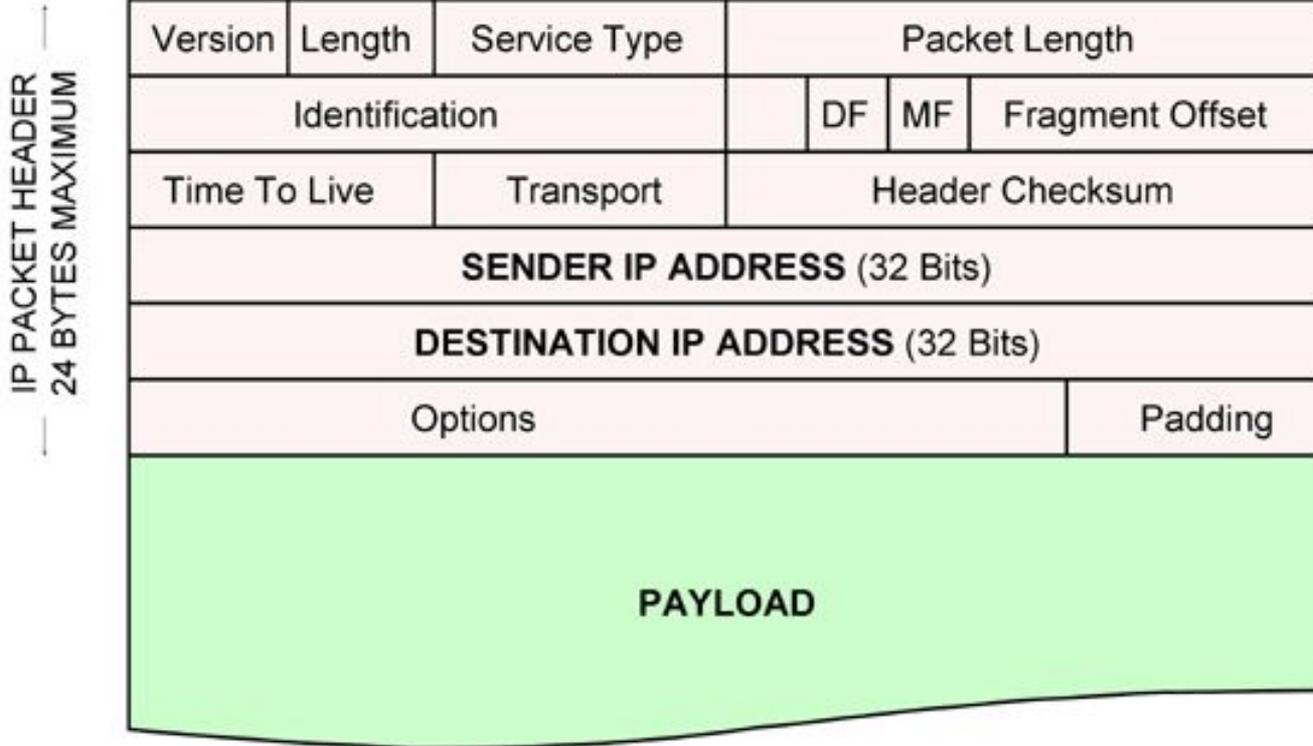
INTERNET PROTOCOL (IP)

Основной житель **сетевого** уровня TCP/IP



“Система адресации,
не зависящая от
способов адресации
узлов
в отдельных сетях”

PDU @ IP: PAKET



IP АДРЕС (192.168.0.1)

- ❖ В каждом IP-пакете **2 IP-адреса**: получателя и отправителя
- ❖ IP-адрес – адрес **сетевоего интерфейса**, а не всей машины
- ❖ В каждом IP-адресе **32 бита** (если это не IPv6)
- ❖ Разбиение на **4 байта условно** и делается для удобства
- ❖ Реально в адресе всего **2 части**: номер сети и номер узла
- ❖ Между частями **нет** универсальной границы

СПОСОБЫ РАЗБИЕНИЯ АДРЕСА НА НОМЕР СЕТИ И УЗЛА

❖ ~~Фиксированная~~ граница

❖ **Маска** подсети (применяется к адресу через AND):

1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0

❖ **Классы** подсетей (по первым битам адреса):

❖ Индивидуальные: A, B, C

❖ Групповые: D

❖ Резервированные: E

ОСОБЫЕ АДРЕСА IP

0.0.0.0

Алиас для **всех** сетевых адресов узла, но только при **прослушивании**.

255.255.255.255

Широковещательный адрес.
Соответствует отправке **всем узлам** сети.

127.x.x.x

Группа петлевых адресов.
Соответствует отправке **самому себе**.



15	1 000 000
14	500 000
13	250 000
12	125 000
11	64 000
10	32 000

Если компьютер достижим по сети,
сколько у него IP-адресов?

А Один

С Два или больше

В Два

Д "_(ツ)_/"





15	1 000 000
14	500 000
13	250 000
12	125 000
11	64 000
10	32 000

Если компьютер достижим по сети,
сколько у него IP-адресов?

А

Один

Два или больше

В

Два

Д

`_(ツ)_/`



СПОСОБЫ НАЗНАЧЕНИЯ СЕТЕВЫХ АДРЕСОВ

Вручную

Администратор в настройках каждого сетевого интерфейса **прописывает** его IP и другие параметры.

