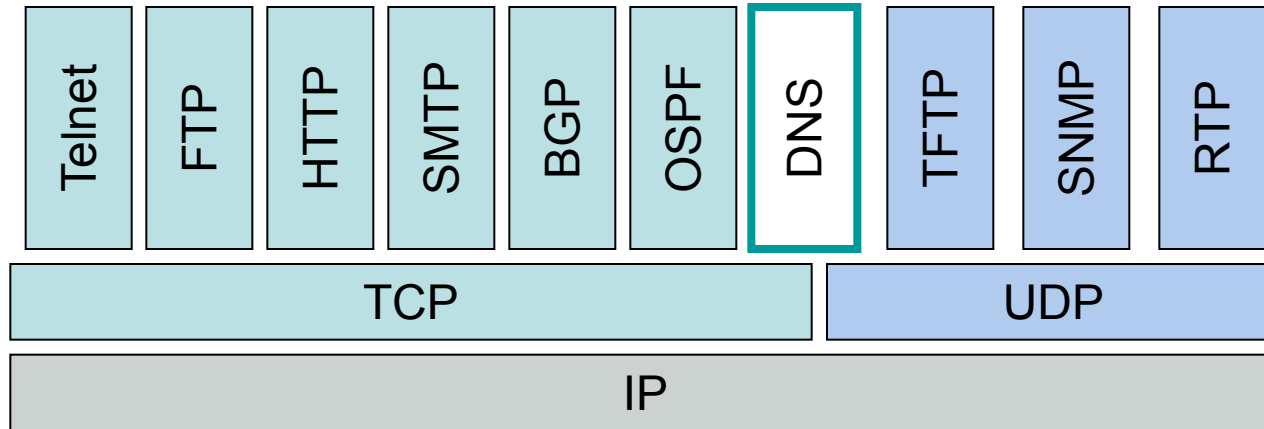


# Протоколы прикладного уровня

# Протоколы прикладного уровня



Каждый протокол разрабатывается для конкретного приложения и работает с определенным портом (логической точкой подключения).

Порты с номерами 0-255 зарезервированы для системных процессов, остальные выделены для пользовательских процессов.

Примеры портов некоторых портов:

23-Telnet; 25-SMTP; 110-POP3; 53-DNS; 161-SNMP; 8080-HTTP; 3128-Proxy

# Функции наиболее распространенных протоколов прикладного уровня

**Telnet** – протокол удаленного доступа, позволяет входить в систему и выполнять в ней действия.

**FTP** – пересылка файлов и удаленный доступ к ним.

**SMTP** – отправка почтовых сообщений

**POP3** – прием (закачка) почты

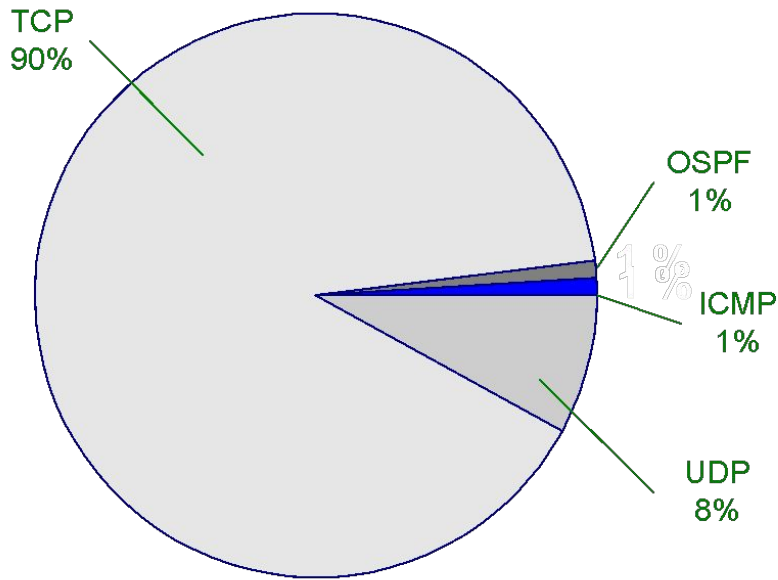
**HTTP** – просмотра гипертекстовых страниц

