Прикладные физико-технические и компьютерные методы исследований

Семинар 11

Сети

- Совместное использование физических и информационных ресурсов (один принтер на офис/youtube.com)
- Ускорение вычислений (кластеры/grid)
- Повышение надёжности путём дублирования узлов (атомная энергетика)
- Коммуникация и т.д.

Вычислительный кластер

- Самый мощный на данный момент кластер в России и СНГ – «Ломоносов» (МГУ)

Протоколы

- Набор правил, по которым осуществляется общение между удалёнными процессами
- Многоуровневые протоколы

Open System Interconnection

- 7-ми уровневая модель взаимодействия
- Физический: напряжения, частоты … формы разъёмов.
- Канальный: передача информации (пакетов) между непосредственно связанными узлами без искажений.
- Сетевой: передача от узла-отправителя к узлу получателю. Выбор маршрута следования пакетов.

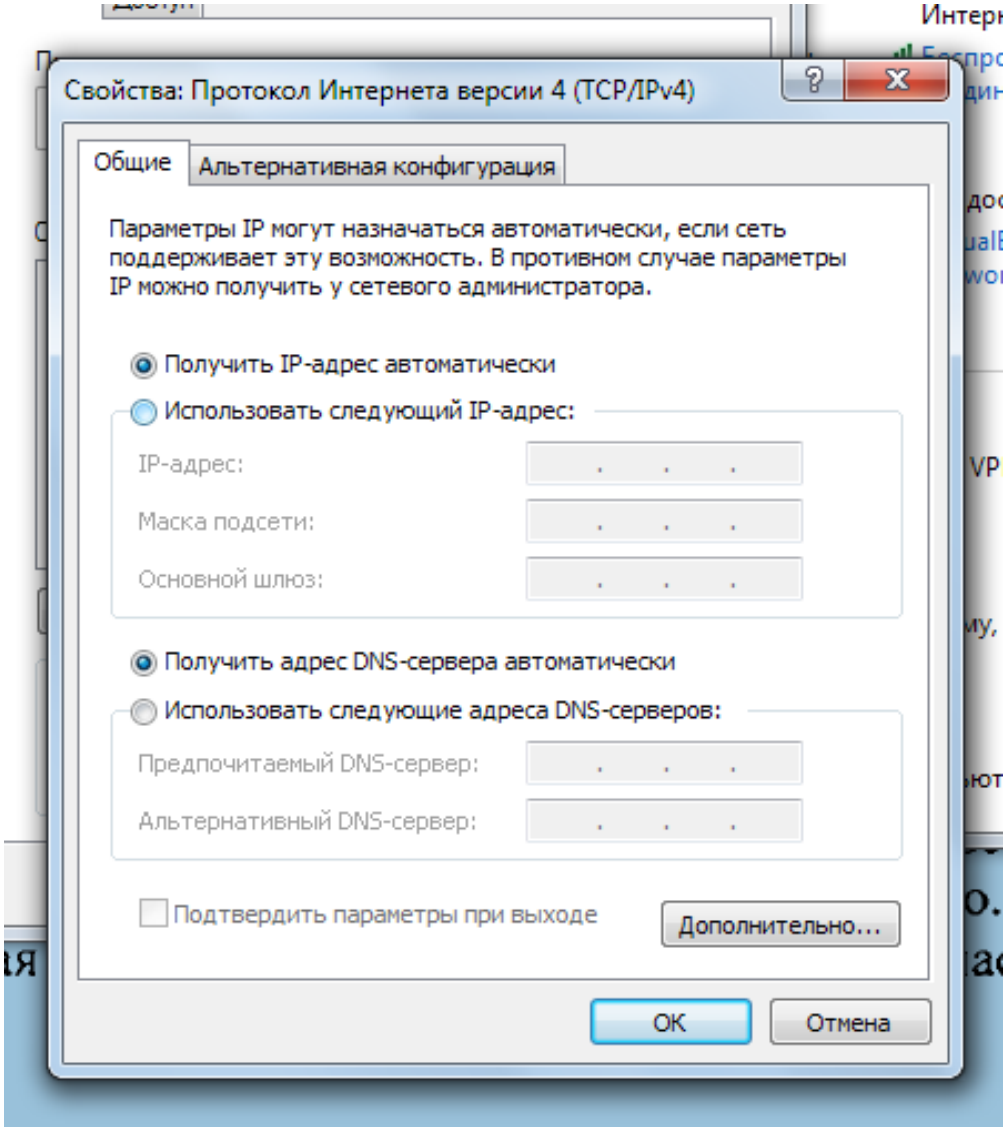
7-ми уровневая модель взаимодействия

- Транспортный: отвечает за надёжность, правильный порядок доставки пакетов, может управлять скоростью передачи данных.
- Сеансовый: синхронизация взаимодействующих процессов. Важно при передаче больших объемов информации.
- Представления данных: шифрования, сжатие и т.п.
- Прикладной: Организация интерфейса между пользователем и сетью.

Адресация

- У каждого пакета есть адрес получателя и адрес отправителя в случае двусторонней связи
- 1-уровневые/2-уровневые адреса

Адресация


Свойства: Протокол Интернета версии 4 (TCP/IPv4)
Общие
Альтернативная конфигурация
Параметры IP могут назначаться автоматически, если сеть поддерживает эту возможность. В противном случае параметры IP можно получить у сетевого администратора.
Получить IP-адрес автоматически
Использовать следующий IP-адрес:
IP-адрес:
Маска подсети:
Основной шлюз:
Получить адрес DNS-сервера автоматически
Использовать следующие адреса DNS-серверов:
Предпочитаемый DNS-сервер:
Альтернативный DNS-сервер:
Подтвердить параметры при выходе
Дополнительно...
OK
Отмена

Локальная адресация. Порты

- PID использовать неудобно, т.к. меняется от запуска к запуску.
- За каждым приложением строго фиксируется номер порта (от 1 до 65535).
- Полный адрес = адрес сетевого адаптера + номер порта

Обмен информацией по сети

- В виде сообщений (UDP)
- Потоковое общение (TCP)

Рождение интернета

- 1969 год – 4 американских института объединили свои вычислительные ресурсы
- В то время не задумывались о надёжности канала связи