Автоматически

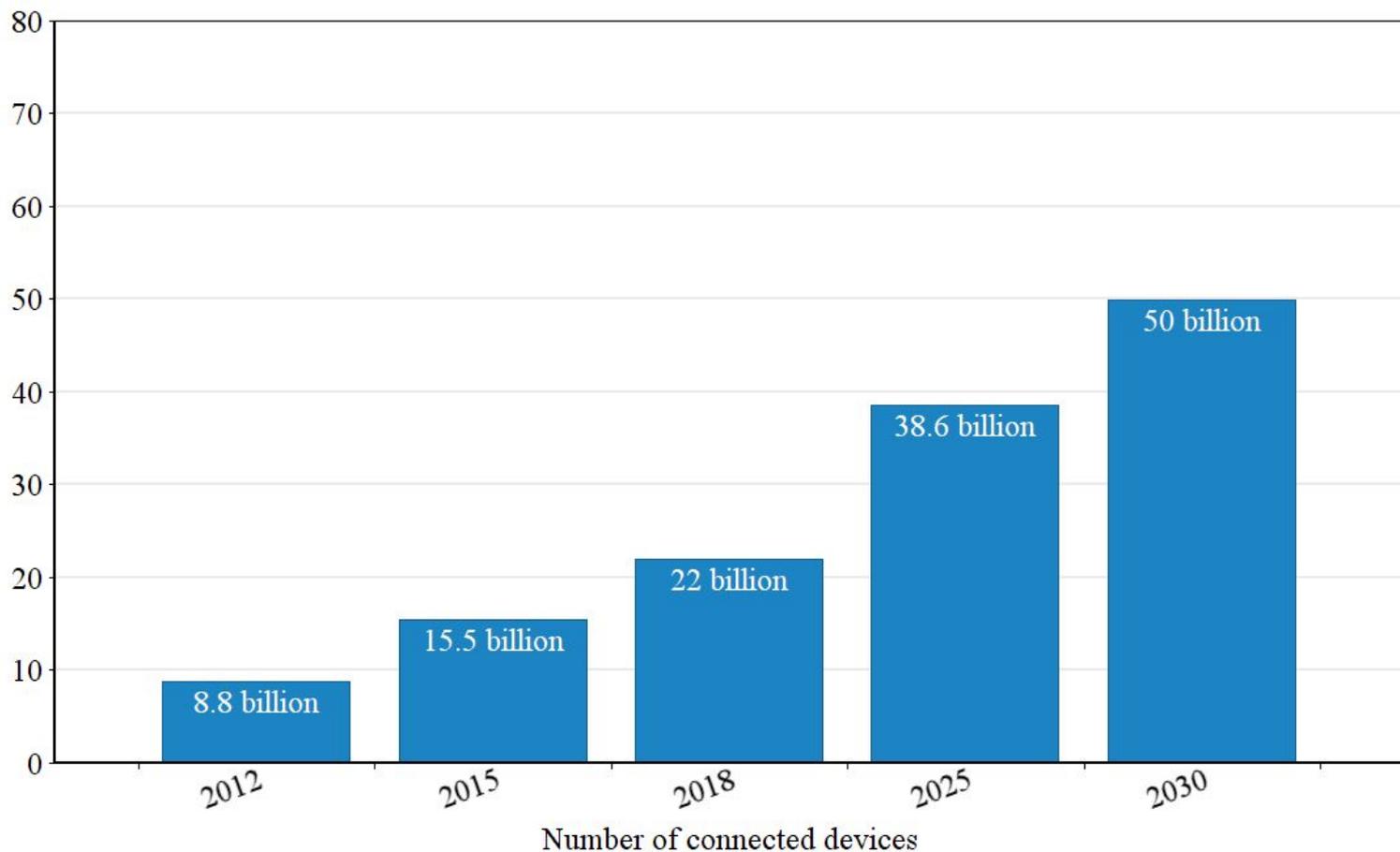
Сетевой интерфейс **сам** получает IP и другие параметры из сети по протоколу **DHCP**.

IP VERSION 6

От создателей IPv4

IPv6: ЗАЧЕМ?

IPv4
обеспечивает
лишь 4,2 млрд.
адресов



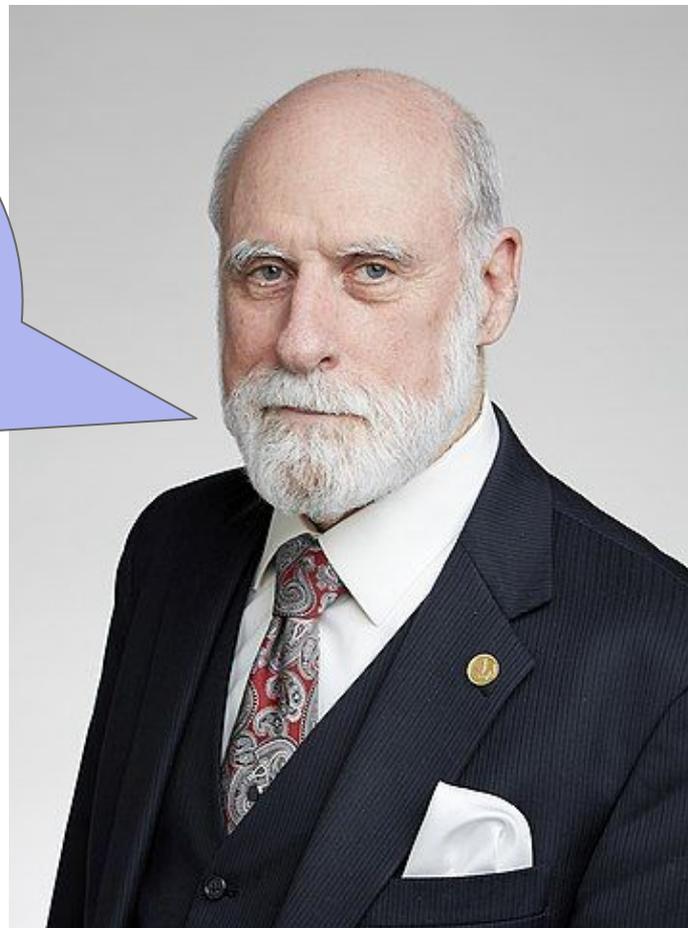
Number of connected devices

Source: statista.com

IPv6:
ЗАЧЕМ?

“I thought 32 bits ought to be enough for Internet addresses.”

Vint Cerf,
“отец Интернета”
(1973)



IPv6: ОСНОВНОЕ

- ❖ Решает проблему **нехватки** адресов
 - Избавляет от нужды использовать NAT (Network Address Translation)
 - Адрес IPv6 имеет длину 128 бит, а в IPv4 – 32
- ❖ Оптимизирует **маршрутизацию** пакетов
- ❖ Обеспечивает **безопасность** на своём уровне (IPSec)
- ❖ Упрощает (авто)**конфигурацию**

FE80::250:56FF:FE83:5BDE

Пример адреса в IPv6.
Мерзость, не правда ли?

127.0.0.1 <=> ::1

INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

Вспомогательный протокол сетевого
уровня TCP/IP



ICMP СООБЩАЕТ О ПРОБЛЕМАХ НА ПУТИ IP-ПАКЕТА

ping

Проверяет сетевую **доступность** узла путем отправки специального эхо-запроса (ICMP type 8) и получения эхо-ответа (ICMP type 0).

tracert/traceroute

Прокладывает **маршрут** к целевому узлу путем последовательного увеличения TTL (TimeToLive) сетевых пакетов.