**SNMP** – удаленное управление системой, используется для удаленного администрирования

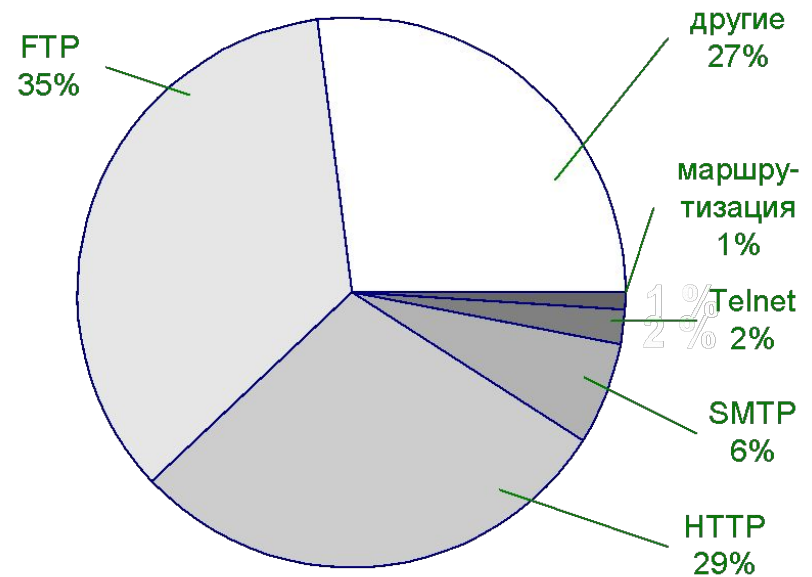
**TFTP** – облегченная версия FTP, например, применяется для организации удаленного управления бизнес-процессами.

**RTP** – протокол реального времени. Используется в IP-телефонии для передачи речевого трафика.

# Распределение полосы пропускания между протоколами



На сетевом уровне (IP)



На транспортном уровне (TCP)

# Типы протоколов прикладного уровня

Функции протокола прикладного уровня определяются приложением и типом трафика, этим приложением генерируемым.

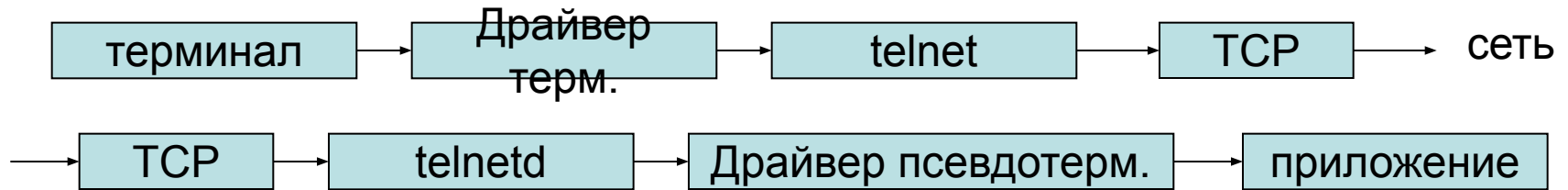
- Протокол, устанавливающий соединение и передающий параметры этого соединения. К таким протоколам относится SMTP и т.д. В этом случае протокол передает в сеть пакеты, содержащие служебную информацию, такую как номер порта, размер блоков и т.д., но может не дописывать заголовков к пользовательской информации. Пакеты с данными пользователя формируются сразу как пакеты протокола транспортного уровня.

- Протокол, формирующий заголовок на прикладном уровне с параметрами, которые необходимо передавать в каждом пакете пользовательских данных. Например, протокол RTP, используемый в IP-телефонии, содержит информацию о номере пакета в потоке, что позволяет обрабатывать фрагменты речи на приеме в необходимом порядке.
- Протокол, объединяющий в себе оба предыдущих типа. Например, HTTP сначала формирует несколько запросов, позволяющих синхронизировать работу клиента и сервера, а после дописывает заголовок к каждому блоку пользовательской информации, содержащий размер передаваемого элемента, тип кодировки и т.п.

# Telnet

- Протокол удаленного доступа к узлу. Порт 23. Позволяет организовать вход в систему и управлять удаленным узлом непосредственно. После установления соединения все действия выполняются на удаленной машине, а не на локальной.
- Поддерживает механизмы:
  - Сетевой виртуальный терминал (используется также в протоколах FTP и SMTP). Если устройство не поддерживает какие-либо функции физически, то виртуально не поддерживает тоже.
  - Механизм согласования параметров каждого сеанса клиент-сервер
  - Симметричен относительно сеанса (клиент-сервер, сервер-клиент)

- Основан на двух утилитах: telnet (на локальной машине) и telnetd (на удаленной машине).
- Схема установления связи:



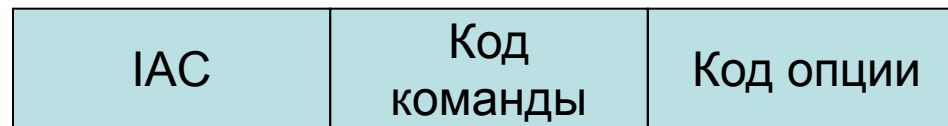
telnet 195.15.27.37

Connected to root-machine (ответ удаленной машины)

login: guest

password: \*\*\*\*\*

- Длина команды 2-3 байта:



\*IAC – символ начала управляющего пакета (1 байт)



# SNMP

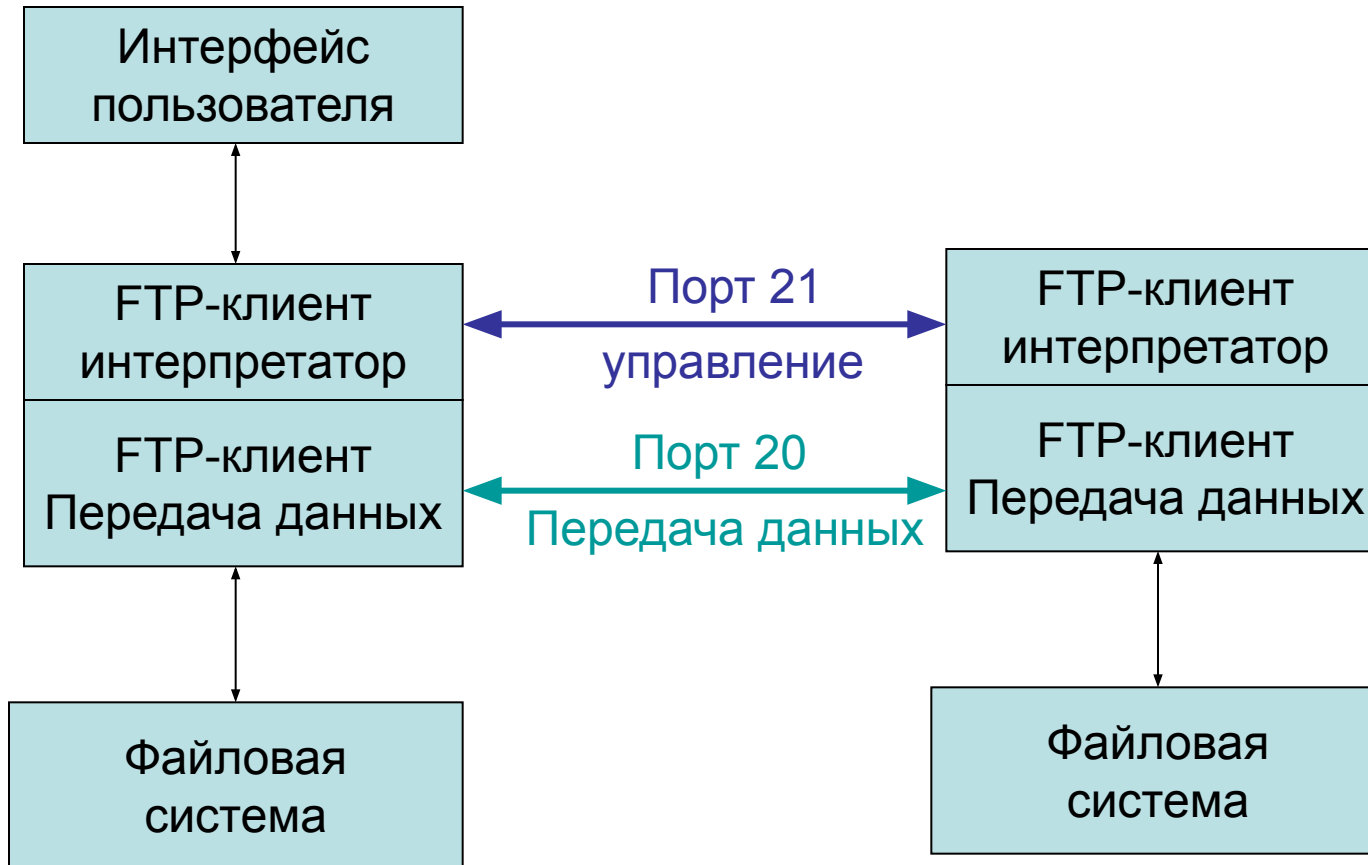
- Протокол управления, предназначен для удаленного администрирования маршрутизаторов, серверов и т. д. не зависим от IP, может работать в других сетях.
- Использует два порта: 161 и 162 (специально для сбора служебной информации).
- Идея: работа с помощью ограниченного количества команд с базой данных, реализованной на удаленном устройстве (**MIB** – Management Information Base), в которой определены операции управления, выполняемые на этом устройстве. MIB описаны в SMI – structure of Management Information.
- Вместо набора команд используется принцип выбора/хранения. Команды: выбор из MIB; сохранение выбранного значения.

