

Основы сетевого взаимодействия



Протокол

- Протокол — набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более сторонами
- Стороны (peers) могут быть физическими устройствами, логическими устройствами, процессами
- Протокол состоит из:
 - Описания формата передачи данных
 - Описания возможных состояний работы сторон
 - Описания правил перехода из состояния в состояние

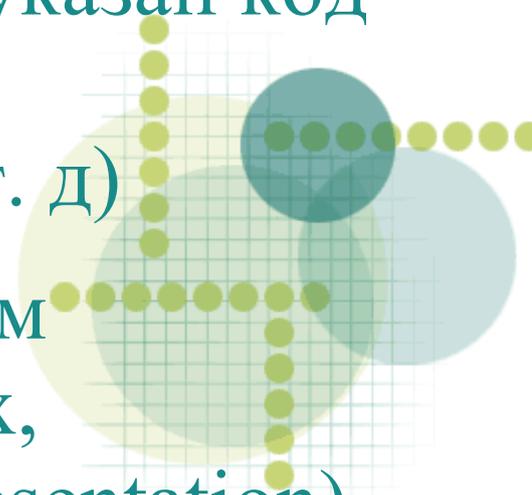


Пример взаимодействия

- Пример: заказ товара почтой, рассмотрим полный путь от отправителя до получателя
- Для заказа нужно заполнить некоторую форму, в которой указать код товара, заказываемое количество
- Форма определяет формат, понятный заказчику и исполнителю, то есть формат обмена данными между ними
- Конкретные приложения взаимодействуют на *прикладном уровне* (application layer)



Пример взаимодействия - 2

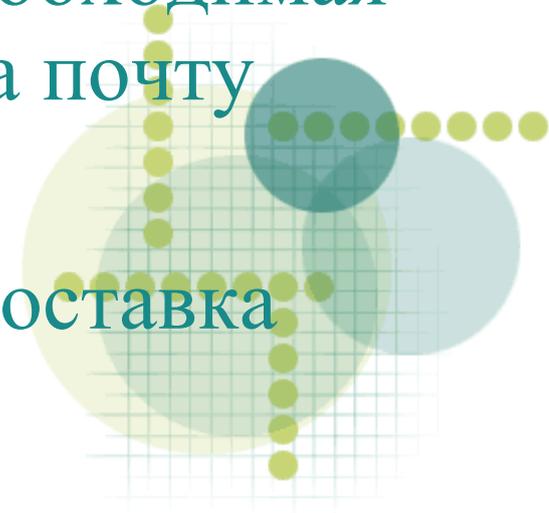
- Если магазин функционирует на международном уровне, необходима договоренность о том, на каком языке заполняется форма
 - Например, мы потребуем вложения формы заказа в конверт, на котором будет указан код языка, на котором заполнена форма (английский, русский, казахский и т. д.)
 - Уровень взаимодействия, на котором определяется представление данных, называется *представительским* (presentation)
- 

Пример взаимодействия - 3

- Магазин может предоставлять скидки покупателям с историей, дарить подарки или еще как-либо поддерживать историю взаимодействия
- Для этого конверт с надписанным языком нужно вложить в конверт, на котором должен быть написан уникальный номер покупателя, номер его покупки и прочая информация
- Уровень, который обеспечивает длительное взаимодействие, называется *сеансовым* (session layer)

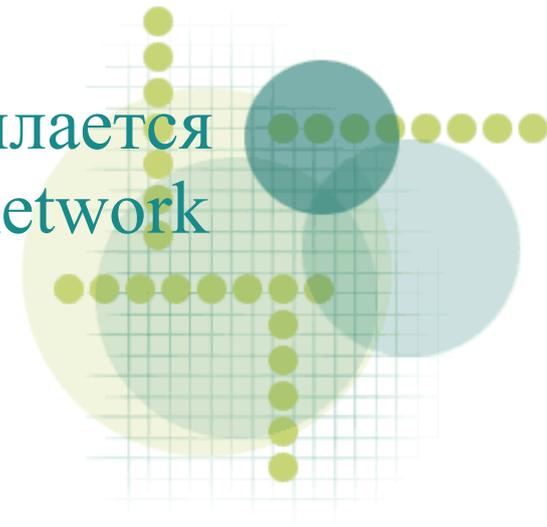
Пример взаимодействия - 4

- Чтобы почта могла доставить заказ, необходимо указать адрес доставки письма
- Для этого конверт, на котором написана информация о номере клиента и пр., вкладывается в конверт, на котором пишется почтовый адрес, индекс и другая необходимая информация — конверт отдается на почту (транспортный агент)
- Уровень, на котором выполняется доставка сообщений (писем), называется *транспортным* (transport layer)



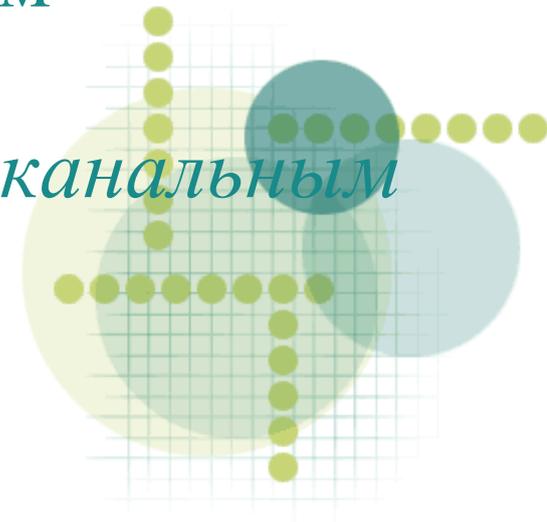
Пример взаимодействия - 5

- Почта, как правило, не доставляет письмо от отправителя до получателя непосредственно: письма могут проходить через несколько промежуточных пунктов, на каждом из которых пересортировываться
- Письма, отправляемые на один сортировочный пункт, упаковываются в мешки, на которых пишется адрес сортировочного пункта
- Уровень, на котором информация пересылается между узловыми пунктами — *сетевой* (network layer)