ICMP ПРОТОКОЛ ЗАЧАСТУЮ ЗАПРЕЩЁН

- ❖ Наиболее известные атаки посредством ICMP:
 - ❖ Перенаправление трафика
 - ❖ Smurf
 - ❖ Ping Flood
- ❖ Поэтому проверять нужно через еще и через **telnet**



IP: РЕЗЮМЕ

- ❖ Система адресации, не зависящая от способов адресации узлов в отдельных сетях
- ❖ По модели OSI – L3
- ❖ Единица данных: пакет
- ❖ Есть версии IP(v4) и IPv6
 - Адреса IPv4 исчерпаны, но переиспользуются

TRANSMISSION CONTROL PROTOCOL (TCP)

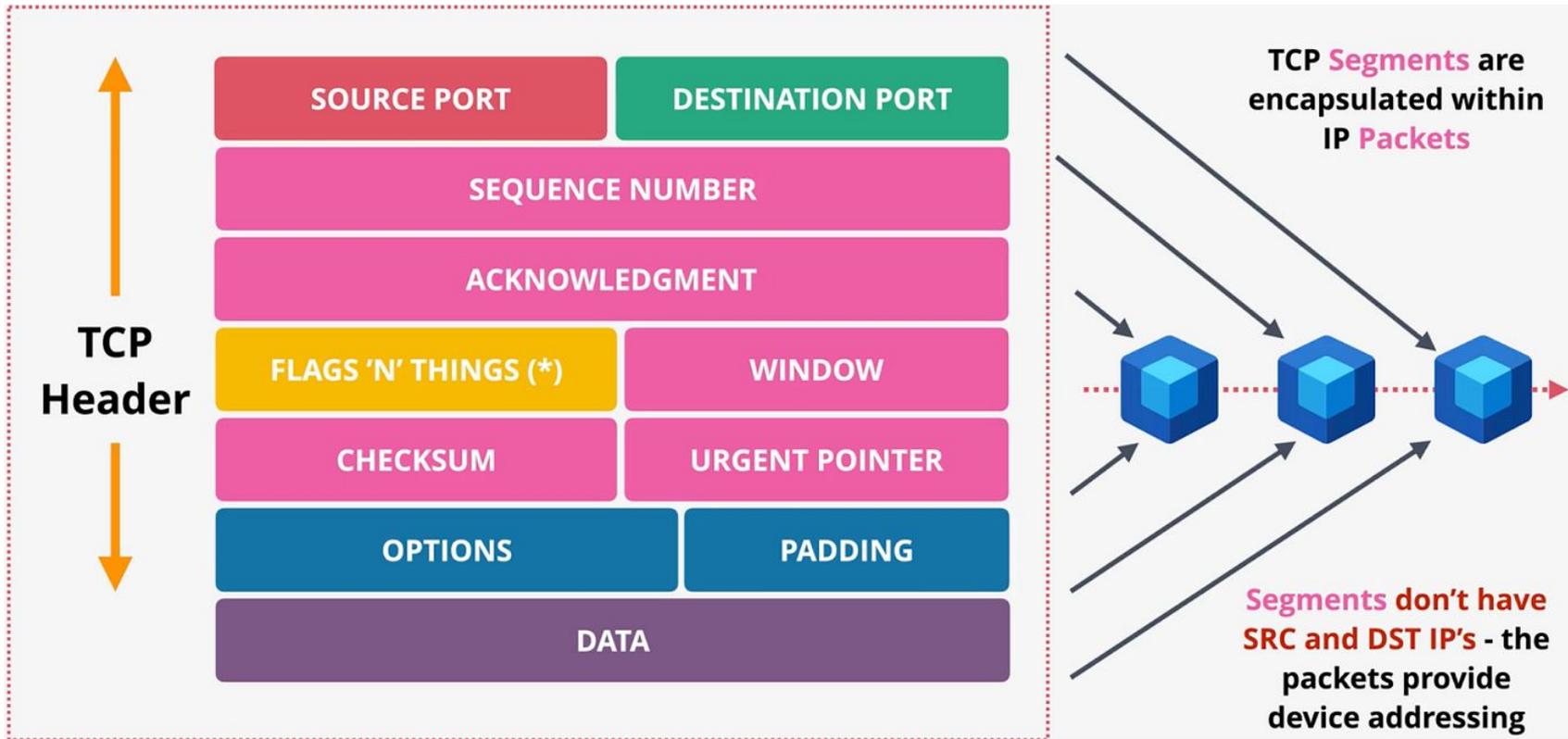
Основной житель транспортного
уровня TCP/IP



ТСР: ОСНОВНОЕ

- ❖ Реализует **соединения**
- ❖ Гарантирует **порядок** доставки
- ❖ Управляет **интенсивностью** потока
- ❖ Детектирует и корректирует **ошибки**
- ❖ Обеспечивает **двунаправленный** обмен
- ❖ Умеет **переотправлять** данные при потерях

PDU @ TCP: SEGMENT



ЧТО ТАКОЕ ПОРТ В ТСП?

- ❖ Способ мультиплексирования процессов на узле
 - ❖ Позволяет **разруливать** трафик разных процессов на одном хосте
- ❖ Может быть либо входящим, либо исходящим
- ❖ У ТСП и UDP **раздельные** множества портов

РЕЖИМЫ РАБОТЫ ПОРТОВ

Приём данных (прослушивание)

- ❖ Один порт – один процесс
- ❖ Удерживаются процессами
- ❖ Получать можно от многих сразу
- ❖ Распределение (условное):
 - 0–1023: системные
 - 1024–10000: прикладные
 - 10000–65535: динамические

Отправка данных (запись)

- ❖ Один порт – один процесс
- ❖ Не удерживаются процессами
- ❖ С одного порта можно отправлять на разные хосты
- ❖ Все исходящие порты распределяются динамически