Семейство протоколов TCP/IP

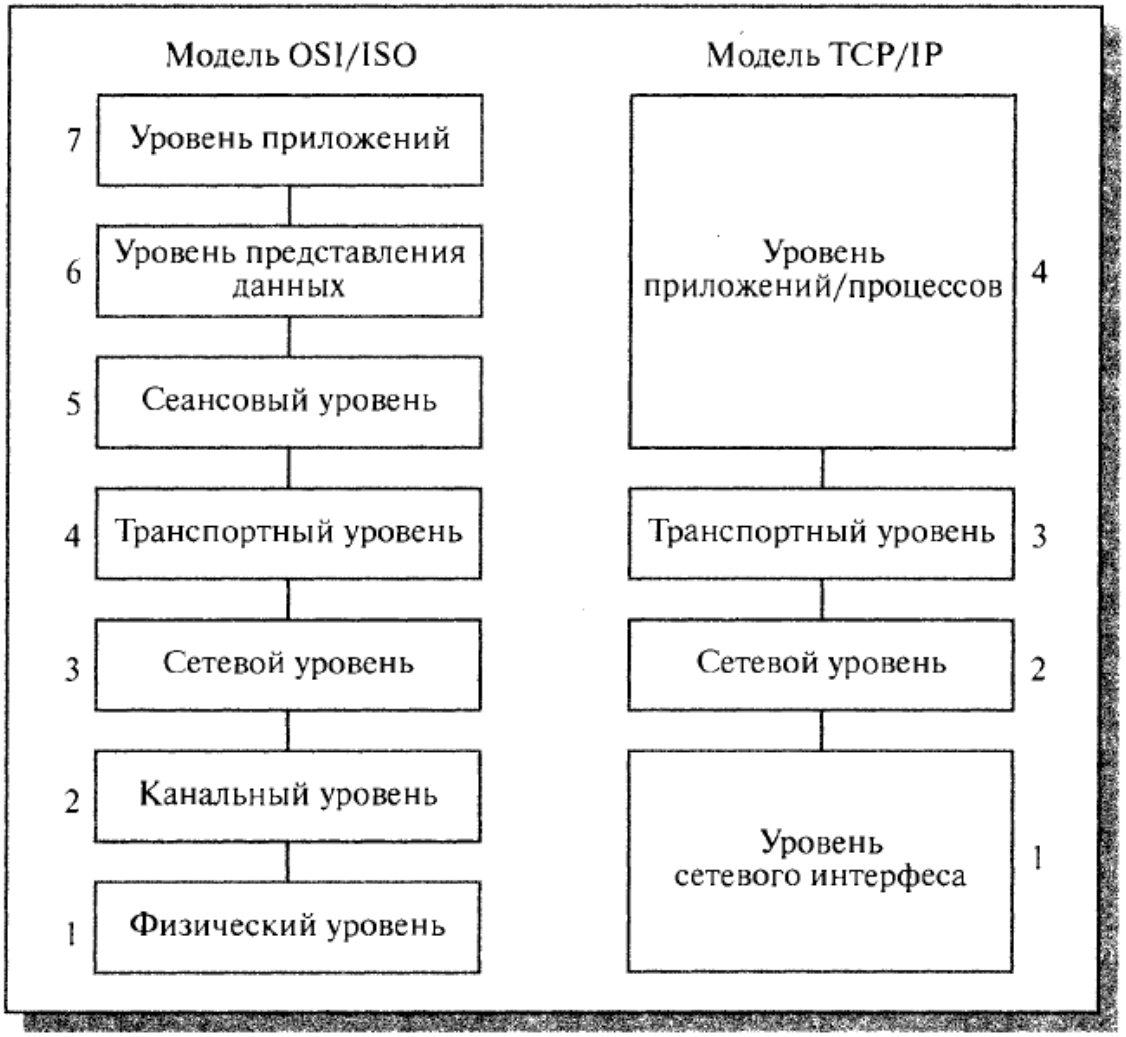