Пример взаимодействия - 6

- Для доставки мешков писем между узловыми пунктами, например, на самолете, необходимо упаковать мешки в самолетный контейнер, написать на контейнере сопроводительную документацию
- Уровень взаимодействия, на котором информация готовится к пересылке физическим носителем, называется *канальным* (data-link layer)



Пример взаимодействия - 7

- Наконец, мешки грузятся в самолет, самолет летит по определенному маршруту, на определенной высоте в соответствии с планом полета
- Уровень, на котором данные реально доставляются, называется *физическим* (physical layer)

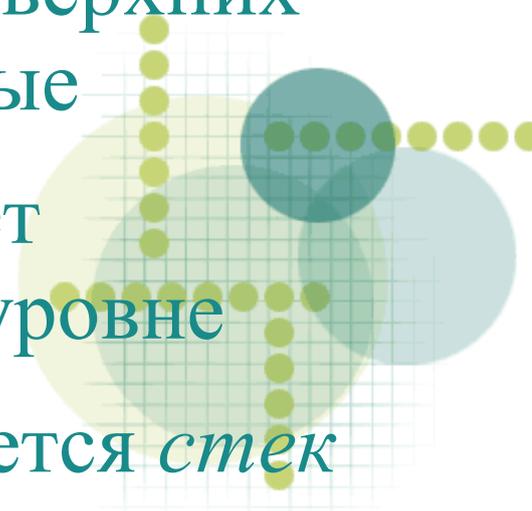


Пример взаимодействия

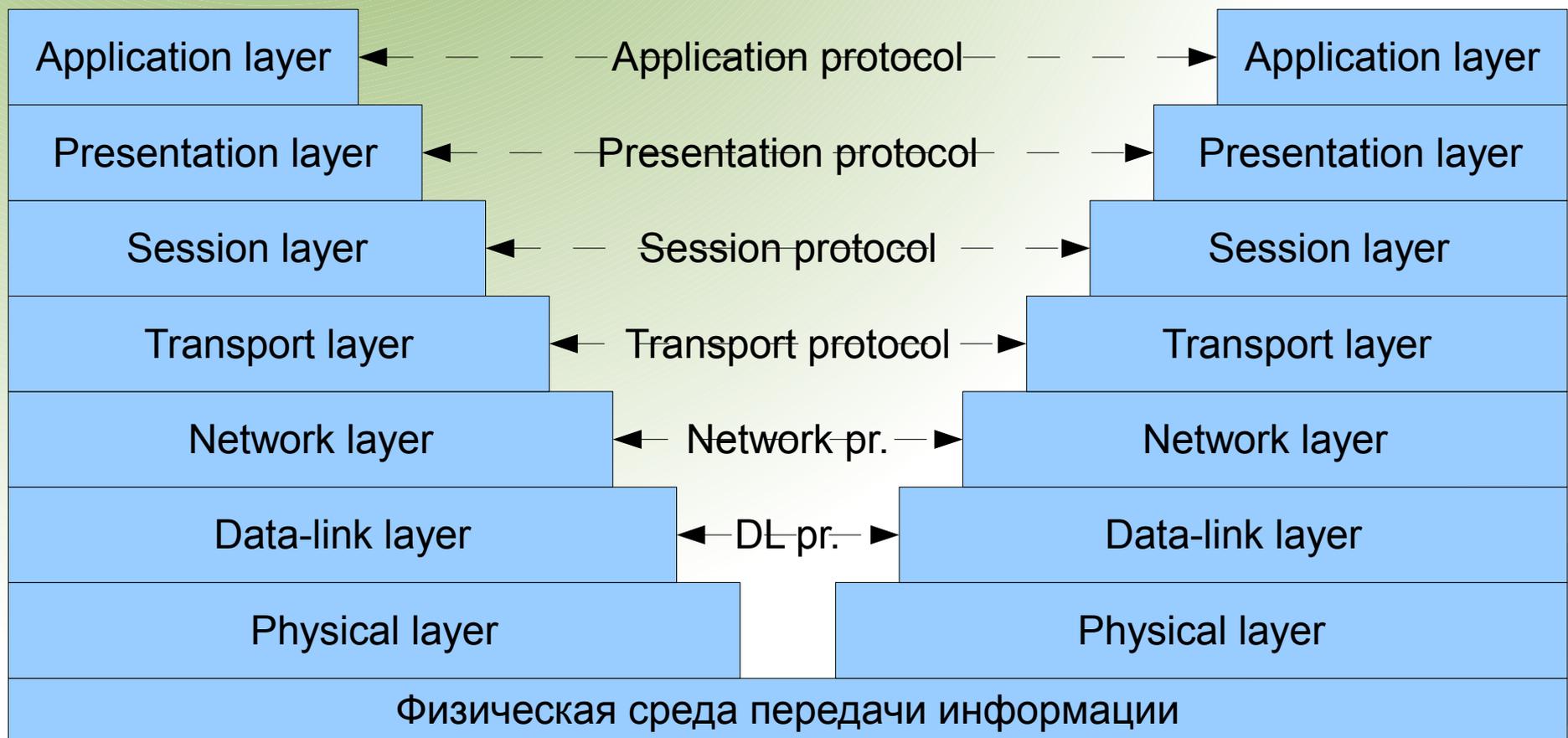
- При поступлении в точку назначения заказ проходит все ступени снизу вверх. На каждом уровне удаляется контейнер или конверт соответствующего уровня и данные передаются на уровень вверх.