НЕКОТОРЫЕ ФИКСИРОВАННЫЕ ПОРТЫ

❖ 80 – HTTP

❖ 443 – HTTPS

❖ 53 – DNS

❖ 20-21 – 

❖ 35 – 

НЕКОТОРЫЕ ФИКСИРОВАННЫЕ ПОРТЫ

❖ 80 – HTTP

❖ 443 – HTTPS

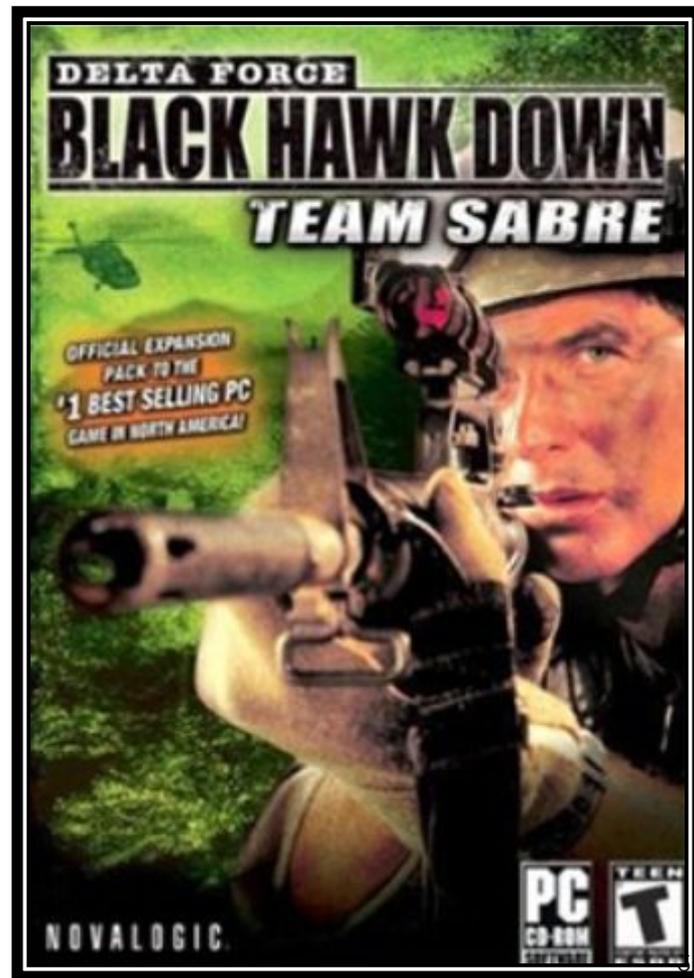
❖ 53 – DNS

❖ 20-21 – FTP

❖ 35 – 

НЕКОТОРЫЕ ФИКСИРОВАННЫЕ ПОРТЫ

- ❖ 80 – HTTP
- ❖ 443 – HTTPS
- ❖ 53 – DNS
- ❖ 20-21 – FTP
- ❖ 35 – Delta Force



СОКЕТ – СВЯЗКА ПОРТА И IP-АДРЕСА

❖ **Socket (Connect) Timeout Exception**

Это когда ответный сетевой пакет не получен.
Чаще всего значит, что к узлу **нет доступа**.

❖ **Connection Refused**

Это когда узел сознательно отвергает сетевой пакет.
Чаще всего значит, что **доступ есть**, но запрашиваемый порт **никем не слушается**.

РЕЗЮМЕ ПО ТСР

- ❖ Соединения обходятся **дорого** по ресурсам узла:
 - ❖ Буферы/таймеры/счетчики
- ❖ В каждом соединении участвуют только **2 узла** (дуплекс)
 - ❖ Но на **одном порту** может быть **много соединений**
- ❖ Мультиплексирование работает на основе **пар сокетов**
- ❖ А если надо что-то **попроще?**

USER DATAGRAM PROTOCOL (UDP)

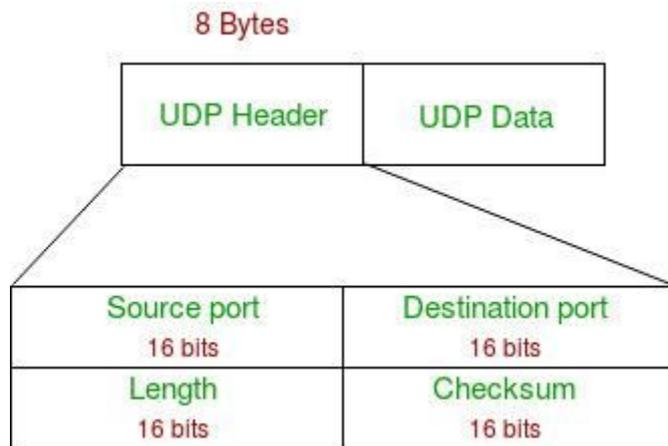
Второй по важности житель
транспортного уровня TCP/IP



UDP: ОСНОВНОЕ

- ❖ Никаких соединений
- ❖ Доставка “как смог”
- ❖ Целостность опциональна
- ❖ Зато **легко** и **быстро**

PDU @ UDP: ДАТАГРАММА



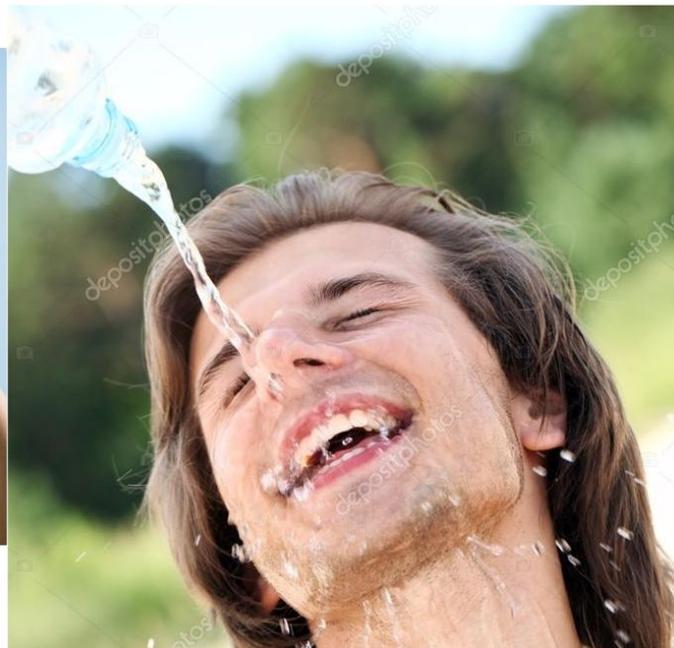
- ❖ Те же номера портов от 0 до 65535
- ❖ Но сами порты **другие**

TCP VS UDP: УПРАВЛЕНИЕ ПОТОКОМ

TCP



UDP





TCP VS UDP



second packet sent



fifth packet sent

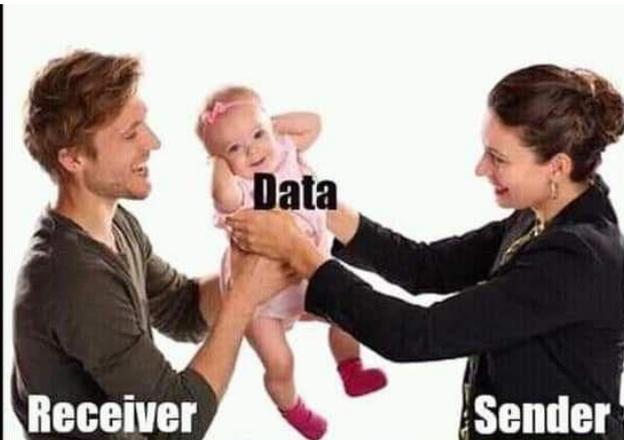


last packet sent

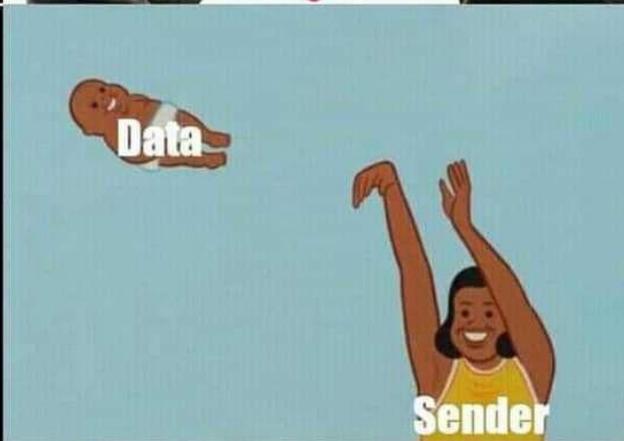


TCP vs UDP: КОНТРОЛЬ ДОСТАВКИ

TCP



UDP



TCP **vs** UDP:
ЦЕЛОСТНОСТЬ



Should be:



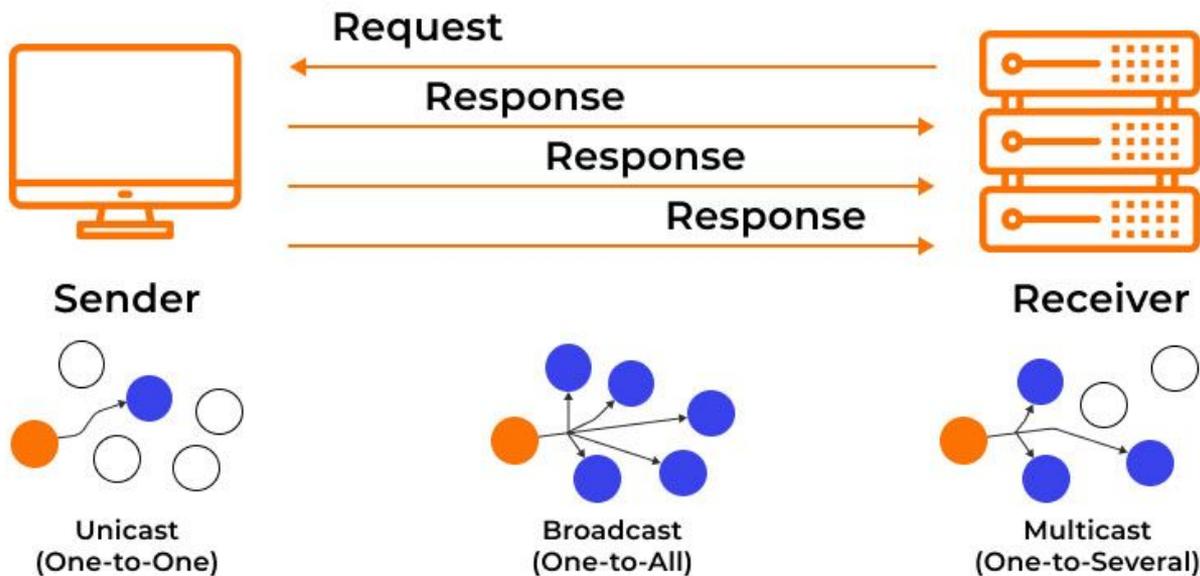
UDP: ПРИМЕНЕНИЯ

- ❖ Потокное медиа (звук, видео)
- ❖ Онлайн игры
- ❖ DNS
- ❖ VoIP
- ❖ (ещё 100500)

ВАРИАНТЫ РАССЫЛКИ ДАТАГРАММ В UDP



How User Datagram Protocol (UDP) Works





15	1 000 000
14	500 000
13	250 000
12	125 000
11	64 000
10	32 000

На какой порт шлёт данные команда ping?

А TCP 80

С Морской

В UDP 32

Д Никакой





15	1 000 000
14	500 000
13	250 000
12	125 000
11	64 000
10	32 000

На какой порт шлёт данные команда ping?

A TCP 80

C Морской

B UDP 32

Никакой



TCP & UDP: СВОДКА

- ❖ Один ~~умный~~ надежный
- ❖ Второй ~~красивый~~ быстрый
- ❖ А если хочется **всего и сразу?**

QUIC

Транспортный протокол XXI века



QUIC: ОСНОВНОЕ

- ❖ Работает поверх UDP
- ❖ Встраивает шифрование в себя
- ❖ Устанавливает и восстанавливает соединения быстро
- ❖ Сохраняет соединения при смене сети
- ❖ Спроектирован под HTTP/3
- ❖ Оформлен как RFC в 2021-ом году

ТРАНСПОРТНЫЙ УРОВЕНЬ: РЕЗЮМЕ

- ❖ По модели OSI – уровень L4
- ❖ На этом уровне появляются “порты”
- ❖ Два основных протокола: TCP & UDP
 - Единица данных у TCP: сегмент
 - Единица данных у UDP: датаграмма
- ❖ TCP надежный
- ❖ UDP быстрый

АДРЕСАЦИЯ В TCP/IP

От макушки до пяток

7.

Прикладной уровень

6.

Уровень представления

5.

Сеансовый уровень

4.

Транспортный уровень

3.

Сетевой уровень

2.

Канальный уровень

LLC

MAC

1.

Физический уровень

АДРЕСАЦИЯ В TCP/IP

❖ В TCP/IP используется **3 вида** адресов:

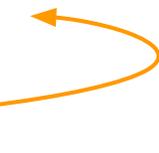
❖ Локальные (аппаратные, MAC)

❖ Сетевые (IP)

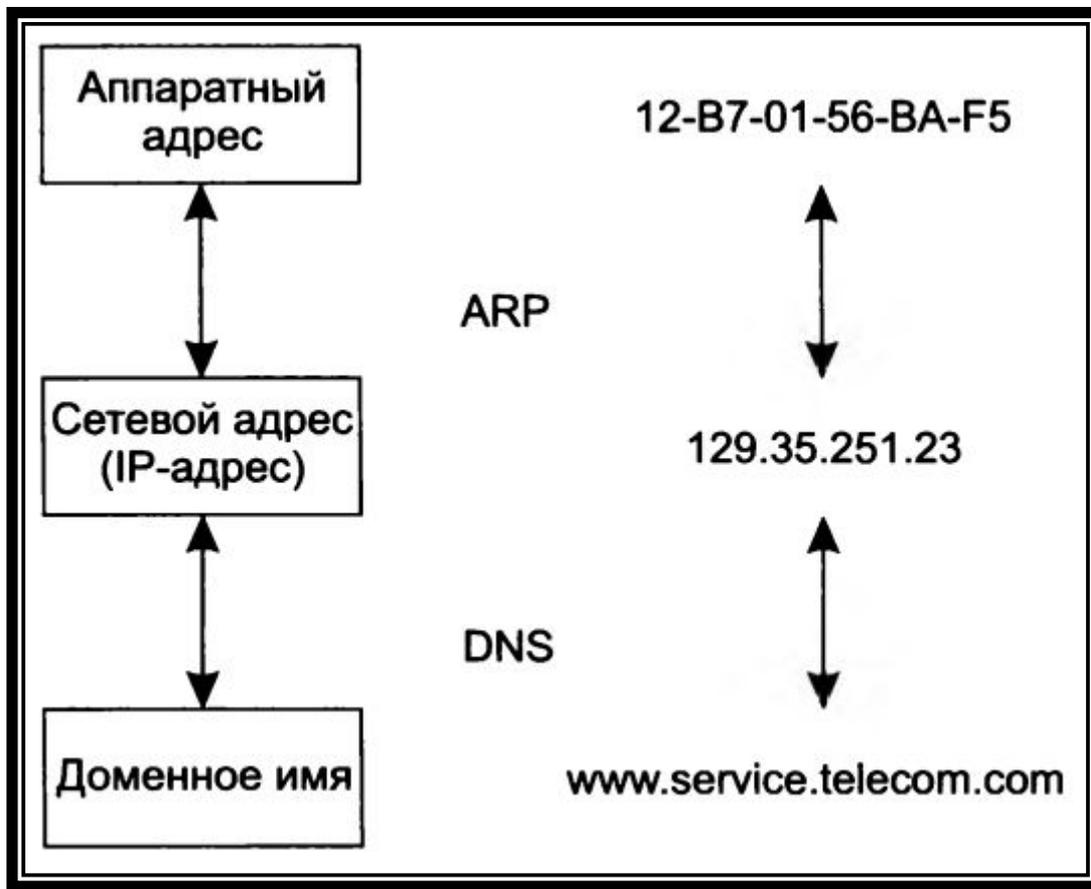
❖ Доменные (символьные, DNS)