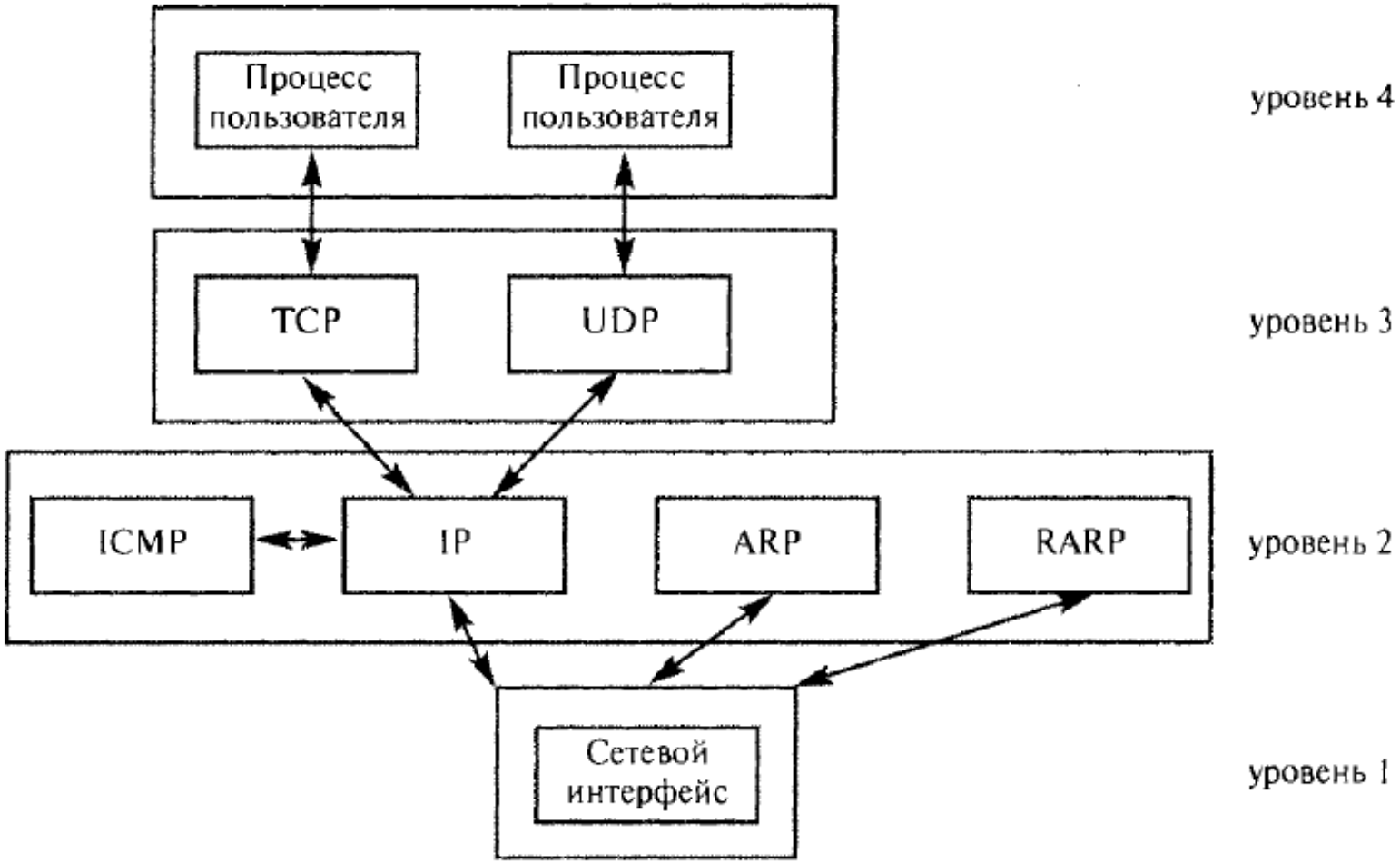