Стек протоколов

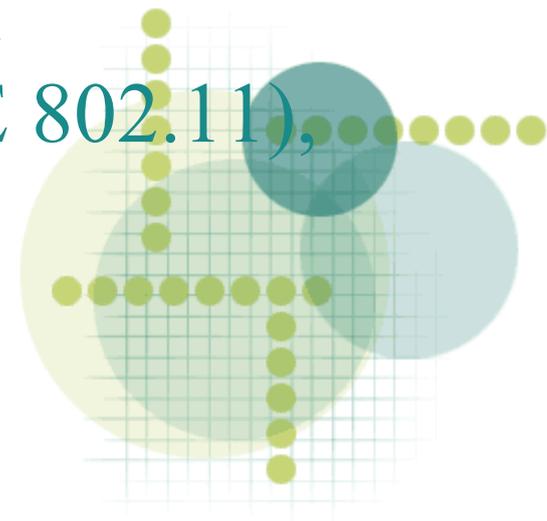
- Для протокола верхнего уровня не важно, каким образом данные будут доставлены до получателя и какая вспомогательная информация при этом еще потребуется
 - Для протокола нижнего уровня не важно, какие именно данные поступили от верхних уровней, доставляются любые данные
 - Протокол каждого уровня описывает взаимодействие сторон на этом же уровне
 - Такое разделение на уровни называется *стек протоколов*
- 

Эталонная модель ISO OSI



Сети TCP/IP

- Работают поверх различных протоколов канального и физического уровня, обеспечивающих пакетную пересылку данных
- Протоколы канального и физического уровня не специфицируются
- На практике используются: Ethernet (протоколы IEEE 802.3), WiFi (IEEE 802.11), GPRS/EDGE, и т. д.



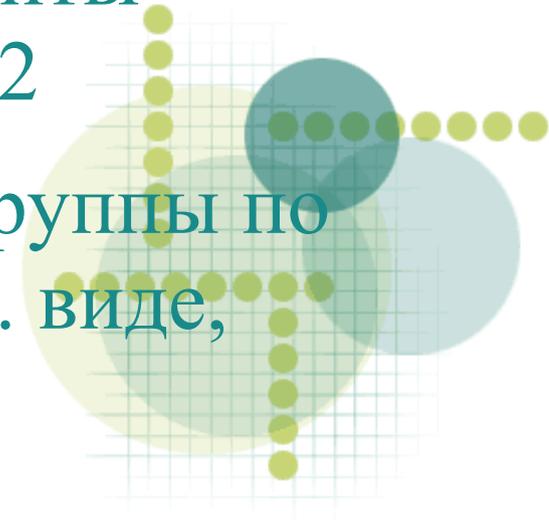
Коммутация каналов и коммутация пакетов

- Коммутация каналов — для пересылки сообщения устанавливается физический канал связи (пример — телефонная связь)
- Коммутация пакетов — сообщение разбивается на *пакеты*, каждый пакет доставляется независимо от других
- В сети с коммутацией пакетов возможно создание виртуальных каналов



Адресация компьютеров

- Каждое физическое или логическое устройство в TCP/IP сетях имеет адрес, уникальный в рамках этой сети или глобально
- У протокола IP v4 размер адреса — 32 бита, адрес делится на 4 байта, каждый байт записывается в десятичном виде, байты разделяются точкой: 212.192.248.182
- IP v6 — 128 бит, адрес делится на группы по 16 бит, каждая записывается в шест. виде, разделяются двоеточием:
fe80::21d:7dff:fe00:e4d6



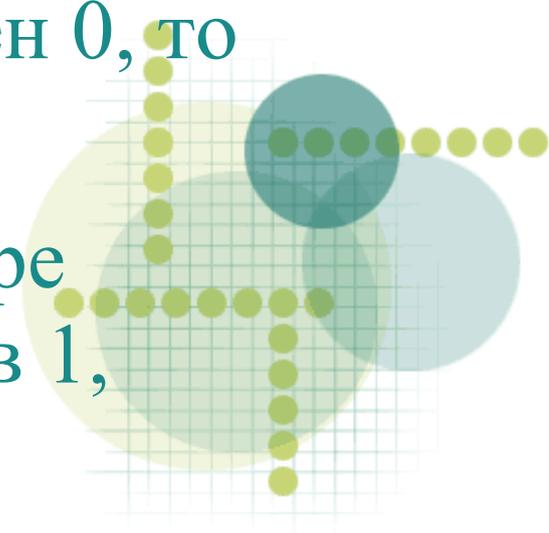
Специальные адреса IPv4

- 127.0.0.1 — localhost — «встроенная» сеть, которая состоит из одного данного компьютера, присутствует на каждом компьютере, даже когда он не подключен к сети
- 192.168.X.Y — адреса, выделенные для локального использования. Выход в Интернет возможен через прокси-сервера или NAT
- 10.X.Y.Z — адреса, выделенные для локального использования



Сети IPv4

- Адрес состоит из двух частей: адрес сети и номер компьютера в сети
- Адрес сети может иметь переменный размер, который записывается как число бит под адрес: 192.168.10.0/24
- Адрес сети: номер компьютера равен 0, то есть 192.168.10.0
- Широковещательный адрес: в номере компьютера все биты установлены в 1, например, 192.168.10.255