ARP

DNS



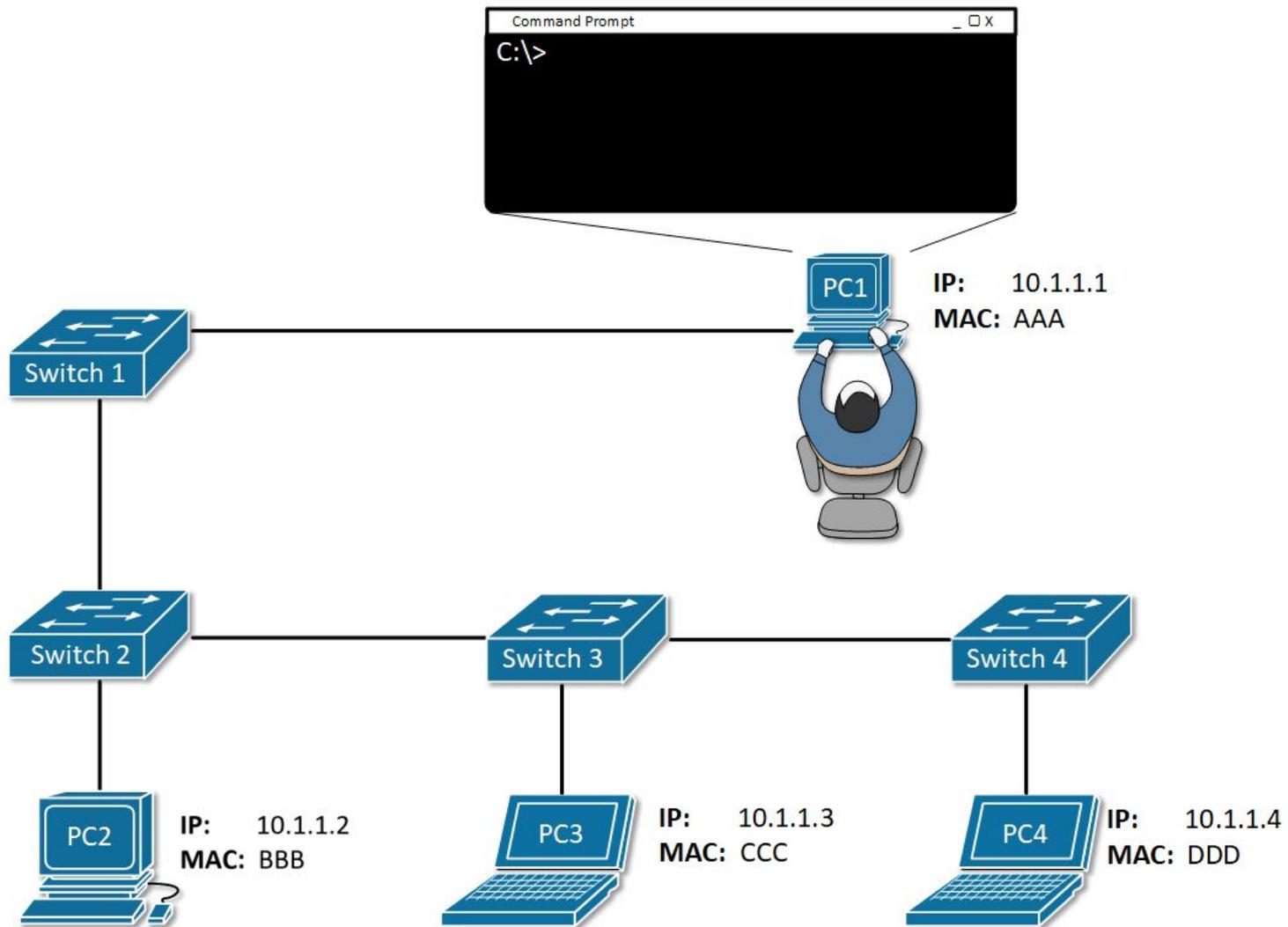
РАЗРЕШЕНИЕ АДРЕСОВ В TCP/IP



ADDRESS RESOLUTION PROTOCOL (ARP)

Преобразует сетевые имена (IP)
в физические (MAC)





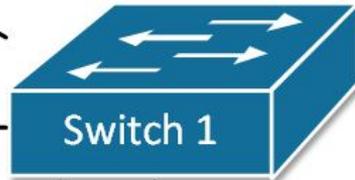
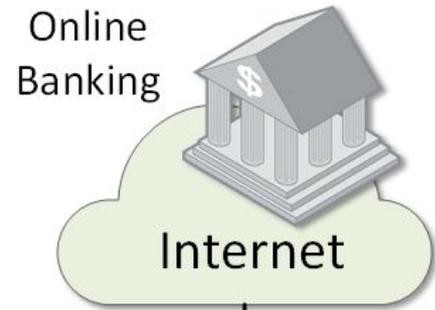
ARP: ОСНОВНОЕ

- ❖ Сопоставляет адреса в IPv4
 - В IPv6 вместо него используется NDP (Neighbor Discovery Protocol)
- ❖ Кэширует адреса
- ❖ Подвержен атаке ARP Spoofing / Poisoning...