- **Функции протокола управления:**
  - Перезагрузка системы
  - Добавление/удаление маршрута
  - Отключение/подключение сетевого интерфейса
  - Очистка кэш-памяти
- **Значения MIB:**
  - Таблица маршрутизации
  - Количество получаемых датаграмм
  - Количество отправленных датаграмм
  - RTT
  - Максимальное количество TCP-соединений
  - Количество сетевых интерфейсов
  - Время перезагрузки узла
  - и т.д.

# FTP

- Назначение: передача файлов и удаленный доступ к ним:
  - Удаленный доступ к файлам и их передача
  - Коллективный доступ в интерактивном режиме
  - Коллективный доступ посредством передачи
- Функции:
  - Установление сквозного соединения
  - Разрыв соединения
  - Спецификация представления данных
  - Контроль за аутентификацией (логин, пароль)
- Важно: после установления соединения все команды выполняются относительно локальной машины.

- Функции контроля за трафиком возложены на ТСР.

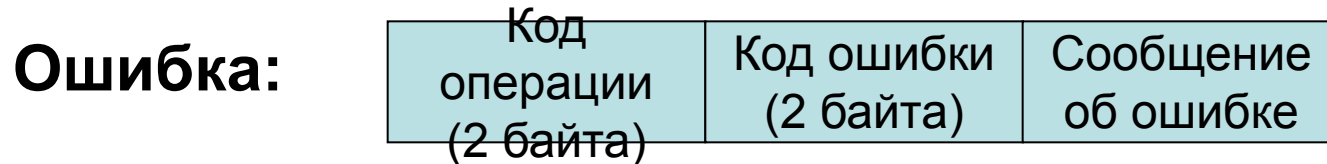
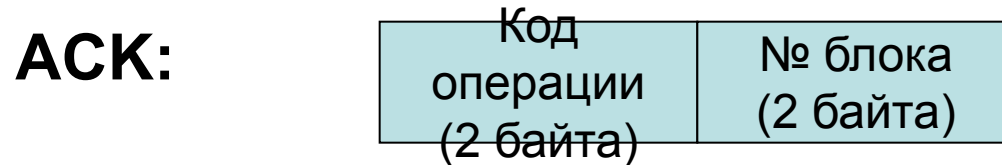
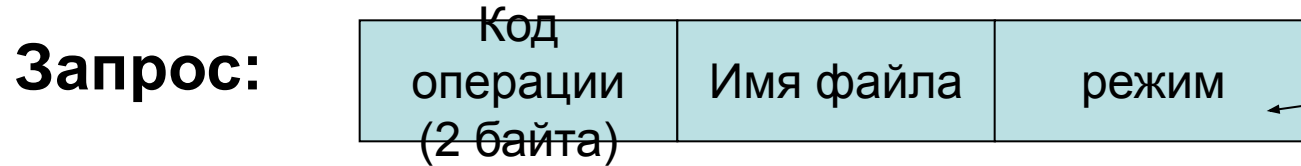


# TFTP

- Работает поверх UDP, порт на удаленной машине 69. Предназначен только для пересылки файлов.
- Функции контроля за состоянием передачи файлов возложены на TFTP. Любая ошибка приводит к аварийному закрытию сессии.
- Работает без установления соединения, следовательно, введены дополнительно функции контроля за трафиком.
- Связывание портов: порт отправителя содержится в идентификационном пакете, порт получателя 69 по умолчанию.