Классы сетей

0	Network (7 bits)	Host (24 bits)	A
10	Network (14 bits)	Host (16 bits)	B
110	Network (21 bits)	Host (8 bits)	C
1110	Адрес многоадресной рассылки		D
11110	зарезервировано		E

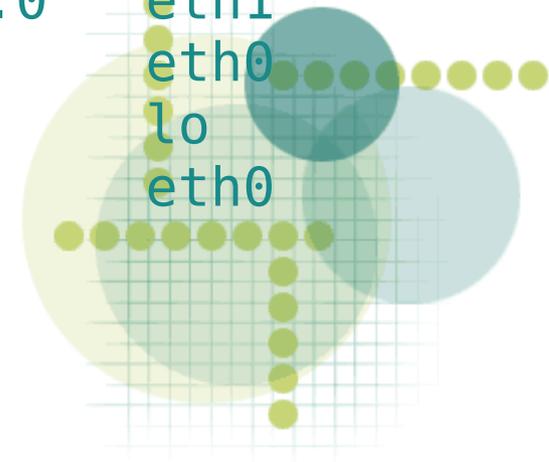
Классовое деление сетей в настоящее время не используется, используются сети с переменным размером (CIDR).

Маршрутизация в IP-сетях

- Для того, чтобы определить, на какое из сетевых устройств, необходимо посылать пакет с IP-адресом, используются таблицы IP-маршрутизации

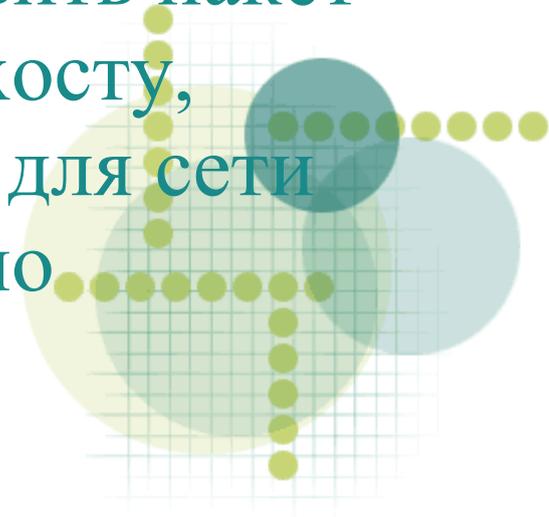
```
# route -n
Kernel IP routing table
Destination      Gateway
192.168.10.0     0.0.0.0
10.3.0.0         0.0.0.0
127.0.0.0        0.0.0.0
0.0.0.0          10.3.0.1
```

```
Genmask          Iface
255.255.255.0   eth1
255.255.0.0     eth0
255.0.0.0       lo
0.0.0.0         eth0
```



Маршрутизатор по умолчанию

- Адрес сети 0.0.0.0 с маской сети 0.0.0.0 в таблице маршрутизации задает маршрутизатор по умолчанию, ядро отправляет на этот адрес IP-пакеты, которые не может маршрутизировать само
- Задача маршрутизатора — переправить пакет либо непосредственно хосту, либо хосту, который указан как маршрутизатор для сети назначения, либо маршрутизатору по умолчанию более высокого уровня