Рис. 14-15.1. Соотношение моделей OSI/ISO и TCP/IP

Основные протоколы TCP/IP



Сетевой интерфейс

- Общение физически/напрямую связанных устройств
- Ethernet, Token Ring
- На данном уровне адресами являются MAC-адреса (уникальные 48-бит для каждого Ethernet устройства).

Сетевой уровень

- ICMP – обработка ошибок, обмен управляющей информацией
- IP – обмен информационными пакетами
- ARP – для отображения IP-адресов в MAC-адреса
- RARP – для отображения MAC-адресов в IP-адреса

Сетевой уровень

- Каждый IP-пакет хранит отправителя и получателя, поэтому может передаваться независимо от других пакетов по другому маршруту
- Ассоциативная связь между пакетами осуществляется на более высоком уровне
- Не гарантирует ни доставку информации, ни то, что сообщение доставлено без ошибок
- Осуществляет фрагментацию/дефрагментацию данных при чрезмерно большом размере пакета

Сетевой уровень

- **NAT** (от англ. *Network Address Translation* — «преобразование сетевых адресов»)
- Концентраторы, коммутаторы, маршрутизаторы.

Транспортный уровень. TCP/UDP

- TCP (потоковый)
  - Проверка контрольных сумм
  - Передача подтверждения в случае успешной доставки
  - Повторная пересылка в случае ошибки
  - Правильная последовательность пакетов
  - Контроль скорости передачи
- UDP (сообщения)
  - Никаких проверок нет –> быстрый

Транспортный уровень. TCP/UDP

- Отличие от IP
  - Проверка корректности сообщения
  - Отправка сообщения не от узла к узлу, а от отправителя к получателю (через много узлов – более абстрактное представление)