# Форматы пакетов TFTP

ASCII или  
бинарный

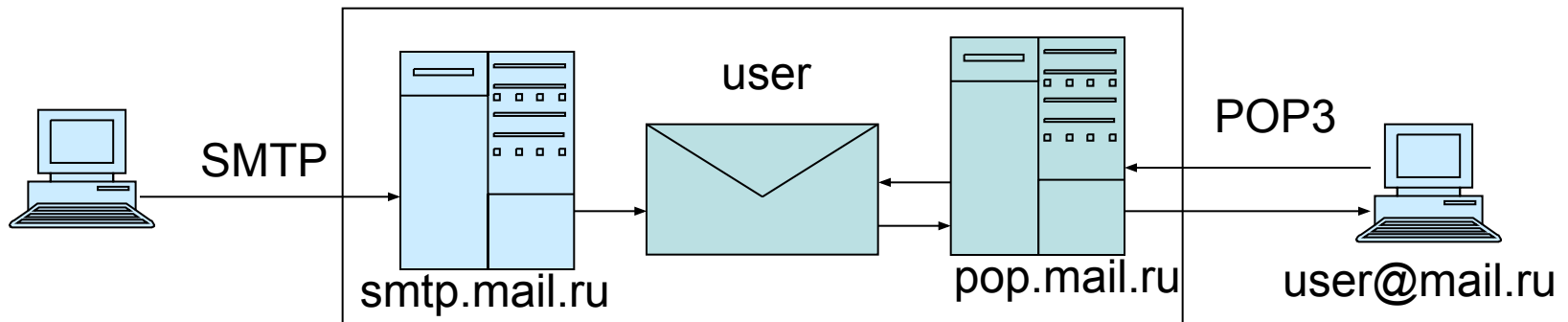


## Код операции:

- 1 RPQ запрос на чтение
- 2 WRQ запрос на запись
- 3 DATA передача данных
- 4 ACK квитанция
- 5 Error сообщение об ошибке:
  - 1 - файл не найден
  - 2 – доступ закрыт
  - 3 – диск переполнен
  - 4 – неправильная команда
  - 5 – неизвестный идентификатор

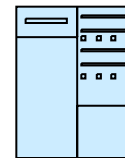
# SMTP

- Протокол для передачи электронной почты. Работает совместно с pop3 (для загрузки почты пользователем).
- Поддерживает функции:
  - Установление соединения
  - Аутентификация
  - Передача данных
- Работа с SMTP происходит непосредственно на сервере получателя:





- Основные команды:
  - MAIL адрес отправителя
  - RCPT адрес получателя (может быть несколько. На каждого получателя отдельно приходит квитанция)
  - DATA передача данных (письма). Производится в ASCII. Данные формируются как сегменты TCP. Конец письма – точка в пустой строке.
  - QUIT разрыв соединения
- Установление и закрытие соединения непосредственно с сервером производится по TCP. Контроль за пользовательскими данными возложен на TCP.



**Соединение по порту 25. Устанавливает TCP**

220 (готов)

**HELO**

250 (ok + ID сервера)

**MAIL FROM ...**

250 (ok)

**RCPT TO...**

250 (ok)

**DATA**

354 (start mail)

**данные**

250 (ok)

передача  
данных

# HTTP

- Предназначен для пересылки и трансляции (язык HTML) веб-документов. Работает поверх TCP, порт 80 и 8080.
- Особенности:
  - Установление и закрытие соединения непосредственно с сервером производится по TCP. Контроль за пользовательскими данными возложен на TCP.
  - Использует принцип запрос/ответ
  - Сервер не сохраняет параметры предыдущих сессий
  - Возможность кэширования (проxy и в браузере)
  - Симплекс (клиент может отправлять формы для обработки серверу)

# Алгоритм работы HTTP

- Анализ URL (адреса документа). Для определения IP-адреса сервера по имени хоста (host name) используется DNS.
- Установление TCP-соединения с сервером
- Отсылка запроса на сервер: GET
- Анализ ссылок без имени хоста (после первого одиночного слэша):

URL: <http://www.sut.ru/news>

Анализируется: [/news](#)

- Сообщения об ошибках отправляются на HTML.

- TCP-соединение открывается один раз на весь сеанс, следовательно:
  - Уменьшается служебный трафик
  - Уменьшается время ответа сервера
  - Уменьшается объем кэша
- Использует конвейерные запросы: несколько запросов подряд, в том числе к разным серверам, не дожидаясь ответа на каждый из них.
- Использует вложенные заголовки с параметрами сеанса.

# Типы заголовков HTTP

заголовки	Трейлер (пустая строка)	Данные
-----------	----------------------------	--------

- Content-Length: 45
- Content-Type: text
- Content-Encoding: ascii
- Content-Language: rus
- Connection: close

В заголовке прописывается длина сегмента. В случае, если длина сегмента не известна заранее, добавляется заголовок Connection, после чего идет закрытие TCP-соединения.

# Согласование параметров

1. Управление производится сервером: запрос на сервер содержит список предпочтений (например, язык – русский). Ответ сервера: страница, соответствующая запросу (загрузится русский вариант).
2. Управление производится агентом (клиентом): Запрос на сервер содержит требование предоставить список возможностей. Ответ сервера: список. Запрос на сервер: выбор из списка. Ответ сервера: в соответствии с запросом.
3. Условный запрос: загрузка параметров только в том случае, если документ удовлетворяет условию. Например, если страница была изменена с такой-то даты.