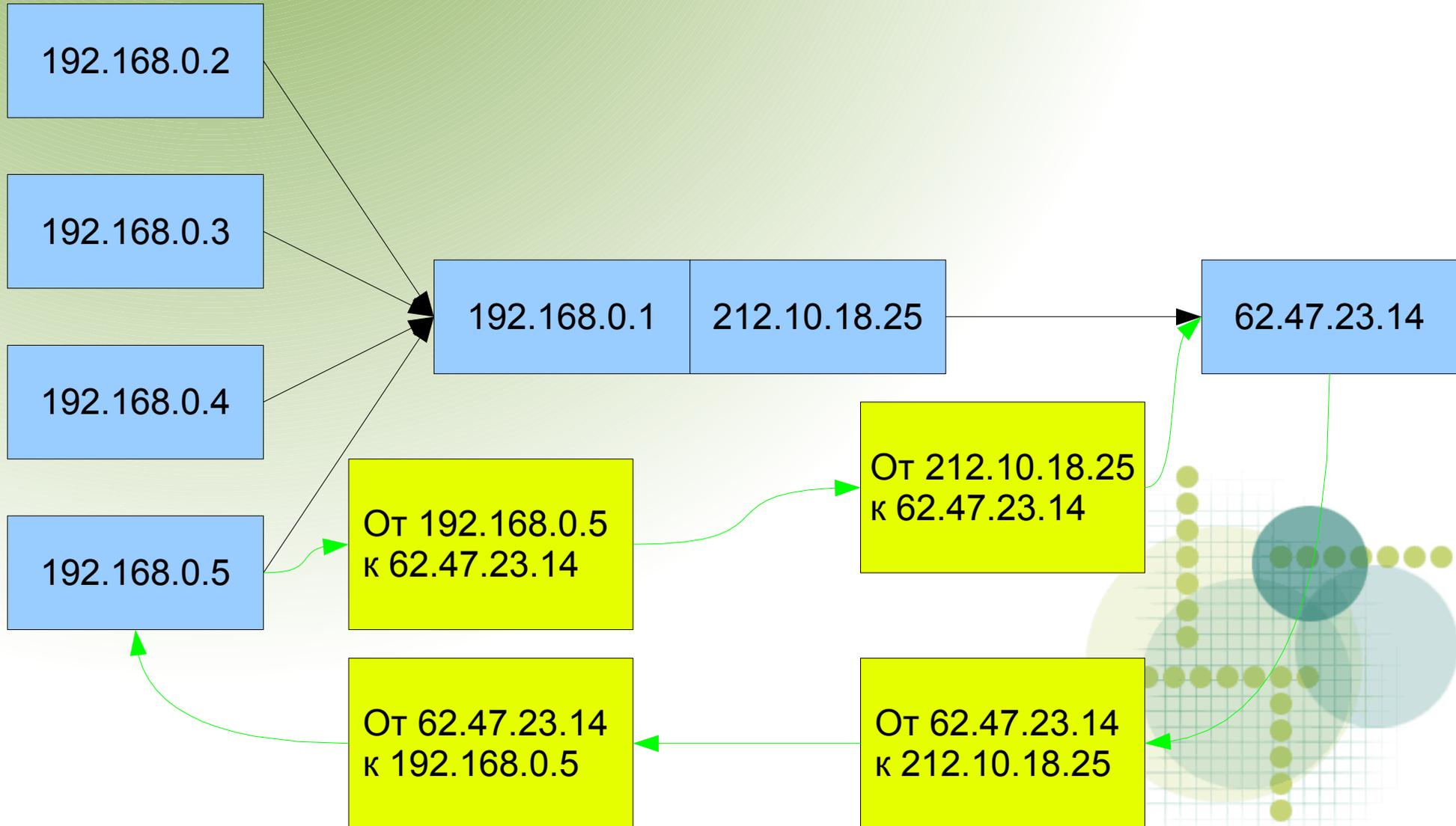


NAT (Network address translation)

- Маршрутизатор скрывает IP-адреса в сети, для которой он является маршрутизатором по умолчанию
- IP-адрес исходного хоста заменяется на внешний адрес самого маршрутизатора, таким образом модифицированные пакеты отправляются вовне
- Пакеты, пришедшие на адрес маршрутизатора, модифицируются и отправляются во внутреннюю сеть



NAT



DNS (domain name system)

- Сервис отображения символических имен в IP-адреса
- Для отображения имени в IP-адрес процесс запрашивает DNS-сервер, который в ответ возвращает IP-адрес
- DNS-сервер может располагаться на другом компьютере