ARP Spoofing

192.168.1.0/24

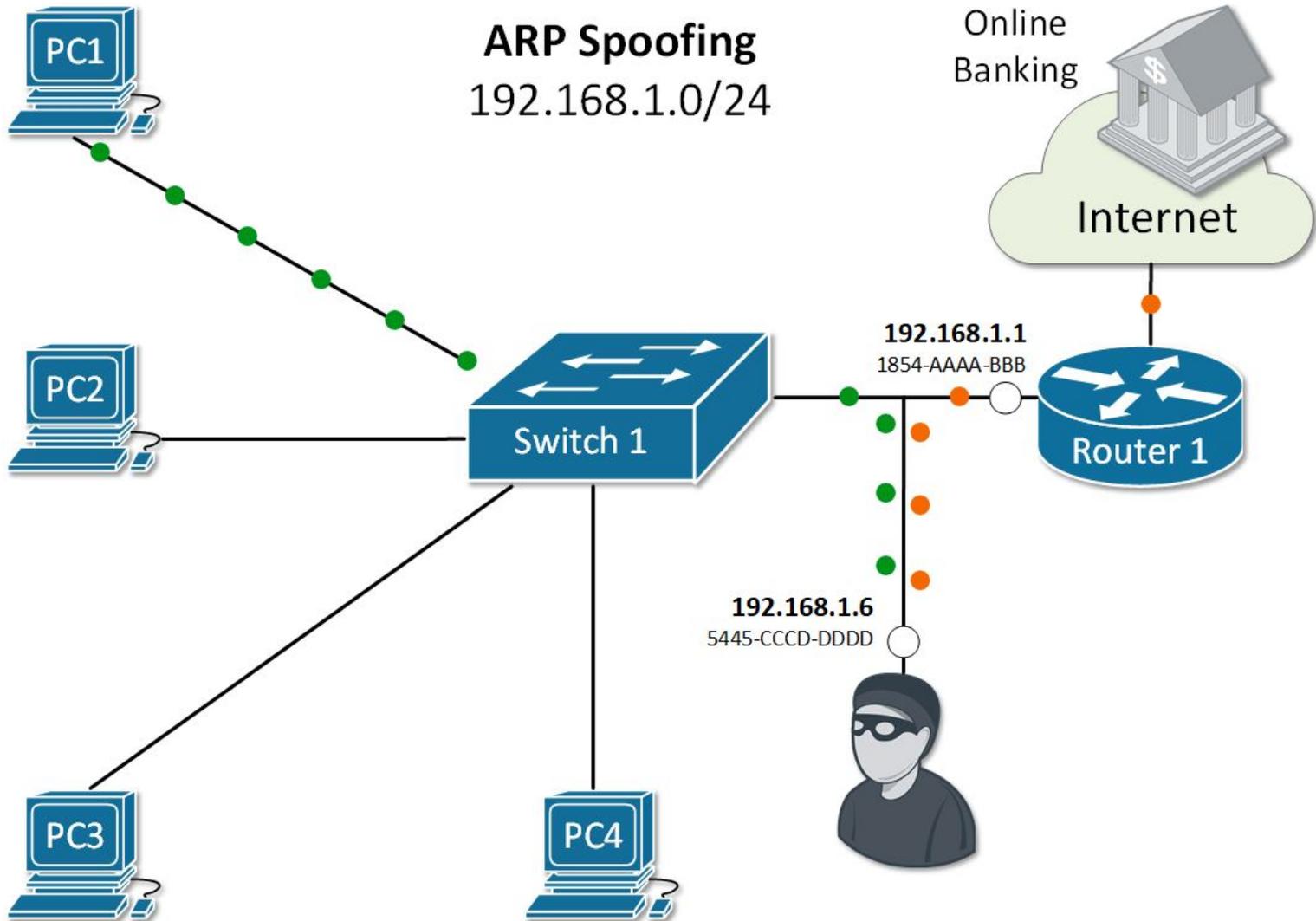


192.168.1.1
1854-AAAA-BBB



192.168.1.6
5445-CCCD-DDDD





DOMAIN NAME SERVICE

Преобразует доменные адреса (DNS)
в сетевые (IP)



7.

Прикладной уровень

6.

Уровень представления

5.

Сеансовый уровень

4.

Транспортный уровень

3.

Сетевой уровень

2.

Канальный уровень

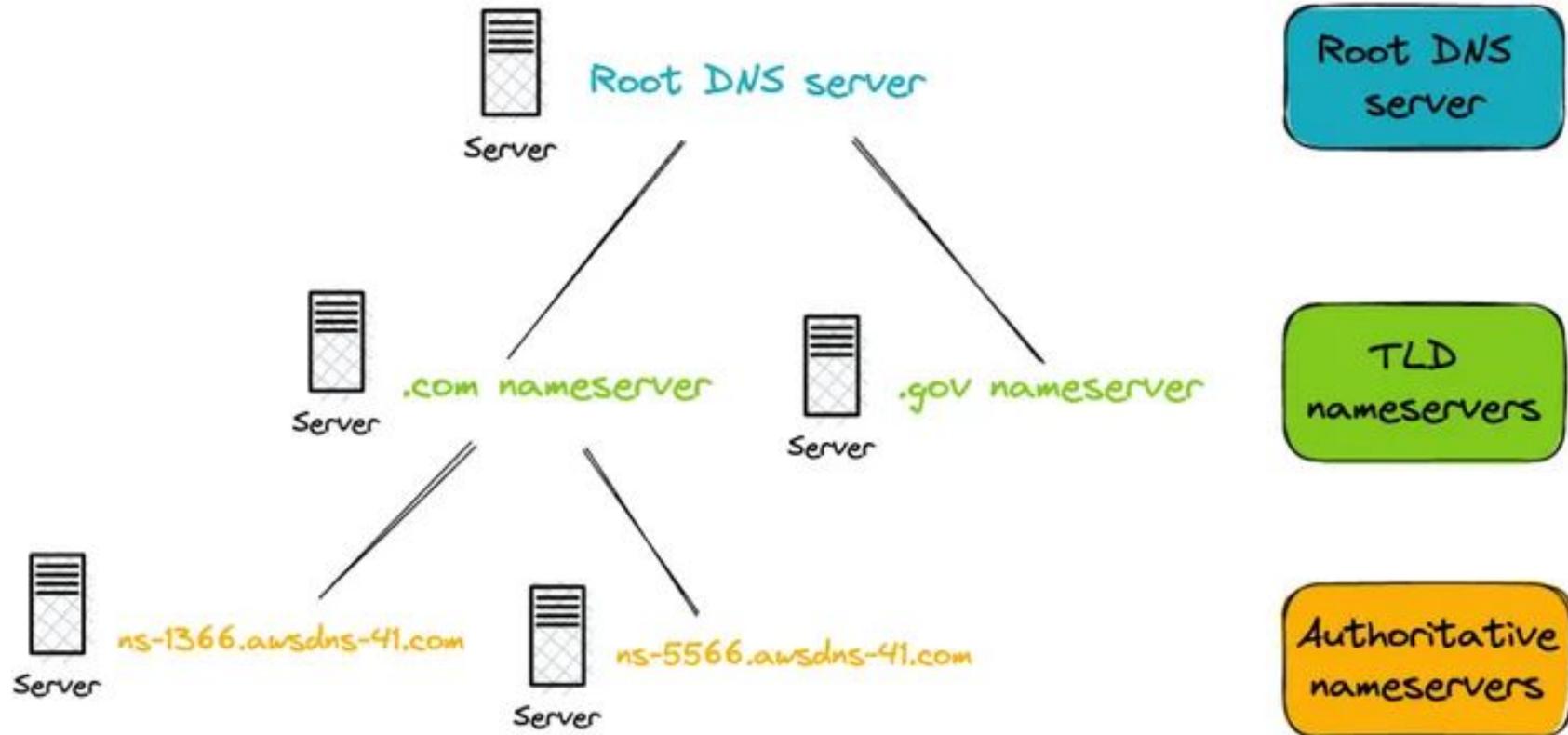
LLC

MAC

1.