Жизнь UDP-пакета

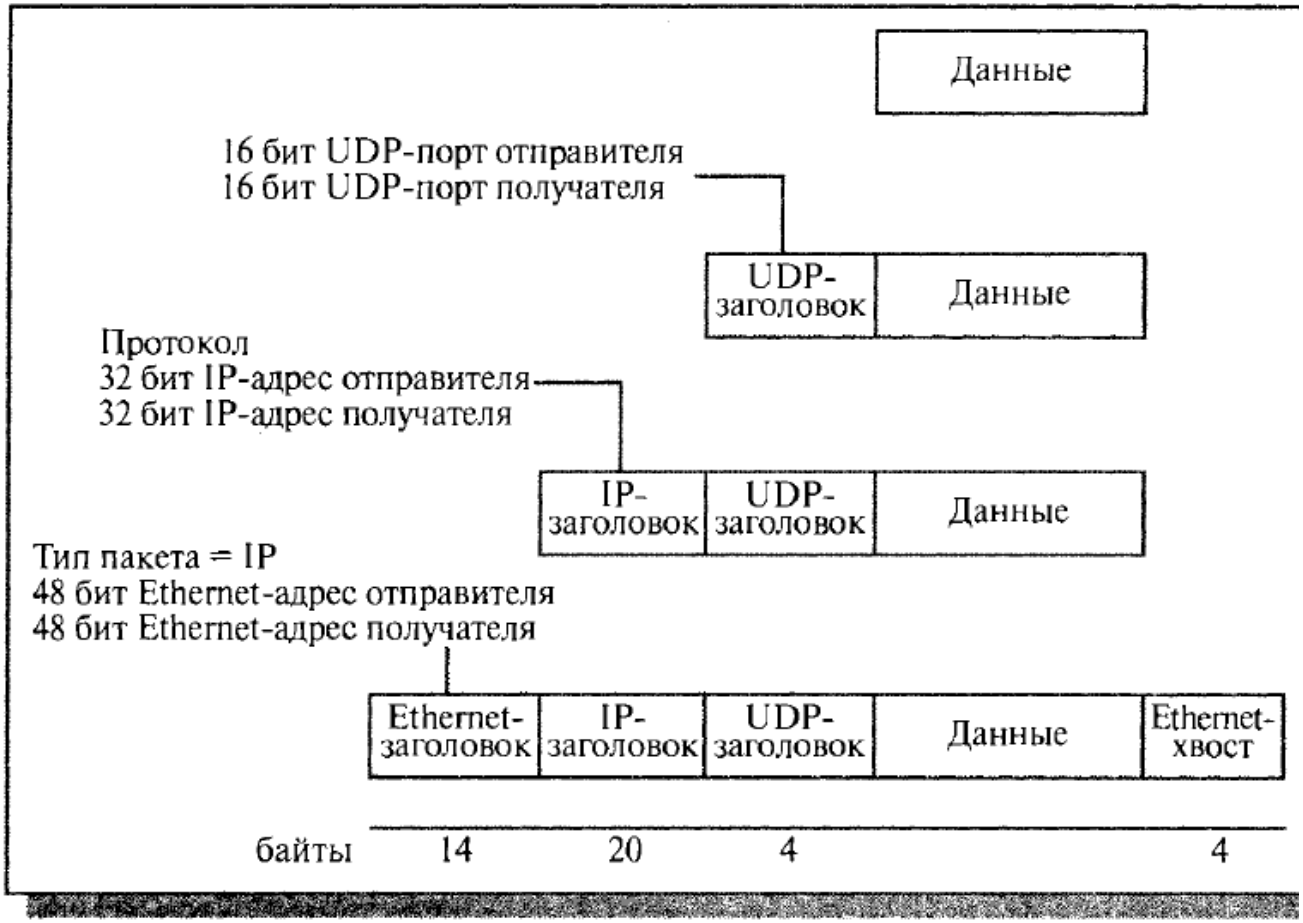


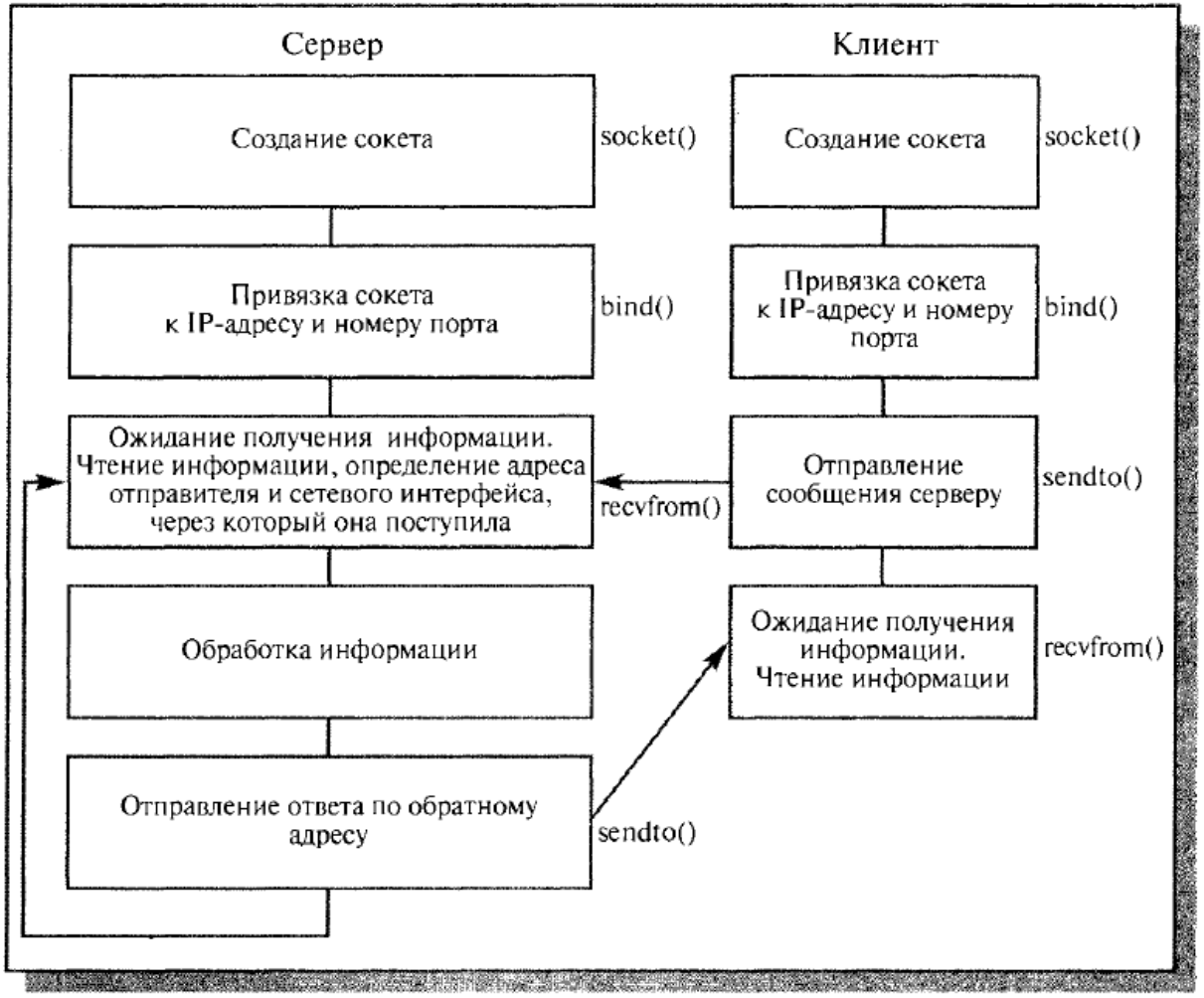
Рис. 14-15.5. Encapsulation для UDP-протокола на сети Ethernet