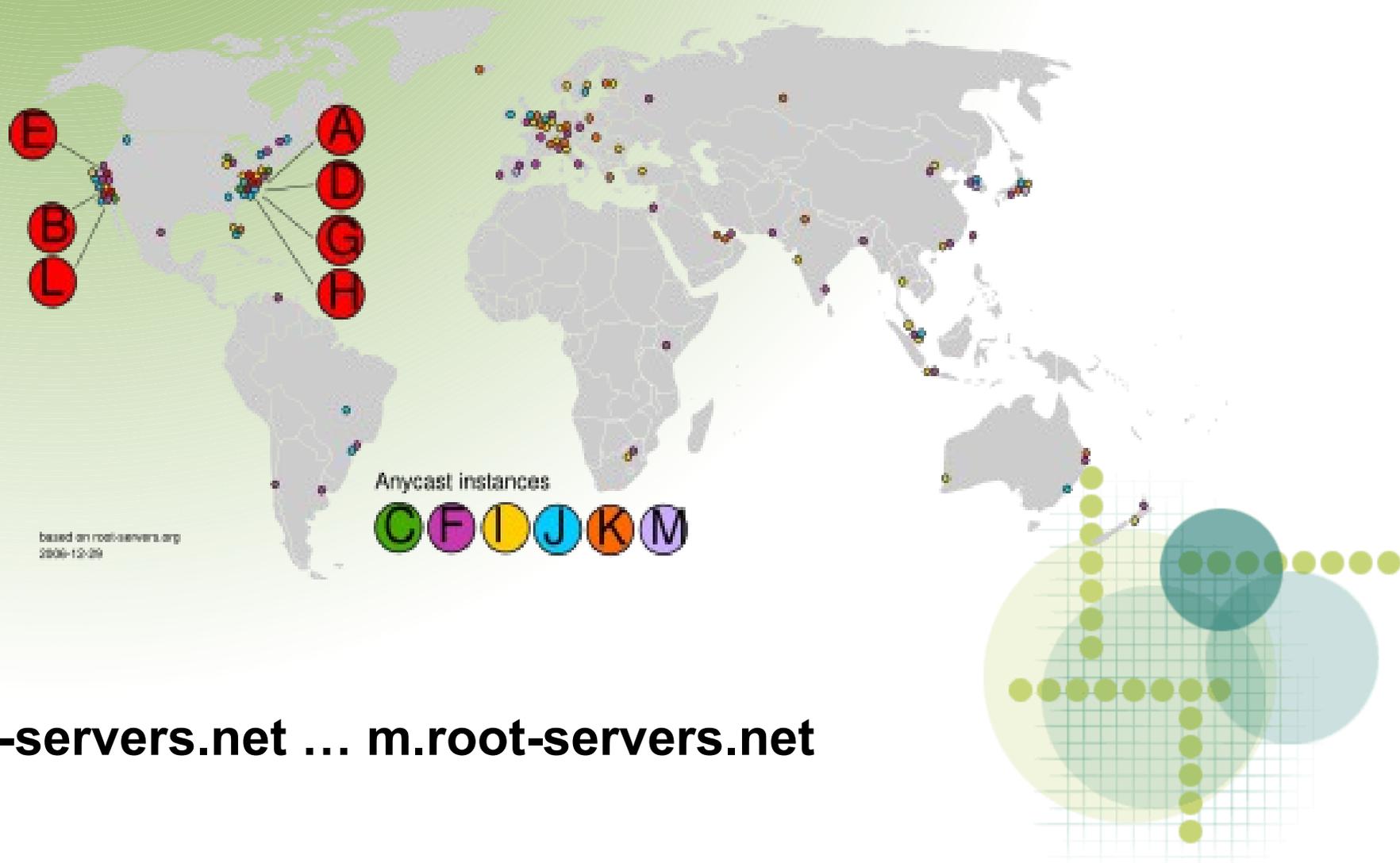


Пример: www.msu.kz

- [kz](#) — домен верхнего уровня
- [msu](#) — домен второго уровня
- [www](#) — имя хоста в домене
- Для получения адресов серверов домена [kz](#) необходимо обратиться к корневым серверам
- Для получения адресов серверов [msu.kz](#) необходимо обратиться к серверам зоны [kz](#)
- Для получения адреса [www.msu.kz](#) необходимо обратиться к серверу зоны [msu.kz](#)

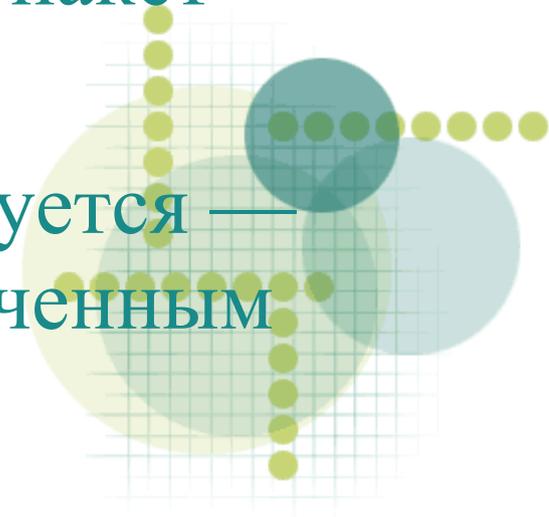


Корневые сервера



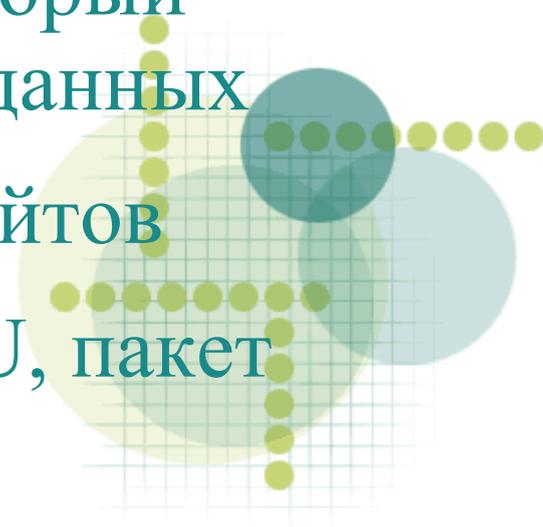
Протокол IP

- Протокол сетевого уровня, доставляющий пакеты от хоста-отправителя до хоста-получателя
- Доставка пакета не гарантируется: если по каким-либо причинам очередной маршрутизатор не может переслать пакет дальше, пакет уничтожается
- Корректность доставки не гарантируется — содержимое может оказаться испорченным



Параметры IP

- TTL (time to live) — количество «прыжков» от хоста к хосту, после которого пакет уничтожается
- MTU (maximal transfer unit) — максимальный размер фрейма (то есть пакета на канальном/физическом уровне), который допустим в данной среде передачи данных
- Например, MTU Ethernet — 1500 байтов
- Если размер пакета превышает MTU, пакет может быть разбит на пакеты, либо уничтожен



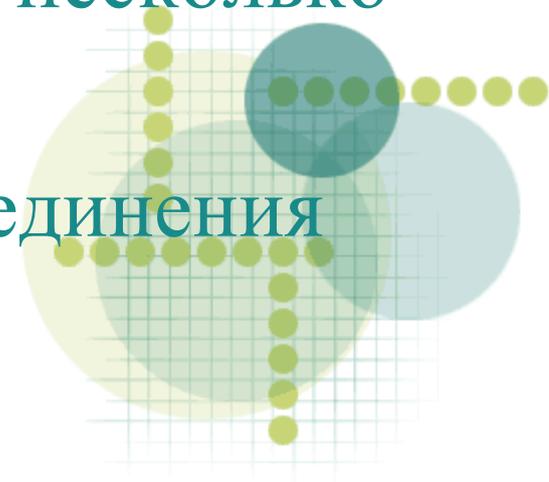
ICMP (Internet Control Message Protocol)

- Протокол сетевого уровня
- Используется для сообщения об ошибках, для проверки доступности компьютера, для обнаружения MTU на пути пакетов и т. д.
- Например, утилита ping посылает на указанный адрес ICMP Echo Request
- В случае успеха хост возвращает ICMP Echo Reply
- В случае недоступности возвращается ICMP Destination Unreachable



Транспортный уровень

- Протоколы транспортного уровня доставляют данные от процесса отправителя до процесса получателя
- На одном хосте могут работать разные процессы, предоставляющие сетевые сервисы
- Один процесс может предоставлять несколько сервисов
- Дополнительная идентификация соединения — номер порта