Физический уровень

DNS СЕРВЕРЫ СОСТАВЛЯЮТ ИЕРАРХИЮ



НО МОЖНО И БЕЗ НИХ

- ❖ На каждом узле есть файл **hosts**:
 - ❖ Windows: `%SYSTEM32%/drivers/etc/hosts`
 - ❖ Linux/MacOS: `/etc/hosts`
- ❖ Пример содержимого файла `hosts`:

```
# 127.0.0.1          localhost
127.0.0.1          yandex.ru
```

DNS КЛИЕНТЫ ВСТРОЕНЫ В КАЖДУЮ ОС

- ❖ Называются «резольверами»
- ❖ Имеют собственный кэш 
- ❖ Не требуют явного вызова из прикладного кода

DNS ПРОТОКОЛ

- ❖ В качестве транспорта обычно использует UDP
- ❖ За службой DNS зарезервирован порт 53
- ❖ Может заполняться как по запросу, так и в фоне
- ❖ Основная утилита: nslookup

ABOVE TCP/IP

Адресация на прикладном уровне



РЕСУРСЫ В СЕТИ ОБОЗНАЧАЮТСЯ С ПОМОЩЬЮ URI & URL

URI

Идентификатор ресурса. Указывает, как называется ресурс. Но не обязан сообщать, где ресурс находится.

Пример: myfile.txt

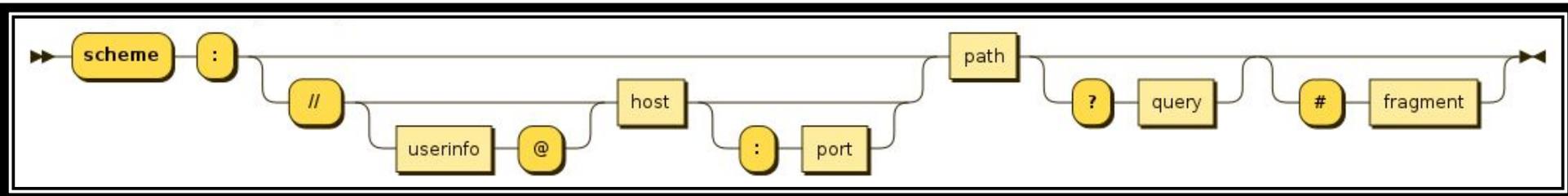
URL

Локатор ресурса.

Указывает не только имя ресурса, но и где ресурс находится.

Является подмножеством URI.

СХЕМА СОСТАВЛЕНИЯ URI



ПРИМЕРЫ URI

- ❖ `https://ru.wikipedia.org:443/wiki/URI`
- ❖ `ftp://ftp.is.co.za/rfc/rfc1808.txt`
- ❖ `file://C:\UserName.HostName\Projects\URI.xml`
- ❖ `ldap://[2001:db8::7]/c=GB?objectClass?one`
- ❖ `mailto:John.Doe@example.com`
- ❖ `tel:+1-816-555-1212`

КАК ПРАВИЛЬНО ЧИТАТЬ URL?

`http://angara.ftc.ru:8081/artifactory/home.html?id=1&v=2`

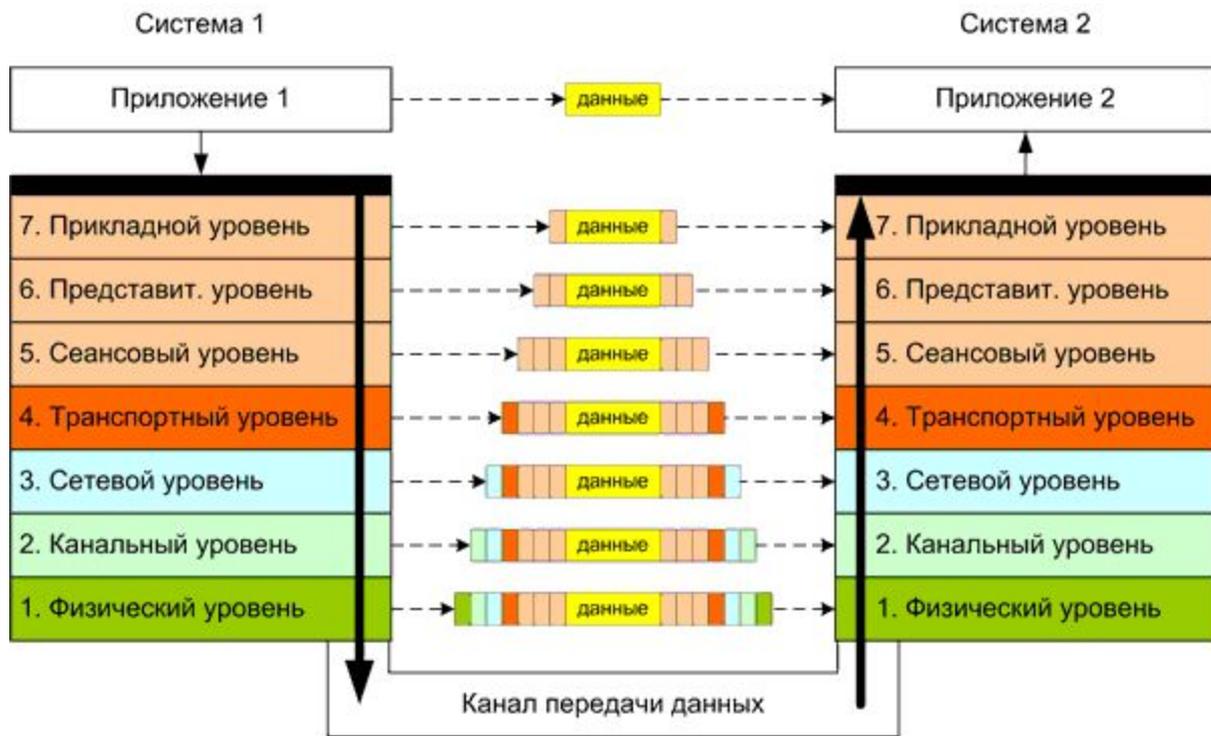
1. Адрес хоста – `angara.ftc.ru`
2. Порт – `8081`
3. Протокол – `http`
4. Путь к ресурсу – `artifactory/home.html`
5. Параметры – `id=1&v=2`

CONCLUSION

Краткая суть предыдущих 100 слайдов

РЕЗЮМЕ

- ❖ Сетевое взаимодействие имеет несколько **уровней**
- ❖ На каждом уровне работают свои **протоколы**
- ❖ Понять, на каком уровне сетевая проблема – залог **успеха** в ее решении



- Логическое соединение между уровнями
- Реализация передачи данных

ЧТО ПОЧИТАТЬ



(руководства
к ПО
и аппаратуре)

Есть вопросы?

Задавайте.

Владимир Плизга

toparvion.pro