Работа с UDP протоколом

Рассмотрение этой схемы мы начнем с некоторой житейской аналогии, а затем убедимся, что каждому житейски обоснованному действию в операционной системе UNIX соответствует определенный системный вызов.

В.Е. Карпов

Работа с UDP протоколом



Сетевой порядок байт

- Сложность с числовыми данными > 1 байт
- Старший байт числа имеет адрес меньше, чем младший (big-endian byte order)
- little-endian byte order
- У разных вычислительных систем может быть разный порядок байт
- По сету «путешествует» всегда big-endian, а уже отправитель или получатель при необходимости переконвертируют

Сетевой порядок байт

```
#include <netinet/in.h>
unsigned long int htonl(unsigned long int hostlong);
unsigned short int htons(unsigned short int hostshort);
unsigned long int ntohl(unsigned long int netlong);
unsigned short int ntohs(unsigned short int netshort);
```

Преобразование IP-адресов из текста в число

```
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
int inet_aton(const char *strptr,
   struct in_addr *addrptr);
char *inet_ntoa(struct in_addr *addrptr);
```

```
struct in_addr {
    in_addr_t s_addr;
};
```

Заполнение последовательности байт нулями

```
#include <string.h>
void bzero(void *addr, int n);
```

Создание сокета

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(int domain, int type, int protocol);
```

С каким уровнем семейства протоколов TCP/IP будет работать?

AF_INET – транспортный уровень

Какой именно тип протоколов использует?

SOCK_DGRAM – сообщения (UDP), SOCK_STREAM – поток (TCP)

Последний параметр 0, т.к. на транспортном уровне всего один потоковый протокол и один протокол, основанный на передаче сообщений

Создание сокета

- Системный вызов socket возвращает номер файлового дискриптора, помещаемый в ту же таблицу, что и для файлов и для pipe`ов.
- Дальше уже можно использовать read, write, close

Привязка сокета к конкретному адресу и порту

```
struct sockaddr _in{
   short sin_family; /* Избранное семейство протоколов
      - всегда AF_INET */
   unsigned short sin_port; /* 16-битовый номер порта
      в сетевом порядке байт */
   struct in_addr sin_addr; /* Адрес сетевого
      интерфейса */
   char sin_zero[8]; /* Это поле не используется,
      но должно всегда быть заполнено нулями */
};
```

Привязка сокета к конкретному адресу и порту

- В качестве номера порта передаётся либо 0, тогда ОС сама выберет свободный, либо заранее оговоренное положительное число (49152 – 65535)
- Если сетевых карт несколько, то можно либо привязать к определенной, задав в sin_addr.sin_addr определённое число, либо к произвольному, указав константу INADDR_ANY

Привязка сокета к конкретному адресу и порту

```
#include <sys/types.h>
#include <sys/socket.h>
int bind(int sockd, struct sockaddr *my_addr,
    int addrlen);
```

Отправка и приём сообщений

```
#include <sys/types.h>
#include <sys/socket.h>
int sendto(int sockd, char *buff, int nbytes,
    int flags, struct sockaddr *to, int addrlen);
int recvfrom(int sockd, char *buff, int nbytes,
    int flags, struct sockaddr *from, int *addrlen);
```

Как узнать свой ip-адрес/mac-адрес?

- Из консоли:
  - UNIX – ifconfig
  - Windows – ipconfig
- Локальный интерфейс:
  - 127.0.0.1