Порт

- Порт — число от 1 до 65535 (16 бит)
- Порты с номерами до 1024 зарезервированы для использования процессами с правами администратора
- Некоторые номера портов приписаны (то есть обычно используются) стандартными сервисами
- Каждому исходящему запросу или датаграмме присваивается номер порта



Идентификация соединений и запросов

- Каждое соединение или датаграмма идентифицируется 4 числами: IP отправителя, порт отправителя, IP получателя, порт получателя



UDP (User datagramm protocol)

- UDP — протокол транспортного уровня
- Простая обертка над IP, доставка не гарантируется, корректность данных не гарантируется, порядок поступления датаграмм не гарантируется
- Для пересылки данных установление соединения не требуется



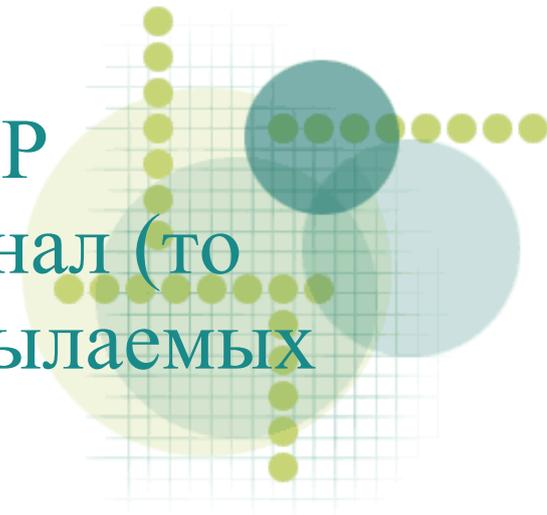
Терминология

- Датаграмма — сообщение на транспортном уровне
- Пакет — сообщение на сетевом уровне
- Фрейм — сообщение на канальном (физическом) уровне



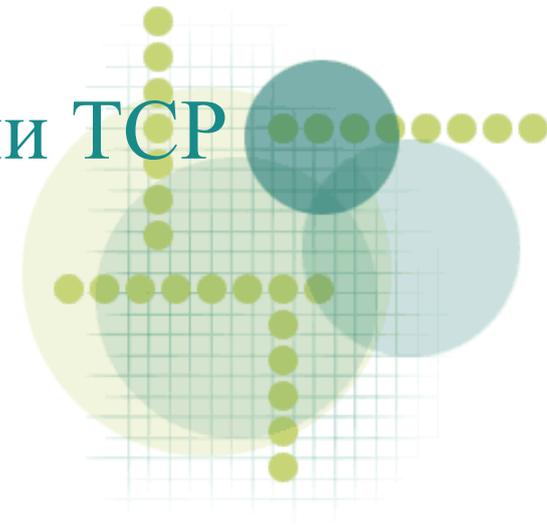
TCP (Transmission Control Protocol)

- TCP — протокол транспортного уровня
- Протокол с установлением виртуального соединения
- Протокол гарантирует доставку данных (при необходимости повторяя пересылку пакета) и гарантирует корректность данных
- После установления соединения TCP предоставляет двунаправленный канал (то есть границы блоков данных, пересылаемых по TCP, не сохраняются)



Протоколы прикладного уровня

- HTTP (hypertext transfer protocol) — TCP
- FTP (file transfer protocol) — TCP
- SSH (secure shell) — TCP
- SMTP (simple mail transfer protocol) — TCP
- DNS — UDP
- NFS (network file system) — UDP или TCP
- NTP (network time protocol) — UDP
- И так далее...

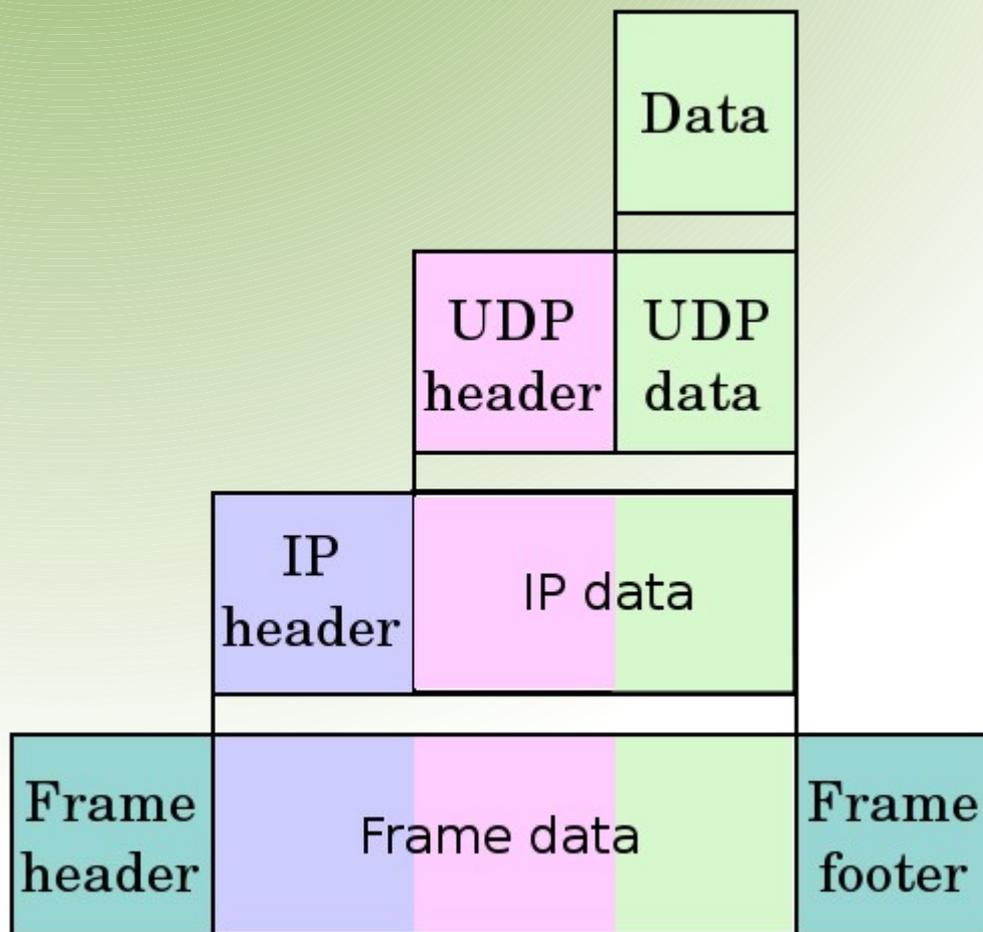


Стандартные номера портов

- TCP/80, TCP/8080 — httpd — веб-сервер
- TCP/22 — sshd — сервер удаленного доступа
- TCP/21, TCP/20 — ftpd — ftp-сервер
- TCP/25 — sendmail — почтовый сервер
- UDP/53 — named — DNS-сервер
- UDP/123 — ntpd — сервер времени
- И так далее



Стек протоколов TCP/IP

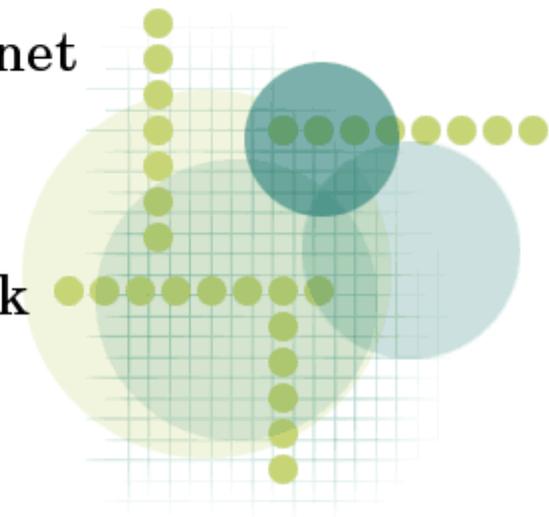


Application

Transport

Internet

Link



Соответствие ISO OSI и TCP/IP

Application layer — приложения
Presentation layer — представления
Session layer — сеансовый
Transport layer — транспортный
Network layer — сетевой
Data-link layer — канальный
Physical layer — физический

Application layer — приложения
Transport layer — транспортный
Internet layer — сетевой
Physical layer — физический

Кодирование данных

- На разных хостах сети могут использоваться разные способы хранения целых и вещественных чисел, разные требования к выравниванию полей структур, разные кодировки символов...
- Преобразование данных в формат передачи данных по сети — сериализация (serialization) или маршаллинг (marshalling)
- Обратное преобразование — десериализация или демаршаллинг

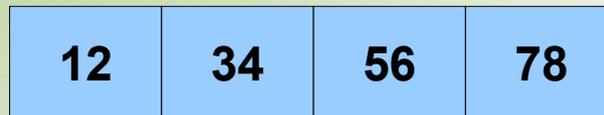


Представление целых чисел

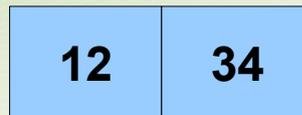
Big-endian

Little-endian

0x12345678



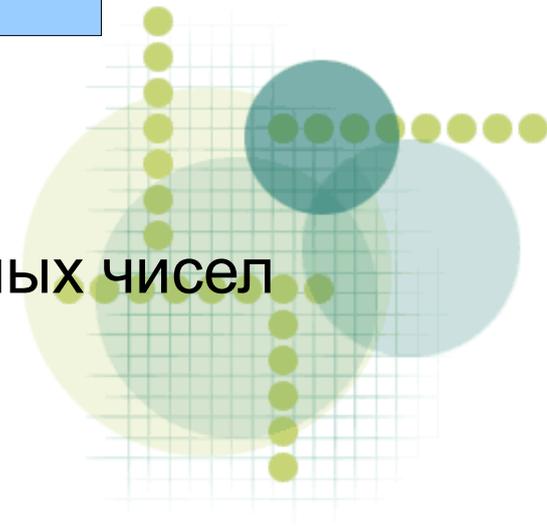
0x1234



PPC, Sparc, Motorola 68000

x86

В качестве сетевого формата представления целых чисел принят Big-endian формат.



Перекодирование целых

- Network byte order — порядок байт в сети
- Host byte order — локальный порядок байт

```
#include <arpa/inet.h>
```

```
uint32_t htonl(uint32_t hostlong);  
uint16_t htons(uint16_t hostshort);  
uint32_t ntohl(uint32_t netlong);  
uint16_t ntohs(uint16_t netshort);
```



XML (eXtensible markup language)

```
<?xml version="1.0" encoding="utf-8" ?>  
<message id="1" sender="x">  
  <entry value="2" />  
  <data>Some text</data>  
</message>
```



XML

- Семейство форматов, настраивается под нужды приложения с помощью описания структуры документа DTD (document type definition). Пример: XMPP (jabber).
- Текстовый формат, не зависит от порядка байт на хосте. В заголовке указывается кодировка текста.
- Позволяет передавать данные произвольной структуры (деревья, графы).
- (-) Огромная избыточность.

